Position Papers for the ASCR Workshop on the Management and Storage of Scientific Data

https://www.orau.gov/MgmtStgeonScData

Suren Byna[1], Stratos Idreos[2], Terry Jones[3], Kathryn Mohror[4], Rob Ross[5], and Florin Rusu[6] (eds.)

January 2022

[1]Lawrence Berkeley National Laboratory
[2]Harvard University
[3]Oak Ridge National Laboratory
[4]Lawrence Livermore National Laboratory
[5]Argonne National Laboratory
[6]University of California, Merced

Disclaimer

U.S. Department of Energy Points of Contact:
Hal Finkel, hal.finkel@science.doe.gov and Margaret Lentz

https://doi.org/10.2172/1843500

Contents

Devarshi Ghoshal	Intelligent HPC Storage Systems for Scientific Workflows
Devesh Tiwari	Incentive-driven I/O Resource Management and Data Management for HPC Users
Dong Dai	Towards Unified Intelligent High Performance Computing Storage Systems
Dong Li	Scientific Data Management in Disaggregated Heterogeneous Memory
Franck Cappello	Storage system design tools for asynchronous, non-uniform, hybrid, highly distributed I/Os
Galen Shipman	Making SSIO FAIR
Gerd Heber	Harnessing Hierarchy
Glenn Lockwood	Meeting the demands of all I/O workloads all the time through dynamic reconfiguration
Greg Eisenhauer	New User Abstractions for Scientific Data Management
Guojing Cong	Intelligent data subsystems for converged AI and HPC workloads
H. Lee Ward	Direct Support For Indeing
Houjun Tang	Scientific Data Access Without Borders
Huihuo Zheng	Data Management for Scientific Artificial Intelligence Workloads
Ioan Raicu	Next Generation Indexing and Search in Large-Scale Scientific Storage Systems
Ivan Rodero	Intelligent Discovery and Delivery of Scientific Data using Knowledge Networks
Ivy Peng	The Need for Portability: Unified Interfaces to Access Scientific Data on Blurred Memory and Storage Tiers
Jay Lofstead	Revisiting Database Technology for Scientific Data
Jerome Soumagne	The Twilight of I/O as a User Concept
Joaquin Chung	Leveraging In-network and In-Storage Computation for Complex Scientific Workflows
John Wu	Automating Data Management Through Unified Runtime Systems
John Wu	Support for In-Flight Data Analyses in Scientific Workflows
John-Marc Chandonia	A self-validated data model to enable integrative, reproducible analysis
Jon Fortney	Data Fabric and Data as a "First Class Citizen"
Joshua Brown	Scientific Reproducibility and Management of Data
Justin Wozniak	Data Object Distribution for Experimental Science Pipelines
Justin Wozniak	XD/ML Pipelines: Challenges in Automated Experimental Science Data Processing

Katie Knight	Toward A Machine-Actionable Future for the DOE Science Mission
Katie Knight	Towards An International Portal Of Ontologies And Metadata Standards For Science
Katie Knight	Understanding the AI in FAIR_Data Management Support for AI and Complex Workflows
Kazi Asifuzzaman	Heterogeneous Memory System Framework for HPC
Kevin Harms	Profusion of Userspace I/O
Kirill Malkin	Composable Data Management Architecture
Kshitij Mehta	Science Campaigns: A Paradigm for Scientific Discovery for Exascale and Beyond
Lance Evans	Object Interface for Massive Storage Disaggretion
Lavanya Ramakrishnan	Supporting Next-Generation Data and Workflows
Lipeng Wan	Enabling Intelligent Scientific Workflow and Data Management through Provenance Learning
Mai Zheng	Towards Unified FAIR Metadata Services for Scientific Data
Martin Klein	Making Scientific Data Available to Machines
Matt Macduff	Scientific Data Lifecycle Management Across Storage Services
Matthew Curry	General Scalable Cooperative Provenance Capture
Michael Brim	End-to-end Scientific Data Provenance: Challenges and Opportunities
Murali Emani	HPCFAIR: An Infrastructure for FAIR AI and Scientific Datasets for HPC Applications
Nathan Tallent	Data-centric Abstractions and Adaptation to Enable Distributed Scientific Exploration
Nik Sultana	Data Management and Storage Over Programmable Networks
Norbert Podhorszki	Challenges and opportunities in utilizing AI to optimize I/O and storage
Patrick Widener	Surfacing and Exploiting Metadata Relationships for Scalable Scientific Data Environments
Peter Van Gemmeren	AI/ML for Storage Parameter Optimization of Large and Complex Data Stores of High Energy Physics Experiments
Philip Davis	Autonomic Data Management for In-Situ Workflows
Philip Davis	Models and Tools for Composing Complex in-situ Workflows
Philippe Bonnet	The promise of computational storage for scientific applications

Qing Liu	Revolutionizing the I/O Paradigm for Scientific Data Analytics
Rafael Ferreira Da Silva	The Jot and Tittle of Workflows Interoperability: Towards FAIR Computational Workflows via Metadata APIs
Rajesh Sankaran	Management and Storage of Scientific Data in the Context of Edge Computing
Saba Sehrish	Integrated data services and workflows approaches for HEP
Sandeep Madireddy	Characterization and Modeling of HPC I/O Variability through Probabilistic and Explainable AI
Sarp Oral	AI-driven Storage Resource Provisioning and Operations: Revisiting Old Assumptions and Meeting New Expectations
Sean Peisert	Co-Design to Enable Trustworthy Data Lifecycles for Scientific Computing
Sheng Di	Boosting Scientific Data Access with Usage-Driven Lossy Compression
Shigeki Misawa	Enhancing the Performance of Data Management Systems by Closing the Control Loop
Shigeki Misawa	Extending the Usable Range of Tape Systems Beyond Cold Archives
Shigeki Misawa	Rethinking Warm Data Storage
Spyros Blanas	SSIO and data management: opportunities for convergence
Stuart Chalk	Knowledge Graphs for FAIR Data and their Empowerment of Digital Twins Development
Sudarsun Kannan	Designing End-to-end HPC Data Reduction by Leveraging Smart Storage and AI Intelligence
Tejas Rao	Challenges of data management with traditional storage systems.
Tom Peterka	Storage Abstractions for Data Movement and Interoperability Between SSIO and Workflow Systems
Vincent Garonne	FAIR Data Principles at Data Centers
Wes Bethel	A Well-Designed Interface is a Trojan Horse for New Capabilities in Data Management and Data-intensive Processing
Xin Liang	Compression-Assisted Data Management in Exascale Scientific Workflow
Xu Liu	Identifying Root Causes of I/O Performance Deficit on HPC Systems through Holistic I/O Stack Analysis
Yong Chen	Fast Dataset Discovery Strategies using Efficient Metadata Search
Zhengchun Liu	FAIR Data and Model Service for AI

Call for Position Papers: Workshop on the Management and Storage of Scientific Data

Important Dates

- December 15th, 2021: Deadline for position paper submission
- January 10th, 2022: Notification of acceptance
- January 24th, 25th, 27th, 2022: Workshop
- SUBMISSION URL: https://orausurvey.orau.org/n/ASCRMSSD.aspx
- WORKSHOP URL: https://www.orau.gov/MgmtStgeonScData

Motivation

On behalf of the Advanced Scientific Computing Research (ASCR) program in the US Department of Energy (DOE) Office of Science, we are organizing a workshop on the management and storage of scientific data.

The purpose of this workshop is to identify priority research directions in the area of data management for high-performance and scientific computing above and beyond HPC's traditional "the parallel file system is the data management system" model. Supporting the breadth of the DOE mission, including the explosion of AI uses and the growing needs of experimental and observational science, motivates revisiting our assumptions about data management. There are many facets of this topic to explore, including:

- Interfaces for accessing data that resides on traditional persistent storage as well as on memory devices
- Storage-system architecture design that supports scientific workflows on varied hierarchical storage and networking devices
- Devising metadata management infrastructure to support FAIR principles (Findability, Accessibility, Interoperability, and Reusability)[1]
- Capturing provenance information about scientific data
- Utilizing AI to learn I/O patterns of emerging workloads for efficient data management;
- Providing data management support for AI and complex workflows
- Understanding the overlap between traditional storage systems and I/O (SSIO) efforts and data management.

The workshop will be structured around a set of breakout sessions, with every attendee getting the opportunity to participate actively in the discussions. Afterward, workshop attendees—from DOE, industry, and academia—will produce a report that summarizes the findings made during the workshop.

---

[1] https://www.go-fair.org/fair-principles/

Invitation

We invite community input in the form of two-page position papers that identify key challenges and opportunities in the area of systems for data management and storage. In addition to providing an avenue for identifying workshop participants, these position papers will be used to shape the workshop agenda, identify panelists, and contribute to the workshop report. Position papers should not present the authors' current or planned research, contain material that should not be disclosed to the public, nor should they recommend specific solutions or discuss narrowly focused research topics. Rather, they should aim to improve the community's shared understanding of the problem space, identify challenging research directions, and help to stimulate discussion.

One author of each selected submission will be invited to participate in the workshop. By submitting a position paper, authors consent to have their position paper published publicly. Authors are not required to have a history of funding by the ASCR Computer Science program.

Submission Guidelines

Position Paper Structure and Format

Position papers should follow the following format:

- Title
- Authors (with affiliations and email addresses)
- Topic: provide a short phrase capturing the topic(s), for example:
    - Interfaces for accessing data;
    - Storage-system architecture design;
    - Metadata management infrastructure to support FAIR principles;
    - Capturing provenance information;
    - Utilizing AI to improve I/O patterns;
    - Data-management support for AI and complex workflows; and
    - The overlap between traditional storage systems and I/O (SSIO) efforts and data management.
- Challenge: Identify aspects of current systems for scientific data management and storage that show the limitations of state-of-the-art practice with examples
- Opportunity: Describe how the identified challenges may be addressed, whether it is through new tools and techniques, new technologies, or new groups collaborating in the codesign process
- Timeliness or maturity: Why now? What breakthrough or change makes progress possible now where it wasn't possible before? What will be the impact of success?
- References

Each position paper must be no more than two pages, in single column format using 10pt or larger font, including figures and references. The paper may include any number of authors, but

contact information for a single author who can represent the position paper at the workshop must be provided with the submission. There is no limit to the number of position papers that an individual or group can submit. Authors are strongly encouraged to follow the structure previously outlined. Papers should be submitted in PDF format using the designated page on the workshop website.

Notional Questions

Position papers should present a view on management and/or storage of scientific data, perhaps taking inspiration from some of the following:

- What are potential interfaces for unifying memory and file spaces?
- What role(s) do database querying interfaces and cloud object storage interfaces play in scientific data management?
- How can we best use computing resources available in hardware such as NVMe, fabric attached storage, SmartNICs, and computational storage devices?
- What are interfaces for performing computation using in-network, in-storage devices?
- How can provenance and ontology information be used to optimize scientific data management?
- What are the costs and benefits of the rich metadata and provenance to be collected and managed for supporting FAIR principles?
- How can AI help understand I/O performance bottlenecks and variability?
- How can AI help design new HPC storage systems?
- How can we take advantage of various computing accelerators, such as TPUs, GPUs, and DPUs for data management tasks?
- How can we optimize data movement in complex workflows?
- What does it mean to bridge the gap between SSIO and data management? What technologies are needed?
- How do we make data management seamless across edge, cloud, and HPC environments?

Selection

Submissions will be reviewed by the workshop's organizing committee using criteria of overall quality, relevance, likelihood of stimulating constructive discussion, and ability to contribute to an informative workshop report. Unique positions that are well presented and emphasize potentially-transformative research directions will be given preference.

Organizing Committee

- Kathryn Mohror, Lawrence Livermore National Laboratory
- Rob Ross, Argonne National Laboratory
- Stratos Idreos, Harvard University
- Suren Byna, Lawrence Berkeley National Laboratory

- Florin Rusu, University of California, Merced
- Terry Jones, Oak Ridge National Laboratory

Sponsor: Department of Energy, Office of Science, Advanced Scientific Computing Research

DOE Points of Contact: Hal Finkel <Hal.Finkel@science.doe.gov> and Margaret Lentz

ASCR Workshop on the Management and Storage of Scientific Data

https://www.orau.gov/MgmtStgeonScData

Version: 1.0

Organizing Committee:
- Kathryn Mohror, Lawrence Livermore National Laboratory
- Rob Ross, Argonne National Laboratory
- Stratos Idreos, Harvard University
- Suren Byna, Lawrence Berkeley National Laboratory
- Florin Rusu, University of California, Merced
- Terry Jones, Oak Ridge National Laboratory

DOE Points of Contact: Hal Finkel, DOE/ASCR; Margaret Lentz, DOE/ASCR

Introduction

The purpose of this workshop is to identify priority research directions in the area of data management for high-performance and scientific computing above and beyond HPC's traditional "the parallel file system is the data-management system" model. Supporting the breadth of the DOE mission, including the explosion of AI uses and the growing needs of experimental and observational science, motivates revisiting our assumptions about data management. There are many facets of this topic to explore including:

- Interfaces for accessing data that resides on traditional persistent storage as well as memory devices;
- Storage-system architecture design that supports scientific workflows on varied hierarchical storage and networking devices;
- Devising metadata management infrastructure to support FAIR principles (Findability, Accessibility, Interoperability, and Reusability)[1];
- Capturing provenance information about scientific data;

[1] https://www.go-fair.org/fair-principles/

- Utilizing AI to learn I/O patterns of emerging workloads for efficient data management;
- Providing data management support for AI and complex workflows; and
- Understanding the overlap between traditional storage systems and I/O (SSIO) efforts and data management.

While the program committee has identified these topics as important areas for discussion, we welcome position papers from the community that propose additional topics of interest for discussion at the workshop. The workshop agenda will include breakout sessions for discussing these and selected topic areas to inform priority research directions for data management for high-performance and scientific computing.

History

Since the early 2000s, the model of “the parallel file system is the data management system” has been dominant in HPC facilities, with file systems such as Lustre and GPFS (now Spectrum Scale) being the trusted persistent store for science data near the platform. At the same time, outside of HPC platforms, a variety of technologies have emerged including GridFTP and data transfer nodes for moving data between sites, metadata catalogs such as iRODS for finding data across multiple locations, and many different forms of data services (e.g., noSQL, document stores, streaming data services) catering to different use cases. While HPC storage research continued largely to focus on how to make best use of these parallel file systems, other communities moved in new directions.

In September, 2018, the Department of Energy, Office of Science, Advanced Scientific Computing Research Program convened a workshop to identify key challenges and define research directions that will advance the field of storage systems and I/O over the next 5–7 years. The workshop concluded that addressing these combined challenges and opportunities requires tools and techniques that greatly extend traditional approaches and require new research directions. Since this time, technologies have matured, the importance of AI has become more obvious, and the need to enable greater FAIRness of data, all motivating a re-examination of these topics and more related to data management for DOE science.

Scope

Supporting the breadth of the DOE mission, including the explosion of AI uses and the growing needs of experimental and observational science, motivates revisiting our assumptions about data management. Additionally the recognition of the value of science data beyond its initial uses encourages us to embrace the challenge of enabling FAIR data principles. At the same time, the high performance, enormous capacity, and resiliency properties that have made HPC storage a success must not be sacrificed.

Changing scientific application landscape: Several trends in scientific applications have diverse data management and storage requirements - high productivity and performance APIs,

large amounts of metadata, support for FAIR principles, etc. The scientific application landscape is changing from traditional HPC modeling and simulation applications to complex workflows that involve data from scientific experiments, observations, sensors, etc. In addition, AI workloads are becoming prevalent on HPC systems. This shift from traditional single instruction, multiple data (SIMD) to multiple instruction, multiple data (MIMD) and multiple program, multiple data (MPMD) paradigms is dramatically changing scientific data management. While traditional applications focused significantly on achieving high performance, new paradigms of scientific computing are focusing not only on performance, but also on productivity. The increase in Python applications and support for Python from numerous analysis frameworks (including AI libraries) demonstrates the need for new APIs, data models, tools, and libraries for efficient data management. With massive data being produced by science applications, metadata is critical in finding the data objects of interest. For instance, a dataset from the Baryon Oscillation Spectroscopic Survey (BOSS) has 100 HDF5 files with 1.5 million objects and 144 million attribute key-value pairs. Finding desired data objects by searching the metadata efficiently requires new metadata storage, indexing, search strategies. Another common requirement of global-scale experiments and observations is sharing of data across a large number of scientists. For instance, scientists conducting climate simulations, earth science observations, microbiome data collection etc. share data in repositories. These data sharing frameworks require making data findable, accessible, interoperable, and reusable (FAIR). In addition to these trends, with memory and storage hardware and architectures on HPC systems changing drastically on exascale systems, the burden of efficient data management currently falls on application developers and end users who are often domain scientists. In order to reduce the burden on users and to achieve high productivity and performance, various R&D activities are required in the following topics.

Data access interfaces Technology shifts and trends are changing the ways that computing systems use and store program data. Conventional distinctions between memory and storage are beginning to fade, and today's restriction of either a *file interface* or a *memory interface* is artificially limiting (especially since the new heterogeneous memory architectures provide significantly reduced performance gaps between layers of the memory hierarchy). The soaring power usage of newer leadership-class machines is mandating an interest in improved power management, but this obvious need remains underutilized as current data management interfaces and infrastructure do not provide controls and configurations that can take full advantage of the underlying hardware potential. In short, while memory and storage systems have continued to evolve and improve, their interfaces have remained mostly unchanged from those that existed decades ago, at a time when main memory was strictly a volatile resource with uniform access and very limited capacity. IO interfaces designed decades ago could not have envisioned the parallel applications of today—it is not surprising that there are lost opportunities when one wishes to convey application context to the underlying IO management system through these decades-old interfaces. Innovations would seem likely in many areas including the separation of file space and memory space, the inclusion of objects and object attributes, and the advances in data management theory. These technology shifts and their implications warrant serious research efforts to reconsider, and potentially replace, the data access interfaces.

Data management architectures. Data management architectures encompass the hardware and software that together provide data management services to scientific workflows: storage and networking devices, file systems, databases, object stores, etc. Successful architectural designs enable productive interactions with data while simultaneously making best use of all the capabilities of the hardware resources. HPC data management architectures have not rapidly adapted to new workloads including AI and experimental data analysis. Research into new architectural designs holds the promise of enabling more productive data management for the rich variety of DOE science uses, building in part on advances in data management technologies developed outside HPC. At the same time, advances in hardware, including NVMe, fabric attached storage, SmartNICs, and computational storage devices provide capabilities that could dramatically change the performance and capabilities of future data management architectures, but research is needed to understand how to incorporate these technologies into specialized HW/SW systems targeting specific use cases. Special attention is needed in how to provide higher-level data management capabilities, such as indexing, that would enable FAIR data principles and elevate these architectures beyond simply “storage architectures”.

Metadata management to support FAIR principles. Storage and I/O technologies have traditionally focused on efficient data storage and access. Metadata for a long time has been used to describe the data components, such as the name of a data object or a file, access restrictions, etc. Self-describing file formats allowed storing and providing more descriptions about data objects. For instance, HDF5 or NetCDF allow describing dimensions of data and attributes to decorate data objects. With the emergence of Find-ability, Accessibility, Interoperability, and Reusability (FAIR) principles, the need for managing metadata has grown significantly. Rich metadata describing the data can speed up discoverability of data and hence the process of scientific discovery. New and enhanced methods are needed for capturing, storing, searching, and accessing machine readable and actionable metadata. I/O patterns of accessing metadata are typically random and small, and often start with a query. R&D efforts are need to develop standards, tools, and technologies for improving find-ability of data, searching massive amounts of heterogeneous metadata, increasing value of data using metadata, maintaining relationships among data objects and datasets from different data sources, maintaining metadata even when the data is no longer available, etc.

Data life cycle provenance. Data provenance, i.e., the lineage of data in its life cycle, plays a critical role in providing integrity of data and reproducibility of scientific results. In the age of artificial intelligence helping numerous fields of science in extracting patterns in large amounts of data, trustworthy data is essential. Provenance has several benefits, including strategies to optimize data movement, avoid reinvention of wheels in scientific exploration, and identify sources and users of data. Despite these benefits for scientific data, collection and utilization of provenance has been sparse or limited to specific scientific repositories. Undocumented changes to data is quite common that could lead to false conclusions in science (see, e.g., Hill, et al., 2015). With the increased use of HPC resources for experimental, observational, and sensor data, provenance gathering throughout the data life cycle becomes a requirement.

Research and software development are needed for documenting the lineage of data life cycle and workflows, annotating relationships across datasets within a repository and across multiple repositories across institutional boundaries, storing vast amount of provenance metadata using efficient data structures, searching the stored provenance metadata, utilizing the provenance for various optimizations, and generating ontologies using AI technologies.

Understanding data movement (local and remote "I/O") performance and tuning. The memory/storage hierarchy on modern computing systems becomes deeper and more complex with every new supercomputer generation. In addition to the traditional cache-ram-disk-tape tiers, SSD and non-volatile memory represent novel technologies that have to be included in the hierarchy. The proliferation of specialized accelerators, such as GPUs and TPUs, which have their local memory, creates a separate branch that has to be added to the global hierarchy. Finally, computational resources are integrated on devices originally designed for other purposes. For example, PIMs integrate processing capabilities on high-bandwidth memory, while SmartNICs add processing to network cards. Given the transformed and heterogeneous computing landscape, data placement and processing becomes a considerably more challenging task. In particular, assigning and moving data optimally across the levels of the extended hierarchy requires novel approaches. We are interested in architectural (re-)designs of the memory/storage hierarchy that are optimized for the advanced analytics workflows specific to machine learning and AI training and prediction. The design of algorithms that optimally schedule processing and data movement across the hierarchy levels is an equally important task that requires significant work in understanding the performance of existing/proposed solutions. Lastly, we consider tuning existing pipelines for the novel analytics workloads that result in improved performance.

AI for Data Management. Data management systems and processes include numerous complex decisions, algorithms, data structures, and scheduling. AI can help reimagining those design decisions to 1) create custom data management solutions to a particular data context and 2) bring new properties that were not possible before such as reducing data footprint. For example, recent works on AI for data management include creating data structures which are customized to a particular workload to maximize their performance, system optimizers that can make more accurate decisions, and replacing data or indexing representations with models to drastically reduce memory footprint (which can in turn be utilized for other components of a data system). We are interested in all aspects of AI-enhanced data management from system design, core components, and all processes and operations for the whole lifecycle of data management.

Data Management for AI and complex workflows. The I/O workloads on HPC systems are changing rapidly. By and large, traditional scientific simulations exhibit simple I/O patterns: read in input at the beginning of an execution, periodically write data files during the execution, and write final data files at the end of the execution. However, now we are seeing a rising trend of packaging traditional scientific simulations into workflows that include complex interactions between the scientific simulations and analytics functionality, e.g., AI or in situ analytics. In these workflows, the data generated by the traditional scientific simulation is input to the analytics

components that can perform many functions on the simulation data. For example, the analytics components can determine whether a particular region of a large parameter space is "interesting" and steer the execution to include more simulations of that region of the parameter space. As another example, the analytics component can perform in situ data reduction operations on the simulation data to greatly reduce the data volume produced. The simulation data can be exchanged with the analytics components in multiple ways, e.g., through files or through an interface and library that provide tightly-coupled interaction between the simulation and analytics components. There is a great need for supporting the interchange of data in complex workflows, from understanding the data dependencies between execution components in workflows, to developing interfaces for exchanging data between workflow components, to developing robust temporary file system support that is efficient for scientific simulation I/O as well as AI/ML I/O.

Bridging the gap between SSIO and DM. Success for the next generation of HPC workloads requires research in the overlap of data management and storage systems and I/O (SSIO). This is largely due to the huge shift in data and I/O requirements in HPC workloads, moving from the simple I/O patterns of scientific simulations to the diverse I/O patterns seen in complex workflows. Emerging HPC workflows require support for widely-varying data dependency patterns between stages of workflow components and support for the vastly different I/O patterns of workflow components, e.g., heavy random reads for AI/ML components versus write-heavy, parallel I/O for traditional simulations. To efficiently support the data needs of emerging HPC workloads, we need to bridge the gap between data management and SSIO research that have traditionally been mostly decoupled areas of investigation. We need to explore methodologies to understand and support data exchange for the large variety of workflow components. At the same time, the underlying SSIO infrastructure that supports that data exchange needs to be expanded to provide efficient support for the data management infrastructure and varied I/O requirements of the workflow components. Additionally, we want to encourage collaboration between the data management and SSIO communities to enable FAIR data systems that can locate and retrieve data with high performance, taking advantage of the I/O and other capabilities of future systems.

References

Baker, Nathan, Alexander, Frank, Bremer, Timo, Hagberg, Aric, Kevrekidis, Yannis, Najm, Habib, Parashar, Manish, Patra, Abani, Sethian, James, Wild, Stefan, Willcox, Karen, and Lee, Steven. 2019. "Workshop Report on Basic Research Needs for Scientific Machine Learning: Core Technologies for Artificial Intelligence". United States. https://doi.org/10.2172/1478744. https://www.osti.gov/servlets/purl/1478744.

Fagnan, Kjiersten, Nashed, Youssef, Perdue, Gabriel, Ratner, Daniel, Shankar, Arjun, and Yoo, Shinjae. 2019. "Data and Models: A Framework for Advancing AI in Science". United States. https://doi.org/10.2172/1579323. https://www.osti.gov/servlets/purl/1579323.

Hills, Denise, Robert R. Downs, Ruth Duerr, Justin C. Goldstein, Mark A. Parsons, and Hampapuram K. Ramapriyan. 2015. "The importance of data set provenance for science." Eos 96, no. 10.1029. https://eos.org/opinions/the-importance-of-data-set-provenance-for-science

Peterka, Tom, Bard, Deborah, Bennett, Janine, Bethel, E. Wes, Oldfield, Ron, Pouchard, Line, Sweeney, Christine, and Wolf, Matthew. 2019. "ASCR Workshop on In Situ Data Management: Enabling Scientific Discovery from Diverse Data Sources". United States. https://doi.org/10.2172/1493245. https://www.osti.gov/servlets/purl/1493245.

Ross, Robert, Ward, Lee, Carns, Philip, Grider, Gary, Klasky, Scott, Koziol, Quincey, Lockwood, Glenn K., Mohror, Kathryn, Settlemyer, Bradley, and Wolf, Matthew. 2018. "Storage Systems and Input/Output: Organizing, Storing, and Accessing Data for Scientific Discovery. Report for the DOE ASCR Workshop on Storage Systems and I/O. [Full Workshop Report]". United States. https://doi.org/10.2172/1491994. https://www.osti.gov/servlets/purl/1491994.

Vetter, Jeffrey S., Brightwell, Ron, Gokhale, Maya, McCormick, Pat, Ross, Rob, Shalf, John, Antypas, Katie, Donofrio, David, Humble, Travis, Schuman, Catherine, Van Essen, Brian, Yoo, Shinjae, Aiken, Alex, Bernholdt, David, Byna, Suren, Cameron, Kirk, Cappello, Frank, Chapman, Barbara, Chien, Andrew, Hall, Mary, Hartman-Baker, Rebecca, Lan, Zhiling, Lang, Michael, Leidel, John, Li, Sherry, Lucas, Robert, Mellor-Crummey, John, Peltz Jr., Paul, Peterka, Thomas, Strout, Michelle, and Wilke, Jeremiah. 2018. "Extreme Heterogeneity 2018 - Productive Computational Science in the Era of Extreme Heterogeneity: Report for DOE ASCR Workshop on Extreme Heterogeneity". United States. https://doi.org/10.2172/1473756. https://www.osti.gov/servlets/purl/1473756.

Global data management and provenance

Aaron S. Brewster[1], Johannes Blaschke[2], Filipe R.N.C. Maia[2,3] and Jan Kern[1]
asbrewster@lbl.gov,
[1]Molecular Biophysics and Integrated Bioimaging Division, Lawrence Berkeley National Laboratory, Berkeley, CA, 94720, USA
[2]NERSC, Lawrence Berkeley National Laboratory, Berkeley, California 94720, USA
[3]Laboratory of Molecular Biophysics, Department of Cell and Molecular Biology, Uppsala University, SE-75124 Uppsala, Sweden

Topic: Data movement and provenance to support FAIR principles

Challenge: X-ray free electron lasers (XFELs) are housed in national user facilities and can produce 10's to 1000's of terabytes of imaging data daily. There are 5 operational, one at SLAC and four others outside of the US. All are highly competitive instruments producing data from a uniquely bright and coherent X-ray source. They serve as an excellent example of US and international data collection facilities producing truly massive datasets in a variety of file formats with different policies for long-term data archival and metadata provenance. Data collection and processing requirements often outstrip local processing centers requiring the transfer of big datasets from detectors to local storage and from local storage to between facilities, potentially across national and oceanic boundaries. This leads to two challenges for global data analysis:

1) *Data collection and movement.* Facilities are being upgraded to produce data at an exponentially-growing rate, causing local networking, storage, and computational resources to struggle, and in the worst cases lose data. Further, as dataset size grows larger, it is difficult to transfer data between those facilities where it is collected, and the home institutions of users. Some facilities, such as LCLS at SLAC, have collaborated with top computing facilities such as NERSC and the ESnet network to provide fast data transfer between facilities, allowing near-real time analysis and the ability to move data to computing centers after collection with relative ease.[1] Other examples include the Pohang Accelerator Laboratory in South Korea and its close collaboration with the nearby KISTI computing facility, or the European XFEL in Germany and its collaboration with the Maxwell computing cluster at DESY. However, none of these facilities with the exception of SLAC are on an international network capable of moving these data between computing facilities, which is vital for post experimental analysis. And even the LCLS to NERSC case requires intervention from experts, making it difficult to use as general users. This is in direct contrast to other large international experiments such as the LHC which ties directly into ESnet for fast data transfer to any of the member nations.

2) *Data provenance*. Adherence to FAIR principles[2] is becoming increasingly vital for XFEL scientists as datasets are aggregated between different facilities. A review of the nearly 200 datasets deposited in the XFEL raw data repository CXIDB[3] reveals that most depositions differ in file and metadata format. The CXIDB is a vital part of the XFEL community's long-term data management solution, but data provenance requires support from two parties: the data creator and the data maintainer. The creator needs to include metadata that describes the experiment and allows subsequent examination. The maintainer needs to provide interfaces for extraction and machine reading of that data. It is not enough for the creator to assume that the interfaces provided by the maintainer will intelligently extract the data in a machine readable manner, nor is it enough for the maintainer to provide brute interfaces into raw arrays without assigning meaning to them using proper metadata descriptions.

Opportunity: Data movement: new and existing facilities need to recognize the computing demands of the data they are creating as part of the design process up front, and either build or partner with computing facilities to provide these resources. Furthermore, facilities need to link into the broader scientific data transfer networks that are being developed world wide, so scientists can get data to the needed computing centers. The good news is that these networks are becoming more connected with

higher speeds, so facilities have huge opportunities to provide the last-mile connections needed to enable collaborations. We observe that similar discussions are being had both at experimental facilities, as well as HPC centers. Now is the time to act, so that the requirements from experimental facilities and computing centers can inform the design of the next generation high-speed networks.

Data provenance: the best driver of adoption of international standards has consistently been funding mandates tied closely to international advisory boards that manage archival systems. A great example of this is the wwPDB, which was cited in the original FAIR paper. Publishing companies respond to funding mandates from governments that researchers deposit atomic coordinates into the wwPDB. The wwPDB provides a web server that validates depositions, ensuring metadata is included that sufficiently describes the experiment. International advising bodies write recommendations to the wwPDB that respond to the changing landscape of the field, adding new meanings to the ontologies maintained by wwPDB, which in turn are adopted by researchers who must meet validation requirements.

All of these systems are nearly entirely lacking for the management and deposition of raw imaging data, but the procedures and examples from successful organizations like the wwPDB are easily transferable given the right motivations from funding bodies. The path forward is clear: we need to move beyond POSIX files, i.e. data collection at the facilities needs to do more than just turn observations into raw data files. As a community, we should pay attention to the following aspects:

- Performance: we need to find solutions to avoid the network and file system becoming the bottleneck, necessitating collaboration between data collection facilities with network and data center operators.
- Accessibility and usability: we need to make data easy to find and retrieve for non-experts. This includes improving facility and networking tools, and universally adopting data object stores.

Timeliness or maturity: Data collection speeds are increasing without corresponding development of network infrastructure and provenance procedures. Indeed, LCLS-II will start to commission the superconducting accelerator in the second half of this year, aiming to increase the number of pulses per second by roughly 4 orders of magnitude. Without immediate response from the scientific community, valuable data is in danger of being lost right now. For example, the LCLS facility has committed to keep its collection of raw data locally stored on tape for 10 years since initial collection. However, data collection started in 2011 and we have reached the 10 year milestone already. These data are still valuable and relevant to current research. These problems are only getting worse as data collection proceeds exponentially, but existing procedures are known for handling the problem, they just need to be implemented. Here we take the position that all stakeholders in large data science need to come together to jointly tackle the urgent issues of data movement and provenance in a world of exponentially growing data collection rates. We highlight several paths forward, and the wwPDB as a concrete example of what the community can do on each level from technical operations, to community standards.

References

1. Blaschke, JP., Brewster, AS., Paley DW., *et. al*. Real-Time XFEL Data Analysis at SLAC and NERSC: a Trial Run of Nascent Exascale Experimental Data Analysis. To appear in CCPE. https://arxiv.org/abs/2106.11469
2. Wilkinson, M., Dumontier, M., Aalbersberg, I. *et al.* The FAIR Guiding Principles for scientific data management and stewardship. *Sci Data* **3**, 160018 (2016).
3. Maia, F. R. N. C. The Coherent X-ray Imaging Data Bank. *Nat. Methods* **9**, 854–855 (2012). https://www.cxidb.org/

Managing and Serving Geospatial and Environmental Sciences data for the AI Workflow

Aashish Chaudhary*, Jeff Baumes*, Dipankar Dwivedi^

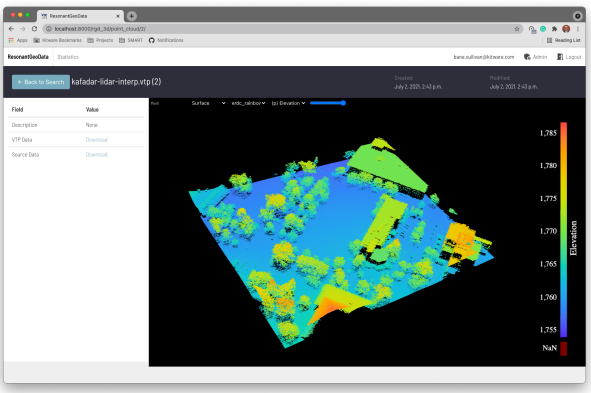
*Kitware, Inc., ^LBNL

Challenge

Recent improvements in artificial intelligence (AI) have made a huge impact on consumer product development, usage, and experiences. However, until now, most of the improvements in AI have been driven by commercial companies focused on building AI algorithms and tools around industrial datasets, which include 2D images, videos, and text. While AI has been very successful in the consumer market, its growth has lagged in some science domains even though many free software packages are available now with countless tutorials and examples. Many reasons exist for the delayed adoption of AI in the science domain, ranging from the availability of hybrid methods (e.g. AI-physics-based models) to the availability of ready-to-use AI software infrastructure for science [Ebert19]. On the software and infrastructure side, some of the reasons behind limited AI adoption include the difficulty to consume complex multi-dimensional scientific datasets, limited access to reusable models and knowledge bases, and the lack of ability to run AI models at scale on cloud and HPC infrastructure.

Current AI tools (e.g., PyTorch, TensorFlow) are not well suited to ingest scientific data formats and their associated metadata, requiring significant effort and resources to preprocess them before any scientific analysis is carried out. Each application that uses these AI tools currently requires custom data loaders for scientific data formats. Examples of such data formats include 2D-4D formats such as rasters slices (e.g. TIFF, GeoTIFF), vector geometry (e.g., Shapefile), and grids (e.g. NetCDF, HDF5). There is a need to replace the custom loaders with a simpler framework that reuses components across AI training applications.

Opportunity



With the increasing adoption of AI in the geosciences and environmental sciences community (e.g., ExaSheds), there is a need for data systems solutions that enable users to consume large amounts of scientific data and annotations in their AI workflows. While some solutions exist for 2D datasets such as Hub, they require data to be duplicated. Furthermore, there is more scarcity of tools to manage and serve non-uniform 3D and 4D datasets. Kitware, with funding from various government agencies, is building a data system, ResonantGeoData [rgd21] to manage and serve scientific 2D and 3D datasets using open source technologies and open standards (see Figure above). While we have been working with NOAA on developing DIVE, a data hosting and do-yourself-AI tool [dive21], more needs to be done to develop standardized data and annotation formats, and tools for sub-domains of science. There is an opportunity to develop better tools for managing and serving scientific datasets for consumption by AI workflows as discussed below:

Managing scientific data for AI development. AI model training requires large amounts of training data which means that the data needs to be stored in a scalable storage system. This necessitates a need for APIs and tooling to upload scientific data to scalable storage systems. Furthermore, a data system that is designed to support AI workflows needs to automate indexing to support search, specifically faceted

search to ensure data can be searched efficiently. In certain cases, a data system may need to perform preprocessing to create efficient data representations for AI model consumption. For instance, in the case of remote sensing data, a data system needs to perform different modes of data extraction and representations such as data chunking and tiling.

Serving AI-ready data. Serving AI-ready data needs to support both web and command-line clients. For web clients, it needs to support all of the data access and transformation operations via RESTful API. It is important that these APIs follow open standards such as OpenAPI. For developers and data scientists, it is important to have a command-line application and programming interfaces in widely-used programming languages such as Python. Finally, it is always better to serve data in open standard formats. For instance, in the case of raster data, serving data in a STAC format is beneficial to ensure that data can be consumed easily across various tools. The SpatioTemporal Asset Catalog (STAC) specification provides a common language to describe a range of geospatial information, so it can more easily be indexed and discovered.

Facilitating consumption of scientific data for the AI model training. MONAI is a system that makes it easy to consume 2D, and 3D medical imaging software in AI models developed using the PyTorch library. Similarly, there is a need to create data loaders specific to AI tools such as PyTorch and TensorFlow that can fetch data from local or cloud storage, use a unified API, and develop appropriate datasets and batches. Ideally, these data loaders will bring only the required raw bytes and not the entire data and perform intelligent caching. Additionally, these data loaders need to support distributed training in more complex workflows, which involves deciding which GPU data goes to for efficient training and evaluation.

Making it easy to find relevant data. It is imperative that a data system needs to support clients that let users filter the list of data available for AI development. Visualization is one of the key means by which a user filters the relevant dataset in addition to the metadata. With the rise of web-based visualization tools, it is now possible to support sophisticated visualization of AI-ready datasets.

Timeliness / Maturity

Web-and-cloud-based ecosystem for serving AI data. The proliferation of modern web and database technologies is solving major hurdles that existed to adopt a web-based ecosystem for serving AI data. From modern frontend frameworks such as React and VUE.js to cluster orchestration systems such as Kubernetes, it is now possible to build data systems with modern web interfaces that let users upload, search, and consume AI-ready data into their workflow.

Findability, Accessibility, Interoperability, and Reuse (FAIR) principles are on the rise. There is a current realization among organizations to emphasize FAIR data practices [Wilkinson16], which will place more focus on data coherency. Science and technology communities are coming together to build open standards and APIs (such as STAC, 3D Tiles, OpenAPI) to support FAIR principles. More could be done for instance standards to catalog multi-dimensional datasets and non-uniforms data, however, now is a time to start building and prototyping data systems that are built using FAIR principles.

References

[Ebert19] Ebert-Uphoff I, Samarasinghe SM, Barnes EA. Thoughtfully using artificial intelligence in Earth science. Eos. 2019;100

[rgd21] Resonant GeoData, https://github.com/ResonantGeoData/ResonantGeoData

[dive21] DIVE, https://github.com/Kitware/dive

[Wilkinson16] Wilkinson, Mark D et al. “The FAIR Guiding Principles for scientific data management and stewardship.” Scientific data vol. 3 160018. 15 Mar. 2016, doi:10.1038/sdata.2016.18

Title: Accelerating Metadata Services with In-fabric Computing

Authors: Ada Gavrilovska (Georgia Tech, ada@cc.gatech.edu, corresponding author), Ana Gainaru (ORNL, gainarua@ornl.gov), Greg Eisenhauer (Georgia Tech, eisen@cc.gatech.edu), Scott Klasky (ORNL, klasky@ornl.gov)

Topic: Metadata management infrastructure

Challenges. Exascale systems are projected to provide much greater increases in computational speed than in I/O bandwidth [8–10]. The utilization of this additional computing power comes with the obvious challenge of the velocity and variety of I/O data elements rising faster than I/O performance improves, but beyond those obvious demands there is also the challenge that the amount of *metadata* generated in the I/O process increases dramatically. Actually, many of the techniques which might be employed to reduce and manage the flood of data themselves rely on metadata to know what data to operate upon and produce more metadata in the course of their operation, so in the course of solving one problem they are exacerbating another. For example, the fact that applications cannot afford to store all generated data drives some to adopt online (in situ)[3] techniques to reduce that data before storage. This approach has been adopted by several large simulations, such as the XGC [2] code which uses staging approaches, for example, to track important particles in the simulation. Ultimately these approaches require additional operations to highlight the 'important' particles before they can be tracked. We have also seen that in other applications, such as E3SM, we can use techniques to identify Regions of Interest, and then filter the data away from the other regions. In all of these cases, we need to use a combination of filtering with techniques to query the data to ingest only the important parts of the data. *This requires accessing metadata in order to identify those regions of the data stream, and generating metadata so that downstream processing knows the origin of the data that remains.* In these cases, which have a high velocity of data flowing, we have already seen that filtering and metadata access/creation can be expensive, since it disrupts the cache line on the nodes running the large simulation, slowing down the simulation. In fact, *a significant factor impeding the effectiveness of this class of streaming I/O techniques is the cost of generating and accessing the rich metadata needed to guide how the data is being filtered, stored, transformed, and streamed through the application workflow.*

Generating metadata, at the very least, requires additional computational resources (e.g., cycles, memory, ...), and introduces in-direct overheads related to cache pollution, particularly when generating metadata relies on multi-input operations (e.g., to determine a bounding box across locally-generated output from a single simulation step). Once data is tagged, its metadata is used in selecting, i.e., filtering, which data chunks should be streamed to which workflow component. This adds metadata distribution and lookup operations in the critical path of the workflow pipeline, and the end-to-end latency of this operation must be minimized, so as not to obviate the expected data movement benefits from data reduction. Additionally, this processing tends to create *more* metadata, to track which data has been processed, where it originated, etc. If this processing is in any way distributed, asynchronous or performed other than where the data is originally generated, this means that we've not only distributed the processing of data, but also of the production and organization of metadata, further exacerbating the overall metadata challenge.

Opportunities. In recent years have seen a ***resurgence and increased commercial availability of programmable I/O devices*** – technologies which blend compute with I/O by integrating on-device I/O processing units (IPUs). Examples of such devices include programmable data or infrastructure services processing units, network interface cards and routers, programmable storage devices, even programmable engines for data movement across memory nodes of a single server platform. While very diverse in their internal capabilities, type of I/O processing units (e.g., programmable FPGAs or general-purpose low-power Arm processors), and the manner in which they can be inter-imposed on the I/O data paths, these devices share several features: The on-device IPU opens up an opportunity where it can be used to offload computation (such as for metadata generation) from the main server components, thereby freeing up compute resources. More importantly, by applying the computation on the data as the data is being moved, it is no longer necessary to move I/O data up the memory hierarchy, reducing data movement time and energy and eliminating in-direct overheads due to factors such as cache pollution. Device-side processors also allow computation (or query) execution to be completed in a way that prevents the system-level interconnect and controllers from becoming bottlenecks. Finally, such I/O processors typically have access to on-device architectural components purpose-built for in-transit processing of I/O data (packets or blocks), such as special I/O instructions, co-processors, accelerators for checksum computations, parsing, compression, etc. This makes them well-suited to accelerate common steps involved in access, interpretation and processing of metadata.

The capabilities of programmable I/O devices comprising the data streaming fabric in high-end systems, are poised to address many of the metadata-related challenges for streaming I/O. Existing solution approaches to leverage these resources are focused on offloading stream processing operations [5, 7] or software infrastructure services [1, 6]. However, in HPC and exascale systems such fabric capabilities remain unexploited by existing software I/O stacks which control data movement and use and manage metadata. As a result, it is neither possible for existing HPC application workflows to benefit from this emerging technology tier, nor is it clear how and when potential benefits can be realized.

To address this problem, it will be important to *explore opportunities for accelerating metadata services for streaming I/O via in-fabric computing*. Specifically, we argue there is a need for a new approach, that establishes a framework that augments existing streaming I/O software stacks with capabilities to directly leverage the programmability available in emerging systems' data paths. In order to allow workflow stacks to fully leverage the emerging I/O acceleration technologies, a new approach must provide a number of new capabilities: (i) an *in-fabric processing runtime* specialized for metadata services for streaming data, such as for inlining metadata generation and for accelerating metadata queries and dissemination, (ii) accompanying *programming abstractions and primitives* and (iii) *compilers and code-generation support*; (iv) *orchestration and management logic* responsible for per-node control and configuration of the in-fabric processing engines, including when dealing with multi-tenancy due to co-running workflows; and (v) *APIs and libraries* that expose these capabilities to applications, upper-layer software components, and data management services.

Timeliness A new approach has the potential to create the I/O acceleration capabilities that will enable offloading and acceleration of I/O-related metadata operations by leveraging computational capabilities of programmable I/O devices that are currently being put in production, with enhanced, more robust versions scheduled on the roadmap over the next few year horizon. These I/O technologies are being already adopted and shown critical for improving the performance and efficiency of hyperscalar datacenter infrastructure [4, 1]. As more of these technologies become commercially available, and new capabilities are designed around new interconnects, fabric-attached accelerator and memory components, there is an opportunity to achieve a new level of *readiness* of the I/O infrastructure used in HPC systems for the deployment of these new technologies in exascale system designs.

References

[1] Introducing the NVIDIA DPU. www.nvidia.com/en-us/networking/products/data-processing-unit/. 2

[2] C.-S. Chang, E. Dart, V. Dattoria, M. Ernst, P. Henderson, M. Hester, S. Jardin, S. Kaye, S. Klasky, R. Laviolette, et al. Fusion energy sciences network requirements. 2012. 1

[3] H. Childs, S. D. Ahern, et al. A terminology for in situ visualization and analysis systems. *The International Journal of High Performance Computing Applications*, 34(6):676–691, 2020. 1

[4] D. Firestone, A. Putnam, et al. Azure accelerated networking: Smartnics in the public cloud. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)*, 2018. 2

[5] T. Hoefler, S. Di Girolamo, K. Taranov, R. E. Grant, and R. Brightwell. sPIN: High-performance streaming processing in the network. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, page 59. ACM, 2017. 2

[6] Intel Infrastructure Processing Unit and SmartNIC. https://www.intel.com/content/www/us/en/products/network-io/smartnic.html. 2

[7] V. Krishnan, O. Serres, and M. Blocksome. Configurable Network Protocol Accelerator (COPA) - An Integrated Networking/Accelerator Hardware/Software Framework. In *Hot Interconnects'20*, 2020. 2

[8] K. Kumaran. Introduction to Mira, 2016. https://www.alcf.anl.gov/files/bgq-perfengr.pdf. Visited June 20, 2016. 1

[9] L. Nowell. Science at extreme scale: Architectural challenges and opportunities, 2014. http://www.mcs.anl.gov/~hereld/doecgf2014/slides/ScienceAtExtremeScale_DOECGF_Nowell_140424v2.pdf.

[10] P. Thibodeau. Coming by 2023, an exascale supercomputer in the U.S., 2014. http://spectrum.ieee.org/computing/hardware/when-will-we-have-an-exascale-supercomputer. 1

Computing and Computational Sciences Directorate
Computational Sciences and Engineering Division
Computer Science and Mathematics Division

Positioning Scientific Computing for New Data Storage Solutions: Welcoming Sustainable, Compact Storage with Biomaterials

Ada Sedova (ORNL) sedovaaa@ornl.gov
Jeffrey S. Vetter (ORNL) vetter@ornl.gov

Topic

Storage-system architecture design

Challenge

As scientific computing enters a new era of AI and big data, our efforts begin to contribute more to the data storage problem which has been growing as a result of our increasingly connected, data-hungry technologies and lifestyles. Digital information that must be stored and archived is increasing at a staggering pace [1, 2]. The International Data Corporation predicts[1] that global data storage needs will increase from 33 zettabytes to 175 zettabytes between 2020 and 2025; this is far beyond the storage capacity of current technologies. Present-day solutions for storage media provide a maximal density of 103 GB/mm^3 [3]; therefore, the amount of space and energy needed to simply store warehoused archival storage (for instance, on tape) will soon become unsustainable. The power consumption required for data storage is also projected to increase rapidly in coming years [4]. Furthermore, solutions such as magnetic tape are prone to degradation, and are not amenable to recycling and thus must be incinerated (producing toxic gases) or placed in landfills.

Sustainable, non-toxic, compact, long-lasting and low-power storage solutions using biomolecules, in particular, DNA, are no longer the stuff of science fiction, as breakthroughs appear to be accelerating [3]. In fact, consortia are forming; the DNA Storage Alliance[2] includes Microsoft, Los Alamos National Lab, ETH Zurich, Seagate, and numerous other institutions and businesses. Several new technologies have been reported in the past few weeks [5, 2], including the unveiling of a commercial prototype for enterprise-level DNA storage, capable of storing 600 billion gigabytes in the same amount of space that traditional methods would use to store only 30 million.[3] In addition, technologies are reducing the latencies for accessing this new medium and may be on the order of hours in the next decade. That said, DNA storage is currently targeting data sets with characteristics such as WORN (Write Once Read Never) and WORS (Write Once Read Seldom) because it can serve markets such as ML training and inference datasets for autonomous driving. Such rapid progress in such a new medium for storage suggests that we must prepare to embrace and understand it on many levels. We will need to understand how our anticipated scientific workflows will move beyond the standard electrical and electronic tools we are familiar with, welcoming the potential inclusion of chemical synthesis and synthetic biology to our digital universe, including our tight collaborations. More importantly, as scientists at laboratories where interdisciplinary partnerships can be easily forged, we have a unique opportunity to contribute to the *development* of these new storage solutions; making use of flagship resources such as our

[1] IDC: https://www.seagate.com/files/www-content/our-story/trends/files/idc-seagate-dataage-whitepaper.pdf
[2] DNA Storage Alliance: https://dnastoragealliance.org/
[3] Catalog DNA: https://www.catalogdna.com

supercomputers and beamlines, and engaging the expert skill sets in computer science, simulation, manufacturing and fabrication, chemistry, and biosciences, in order to help accelerate progress in a much-needed solution to our storage problem.

Opportunity

Multiple aspects of DNA storage technologies are currently suboptimal and still require further development. At the same time, numerous different solutions are being presented, from the use of cell-free enzymatic methods to the design of nanoelectrode arrays with electrochemical control of solid-state synthesis. Our capabilities will allow us to collaborate across these new interdisciplinary lines, using HPC simulation and AI to help codesign optimal platforms and biochemical reactions, while also considering how standard security and I/O protocols will be best implemented with this new type of storage medium. In terms of workload characterization and optimization, the practical aspects of data reading and writing for traditional and emerging HPC scientific workflows must be optimized, including an understanding of required conditions for both productive science and FAIR principles. With the facilities and interdisciplinary teams at DOE laboratories, it could be possible to create a number of end-to-end prototype systems outside of proprietary industry control, paving the way for a more rapid production realization of this much-needed solution.

Timeliness or maturity

Progress in the DNA storage field has accelerated rapidly since the seminal result by Church and co-workers in 2012 demonstrated the encoding of a book with 53,426 words, 11 JPG images, and a JavaScript program in DNA [1]. The challenge now is no longer "how," but "using which method." It is exactly at this point in the technology that large optimization and prototyping campaigns become the most powerful, as demonstrated by Catalog's Shannon commercial prototype. The failure of the national labs to enter this field of research at this point in time in order to establish the needs of the scientific computing community would be highly regrettable. With our flagship facilities and interdisciplinary experts, the time is now to help to shape this new medium towards a sustainable solution to the storage problem.

References

[1] George M. Church, Yuan Gao, and Sriram Kosuri. Next-generation digital information storage in DNA. *Science*, 337(6102):1628–1628, 2012.

[2] Bichlien H. Nguyen, Christopher N. Takahashi, Gagan Gupta, Jake A. Smith, Richard Rouse, Paul Berndt, Sergey Yekhanin, David P. Ward, Siena D. Ang, Patrick Garvan, Hsing-Yeh Parker, Rob Carlson, Douglas Carmean, Luis Ceze, and Karin Strauss. Scaling DNA data storage with nanoscale electrode wells. *Science Advances*, 7(48):eabi6714, 2021.

[3] Yiming Dong, Fajia Sun, Zhi Ping, Qi Ouyang, and Long Qian. DNA storage: research landscape and future prospects. *National Science Review*, 7(6):1092–1107, 01 2020.

[4] Peter Hormann and Leith Campbell. Data storage energy efficiency in the zettabyte era. *Journal of Telecommunications and the Digital Economy*, 2(3):51–1, 2014.

[5] Namita Bhan, Alec Callisto, Jonathan Strutz, Joshua Glaser, Reza Kalhor, Edward S Boyden, George Church, Konrad Kording, and Keith EJ Tyo. Recording temporal signals with minutes resolution using enzymatic dna synthesis. *Journal of the American Chemical Society*, 143(40):16630–16640, 2021.

Hosting FAIR Data from Diverse Physical Science Disciplines

Adam Hayes (ahayes@bnl.gov), E.A. McCutchan (mccutchan@bnl.gov), D. Brown (dbrown@bnl.gov), National Nuclear Data Center, Brookhaven National Laboratory, and C. Soto (csoto@bnl.gov), S. Yoo (sjyoo@bnl.gov), Computer Science Initiative, BNL

Topic: Strategy for FAIR data sharing in highly fragmented fields

Challenge

Most fields in the physical sciences today are highly fragmented and specialized, with heterogeneous data output and methods. Research data come from small laboratories, user facilities of various sizes, and in some fields, from a few high-profile, well-funded facilities. For example, in nuclear and particle physics, there is a sharp contrast between large-scale facilities, such as the Large Hadron Collider ("LHC"), the Relativistic Heavy-Ion Collider, and FRIB at MSU—and relatively smaller facilities, such as the ATLAS accelerator at ANL, LANSCE at LANL, Texas A&M, and TUNL—in the level of internal support available for producing FAIR Open Data.

A large survey of researchers found that the majority of respondents were in favor of making Open Data a requirement for funded research [1], however, truly reusable data is not often available in many fields. In a study of research data in the field of hydrology [2], the authors were only able to reproduce results from 2% of 360 data sets that had been provided as Open Data (Figure 1).

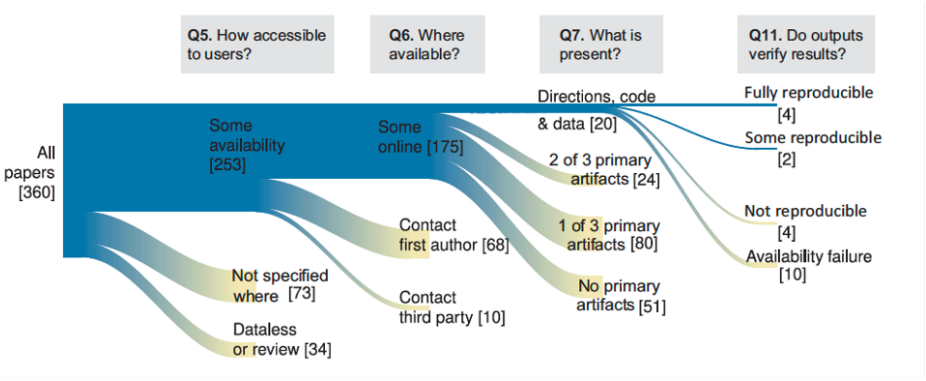


Figure 1: From Stagge *et al.* [2]. Few analyses remain reproducible following publication.

The problem space of data produced at smaller facilities with widely-varying data and analysis methods presents special needs for data management, particularly in ensuring FAIR principles and the greatest degree of reusability possible.

At the LHC, virtual machines (VMs) are available online and pre-configured for users to analyze their own data or repeat the analysis of shared data sets. In contrast, it is not a viable solution to produce a "universal" VM image that can accommodate data from the wide variety of experimental hardware and techniques at small-scale user facilities ("SSUFs").

In low-energy nuclear physics, for example, there are a number of parameters that will vary significantly from one subfield to another (*e.g.* nuclear reactions, nuclear structure, decay measurements), from one laboratory to another, and even within each laboratory depending on the detector systems used:

- detector arrangement and the conditions that trigger event collection

- data format and type (timing, energy, position, *etc.*)
- detector background measurements, calibration data and methods
- the software packages needed for analysis

Special domain knowledge is required in order to support researchers in accurately documenting these experimental parameters, the analysis methods, and the data provenance—including analysis scripts and expected output files.

Opportunity

The problem of producing and hosting FAIR Open Data from the experimental output of diverse small facilities can be addressed by creating focused host centers with an appropriate mix of (1) FAIR data experts to advise on best-practices, necessary metadata, preparation for Machine Learning applications, legal and licensing issues, and (2) scientific staff who have direct research experience with the methods and apparatus that produce the data they are curating. Data science experts can incorporate powerful AI-enabled curation techniques and ensure interoperability with AI/ML applications; a specialized scientific staff is needed so that a host facility can realistically enable independent reuse of data without extensive direct guidance from the original researchers. A general-purpose repository, however well managed, cannot fully ensure the completeness and reusability of data from myriad user facilities.

Specialized data hosts can effectively educate and train researchers in the principles, practices and preparation they need to adopt in order to produce truly reusable FAIR data. As with the data curation itself, training and education require a combination of data curation experts and scientists with domain knowledge. The concurrent access to domain scientists with expertise from multiple curated subfields also offers an opportunity to identify cross-disciplinary blind spots in data reusability efforts that may otherwise go unnoticed.

Below are some of the fundamental decisions and actions needed in the short term, in order to make FAIR Open Data more prevalent in the physical sciences.

- Identifying the facilities with the necessary expertise in both producing FAIR data *and* in one or more fields of scientific research.
- Allocating funding for FAIR Open Data curation, *e.g.* by direct support from funding agencies, and individual contributions from research grants.
- Encouraging funding agencies to require the use of a funded Open Data host in Data Management Plans.

Timeliness

The path to FAIR Open Data in all funded fields is a long one. It will be achieved on a scale of years, not in a single funding cycle, because of the diversity of fields in the physical sciences, and the special domain knowledge required of the data hosts. Ideally, guidelines and best-practices for data management would be developed even before a facility begins operations. Thus, it is imperative to develop a model—including staff, infrastructure, and funding sources—for host facilities in the very near future.

References

[1] B. Fane *et al.* The State of Open Data Report 2019. Oct 2019. https://doi.org/10.6084/m9.figshare.9980783.v2.

[2] J.H. Stagge *et al.* Assessing data availability and research reproducibility in hydrology and water resources. *Sci Data*, 6:190030, 2019.

Curate, Reuse and Reproduce Exascale Scientific Data and Campaigns

Aditya Tanikanti, Venkatram Vishwanath,Thomas Uram, Sreeranjani Ramaprakash, Michael E. Papka
Argonne National Laboratory
Email: atanikanti,venkat,turam,ramprakash,papka@anl.gov

I. Topic

We discuss challenges with data management, provenance, and data curation for science on large-scale scientific high performing computing (HPC) facilities. We explore promising research avenues to explore and tackle challenges ranging from support for automated metadata management complying with FAIR principles, intuitive interfaces for accessing and exploring large data, capturing provenance and aiding reproducibility.

II. Challenge

FAIR scientific principles, including for code and data, are critical to scientific progress and discoveries[1]. Today, a research group executes their science campaign on HPC facilities and uses the underlying filesystem to store and manage their datasets. Curation is ad hoc and requisite metadata is encoded in scripts, code, notebooks, and other sources. Science teams also need to contend with project storage capacity limits, resulting in either purging of datasets or moving datasets onto archival storage, such as tape or storage systems at other institutions. More recently, research groups have been evaluating the use of data repositories for campaigns. General-purpose scientific data repositories such as Zenodo[2], Figshare[3] and domain-specific data repositories such as Open Quantum Materials database (OQMD) [4]and NIST Materials Data Repository[5] provide for data storage and metadata management. However, the curation of data and metadata on these systems is currently a manual process: It relies on a researcher to upload the data to these repositories. Moreover, the quality of the metadata specified is at the discretion of the researcher and usually fails to capture the entire provenance of the data lifecycle. Another limitation with these data repositories is that the repositories have a storage limit of $\tilde{1}$0s GBs, which fails to meet the needs of current and future requirements of science, especially in a data-rich environment. We expect user facilities, such as light sources with their upgrades, to produce 10-100X more data in the near future [6], and to process this data across multiple facilities. Solutions that can scale to meet the data needs and accommodate solutions ranging from single tenant systems to federated systems across multiple facilities will be needed.

As science campaigns generate and analyze datasets on HPC systems, research teams do not typically capture and curate metadata needed for end-to-end lifecycle provenance as part of their workflow. This includes various job logs used to process and analyze the data, associated workflow logs, system environments, and application-specific metadata. This information is available at the compute facilities though never curated. Challenges also lie in the fact that the metadata provided is often insufficient. With scientific projects spanning across several facilities and the inter-disciplinary teams involved, the challenge to define and capture metadata is further amplified.

III. Opportunity

HPC user facilities such as ALCF, OLCF and NERSC are conduits for exceptional research providing exascale computing infrastructure in the near future. Science campaigns run complex workflows and store rich scientific data on the file systems. More often than not, the entire scientific workflow is executed within a single facility, although this is changing: we are witnessing an increasing use of cross-facility workflows to meet the needs of users, in the face of queue wait times, co-scheduling of experimental facilities and computing facilities, and planned and unplanned facility outages.

There is an opportunity to research and develop data management software that interacts directly with the data and codes on the file system hierarchy (including tape storage), and across facilities - all in an automated manner. Promising directions to explore include understanding data management frameworks such as Invenio[7], ckan[8], Dataverse[9] wherein efforts have been made to comply with FAIR standards, and how they can scale to meet the diverse needs of DOE science.

Specific opportunities include:

- Improved metadata support: Planning for metadata capture and having mechanisms for cataloging and querying this data will be essential to deriving understanding from enormous datasets. Solutions for harvesting metadata from existing datasets will open historical data to future metadata-based queries. Both of these rely on metadata support and community involvement. Data harvesting techniques like Open Archives Initiative Protocol for Metadata Harvesting (OAI-PMH)[10] and Data Catalog vocabulary (DCAT)[11] aid in making these data a part of federated search engines such as Google Datasets[12].

- Version control schemes can help maintain versions of the data through changes such as input data and execution conditions.
- As workflow tools mature and encapsulate a larger fraction of the flow of the scientific computing process, they have an increasing purview of how data was produced and can capture provenance and improve reproducibility, *assuming suitably usable metadata systems*.
- Minting DOIs allows for the datasets to have permanent identifiers and record citations, making them accessible via current and future filesystems at DOE facilities, aiding the capture of provenance.
- The maturity of the curation process varies across scientific domains – some domains having detailed and pretty standard ontologies and others who have multiple ways to curate with seemingly no agreement on an acceptable way. For curation to be scalable at the facilities and across scientific communities, there is an opportunity to drive toward a common standard together with best practices. This would involve an opportunity to work across science communities and user facilities to understand common abstractions needed for curation.
- Extensions for visualization of certain files and data for common file types such as pdfs, CSV files, geographical data, HDF/netCDF hierarchical data viewers would add a graphical component to data made discoverable by sufficient metadata.
- To support data-driven science, research teams need the ability to share, access, discover and collaborate with datasets. A key facet to realize this are policies driven by facilities and science programs to incentivize this. Another aspect is effective and intuitive interfaces for science teams to explore rapidly the large datasets at facilities, either produced by them or by other science teams.
- There is a need to develop tools to crawl through datasets at facilities to discover and curate metadata from scientific datasets. To accommodate the scale of the files, systems, and diversity of file types, we expect the need for scalable crawler and indexing agents - a large-scale computing problem. We expect one would need to leverage AI driven models to extract relevant metadata and translate them across domains spanning various ontologies, to facilitate cross-disciplinary discovery and collaboration.
- Sufficient metadata will also support the discovery and introspection of datasets by intelligent agents, specifically by artificial intelligence models that require additional training data to improve their accuracy, without the need for human intervention.

Overall, a holistic solution for reproducible and easily accessible scientific data can potentially be achieved at DOE facilities. This solution relies on scalable data management software. It requires automating the curation process and making it scale to accommodate for growing and diverse data needs. Ideally, this solution will seamlessly work with the petascale/exascale file systems at the facilities and operate without any impact on performance to the science.

IV. Timeliness or maturity

We are moving to an era wherein simulation, data and learning are needed to glean scientific insights. Without effective support for data curation, provenance and management, we will fail to make progress in a wide gamut of science. Thus, we are at the juncture wherein science teams are becoming more vested to embrace strategies to curate, reuse, and reproduce data.

References

[1] "Announcement: Transparency upgrade for Nature journals". In: *Nature* 543.7645 (Mar. 2017), pp. 288–288. ISSN: 1476-4687. DOI: 10.1038/543288b. URL: https://doi.org/10.1038/543288b.

[2] *Zenodo*. https://zenodo.org/.

[3] *Figshare*. https://figshare.com/.

[4] James E. Saal et al. "Materials Design and Discovery with High-Throughput Density Functional Theory: The Open Quantum Materials Database (OQMD)". In: *JOM* 65.11 (Nov. 2013), pp. 1501–1509. ISSN: 1543-1851. DOI: 10.1007/s11837-013-0755-4. URL: https://doi.org/10.1007/s11837-013-0755-4.

[5] Talapady N. Bhat et al. "Strategy for Extensible, Evolving Terminology for the Materials Genome Initiative Efforts". In: *JOM* 67.8 (Aug. 2015), pp. 1866–1875. ISSN: 1543-1851. DOI: 10.1007/s11837-015-1487-4. URL: https://doi.org/10.1007/s11837-015-1487-4.

[6] *Office of Basic Energy Sciences Update*. https://science.osti.gov/bes/besac/Meetings/Meeting-Presentations/201907. 2019.

[7] *Invenio RDM*. https://inveniosoftware.org/products/rdm/.

[8] *ckan*. https://ckan.org/.

[9] *Dataverse*. https://dataverse.org/.

[10] DeeAnn Allison. "OAI-PMH Harvested Collections and User Engagement". In: *Journal of Web Librarianship* 10.1 (2016), pp. 14–27. DOI: 10.1080/19322909.2015.1128867. eprint: https://doi.org/10.1080/19322909.2015.1128867. URL: https://doi.org/10.1080/19322909.2015.1128867.

[11] *Data Catalog vocabulary (DCAT)*. http://hdl.handle.net/10421/7474/. 2014.

[12] *Google Datasets*. https://datasetsearch.research.google.com/.

AI Methods for Efficient Data Management and IO Characterization

Ahmad Maroof Karimi, Sarp Oral, Feiyi Wang
{karimiahmad, oralhs, fwang2}@ornl.gov
Oak Ridge Leadership Computing Facility, Oak Ridge National Laboratory

Topic: Utilizing AI to learn I/O patterns of emerging workloads for efficient data management.

Challenges: With the onset of exascale HPC systems and future zettascale supercomputer, we will have an unprecedented volume of data at a very high speed both for large-scale scientific workloads as well as the operational data generated by sensors and characterization tools for monitoring the various aspects of HPC workloads. In our recent publication [1], we have analyzed about a million HPC workloads collected by HPC IO characterization tools and proposed a methodology for the identification of HPC IO patterns of emerging workloads. However, during the project, we realized that a critical aspect of improving the system is IO workload characterization, and to have real-time analysis for detecting new IO patterns would require solving the critical challenges related to efficient data management and computation. Similarly, while working on another project for analyzing and predicting power characteristics of an HPC facility, an effort towards building an intelligent system requires the processing of billions of data points, with each data point having hundreds of features[2][3]. We realized the need to build a priority-based standard for feature selection and dimensionality reduction to avoid dumping unproductive data in the data lake and make space for more valuable data as well as to prevent the wastage of computation time & resources.
As the demand for building more intelligent HPC analysis and monitoring tools grows, the data-driven based AI models would become more data greedy and require higher resolution datasets. The sheer size of the raw dataset collected by sensors and characterization tools at a high frequency of 1Hz, which is needed for more profound analysis and predicting tools, would be of the order of size of several hundred PB for a few year periods. Even the largest file systems like Alpine for Summit or upcoming Orion for Frontier supercomputer would be insufficient to store data of such high volume. The problem of enormous data volume leads to another challenge of conflicting requirements of data reduction necessities and the high accuracy of the data-greedy AI models.

Opportunity or Research Direction: We propose that the opportunities lie in data-driven and AI model-centric techniques to address the above challenges, in contrast to more conventional kernel-extraction or mini-proxy based approach. We can classify the scope of the research direction into a few broad categories. The first approach is to leverage AI and deep learning techniques to engage in data reduction methodologies. It will help reduce the volume of the data while still maintaining data integrity. The data reduction methods will allow machine learning models to process a large amount of information representing original data but much less in volume. For instance, the limitations in existing IO characterization tools, which usually collect summarized data, or if it collects data with temporal characteristics, then it severely affects the performance of HPC workloads, there is a need to build machine-learning models for generating IO traces of HPC workloads. The machine learning models should fulfill the missing gaps by rendering high-resolution datasets without impacting the performance of the workloads. At the same time, there is an opportunity for developing machine learning models for feature selection or dimensionality reduction on data collected from the IO characterization tools, thus facilitating efficient storage and processing tasks. There hasn't been a standard set of features defining IO profile, so we also need to design an IO trace generation model and develop a standard set of metrics that can represent the IO traces with high precision and accuracy. The intelligent AI model should also be able to generate traces with a more fine-grained resolution having temporal characteristics, unlike

summarized data generated by IO characterization tools. These AI models will be expected to run independent of the HPC workloads, except collecting features from IO characterization tools, causing a minimal effect on the performance of jobs. Developing data-driven parametric models for IO trace generation is another method where instead of storing the actual data, the models can generate traces. Another dimension of this research work is analyzing the IO performance of a job on all other concurrent jobs running on the same filesystem. Dynamic network modeling has been proven to be an effective approach to understand the effect of one node or set of nodes on the system and can be extended for modeling the IO traces exhibited by jobs running in parallel. The network training needs high-resolution data, and the generative model we proposed above can play a significant role in building the required dataset. We consider that this approach can be further advanced to develop an intelligent scheduling system. If a network model can reliably estimate the IO load of a submitted job on the system. In that case, the network can provide feedback to the job scheduler to re-order the jobs in a queue to maintain the IO load balance on the filesystem server.
The second approach is to build a high-fidelity AI workflow pipeline that can perform effective analysis with the reduced dataset. The direct consequence of dataset reduction, if not done carefully, will affect the performance and accuracy of all data-driven approaches, be it analysis or modeling. Future-generation data-driven based AI models for IO profiling and characterization should apply online-learning strategies so that IO models should be trained continuously. The models based on online learning and the self-training approach will be more reliable towards evolving HPC workloads IO patterns. However, online learning methods must perform in an unusually dynamic and noisy environment. Identifying temporal patterns for trends and seasonality will be challenging to ascertain concept drift or data drift. Signal decomposition methods can be employed for disaggregating trends and seasonality to help trigger online learning models for training.

Timeliness: Evolving workloads running on exascale HPC computing would reveal new IO patterns. Exascale HPC computing will enable new workloads exhibiting evolving IO trends. The demand and resources for running more computationally complex, diverse, and extended workloads are increasingly available. The emerging workloads are compelling HPC scientists and engineers to profile and predict IO patterns more accurately and reliably. The need for real-time analysis is pushing the boundaries of the existing approaches to process far more information in either real time or near-real time. It also brings the need to manage the data smartly to use the same amount of limited disk resources to store more information and consecutively reduce the amount of data that the AI models have to process. We believe that the proposed data-driven and model-centric approach and advances in AI/ML algorithms and computational framework will likely shed new light and open up avenues to solve the challenges of the SSIO community.

References:

[1] Arnab K. Paul, Ahmad Maroof Karimi, and Feiyi Wang, “Characterizing Machine Learning I/O Workload on Leadership Scale HPC Systems”, 29th International Symposium on Modeling,
Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), 2021.
[2] Woong Shin, Vladyslav Oles, Ahmad Maroof Karimi, J. Austin Ellis, and Feiyi Wang, “Revealing Power, Energy, and Thermal Dynamics of a 200PF Pre-Exascale Supercomputer”, 2021 Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC’21).
[3] Chengcheng Li, Ahmad Maroof Karimi, Woong Shin, Hairong Qi and Feiyi Wang, "The Challenge of Disproportionate Importance of Temporal Features in Predicting HPC Power Consumption," 2021 IEEE International Conference on Cluster Computing (CLUSTER), 2021.

Automating AI workflows used by HPC scientific studies

Ana Gainaru[1], Tahsin Kurc[2], Steven Young[1], Dmitry Ganyushin[1], Joel Saltz[2], Scott Klasky[1]

[1] Oak Ridge National Lab - {gainaru, youngsr, ganyushindi, klasky}@ornl.gov

[2] Stony Brook University- {tahsin.kurc, joel.saltz}@stonybrook.edu

Topic: Providing data management support for AI and complex workflows

Challenge

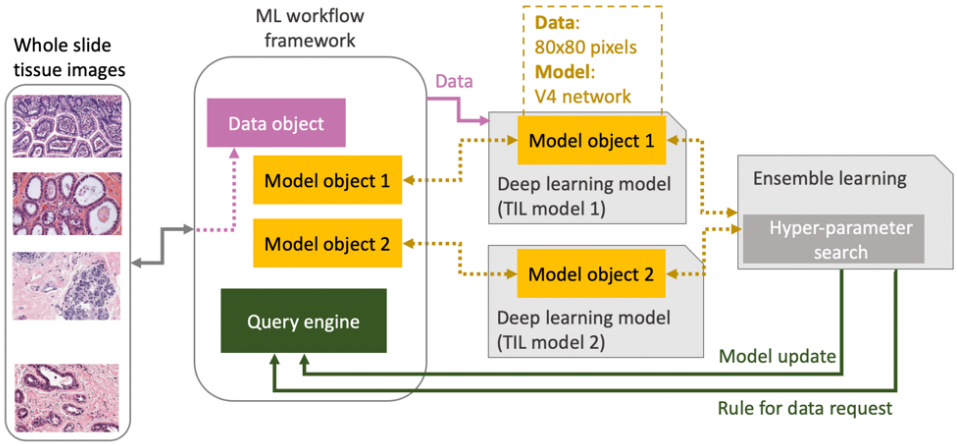
Artificial Intelligence (AI) methods have shown impressive performances in a variety of data analysis tasks, especially in prediction and classification. As the scientific community faces unprecedented amounts of complex scientific datasets, either generated by advanced instruments or by large-scale simulations run on supercomputers at leadership computing facilities, AI methods are rapidly becoming a key component of scientific applications, including applications in fusion energy science, computational fluid dynamics, bioinformatics, and medical research. Applications in these domains employ AI models for a large variety of tasks, such as classifying types of tumors in medical images [1], reconstructing the 2D plasma profile [2] and predicting protein structure properties [3]. In addition, more and more domain scientists are replacing expensive computational kernels within their codes with AI models. For wider and more effective adoption of AI in scientific research on HPC systems, however, we have to address not only computational challenges but also major data challenges: (i) datasets are being produced at much higher resolutions and much higher speeds than ever before, increasing the volume and complexity of data that HPC systems have to support and (ii) application of AI methods is introducing I/O patterns which are different than the I/O patterns for which HPC systems are designed for.

AI workflows involve iterative processes and neural architecture searches in which datasets are analyzed multiple times with different configurations to find accurate, reliable and efficient models. In these workflows, multiple models may be trained concurrently, and results from the models may be compared with each other to improve the models. Model training and model inference workflows may consist of multiple steps where data is streamed from one step to another. Moreover, multiple workflows may be executed in parallel. Examples of such workflows targeted by members of our group include microscopy analyses of whole slide digital Pathology images where deep learning network (DNN) models support cancer biomarker development research [1] and in materials science where analyses target error identification for electron microscopy images of materials [6]. While specialized hardware and software systems have been developed to optimize DNN performance, they are all focused on computation and not data management. Data management is implemented by each application in an ad-hoc way leading to substantial challenges at scale: (i) there are **massive amounts of data that need to be loaded into multiple AI codes**, each working on different pre-processing formats switching the focus of HPC to read intensive workloads; (ii) most recent hyper-parameter search algorithms like MENNDL [4] or CANDLE [5] are **limited to in-memory data and focus only on searching different model parameters** that give performance without investigating data pre-processing parameters; (iii) different AI methods have different I/O patterns, **optimizations are needed to efficiently support multiple I/O patterns and to adapt to future AI algorithms**.

The local I/O management is also visible in model management. Applications are generating models with different accuracies and performances (from more accurate with large memory footprint to faster but less accurate models). However this information is rarely shared among applications unless they are explicitly coded together which means scientists need to change the codes for each study that requires a new workflow.

Opportunity

Presently application domains implement ad hoc, often application specific, mechanisms to address the data management challenges. Our vision is to **separate the data and computational planes (training or inference) and offload the data management to specialized I/O libraries**. Figure 1 presents such a solution. The ML/AI workflow framework is separated from AI applications. It coordinates data reads and pre-processing with the requirements of all the AI applications running concurrently. The framework implements support for reusing data loaded from storage for all the applications that require it and keeps track of the models generated. Hyperparameter search codes can use the framework to query the data pre-processed in different ways (e.g. different tile size) and use



models with different properties. By separating the data plane from the AI application, **the framework can fetch in a more efficient way the required input data in the format and order given by the needs of the application and optimize the data path between collaborative processes**. It would also allow scientists to create new workflows coupled in a natural way based on their input needs and model requirements.

For such frameworks to be developed, the community needs to rethink how to represent and manipulate the models and scientific datasets at large-scale in a scalable and efficient manner to be able to adapt to the new access patterns introduced by the current and future generation large-scale applications that are starting to heavily rely on machine learning algorithms. The I/O patterns of AI applications (at individual application level and at workflow level) have to be systematically studied and characterized in order to implement efficient support for automating AI workflows and minimizing I/O and data management overheads. For example, data access patterns for both training and inference can be leveraged by data management systems to prioritize and pre-process the data across multiple workflows and to retrieve and deliver the data quickly where it is needed.

Timeliness

With compute power increasing at a much higher rate than storage or network technology, AI workflows that analyze large amounts of data in intricate ways are increasingly limited by the I/O sub-systems of supercomputers. Existing I/O optimizations have to be reevaluated based on the new patterns introduced by AI workflows. Development of a ML workflow framework that separates the data and computation planes can provide a flexible, adaptable, and scalable framework. Such a framework can offer high I/O performance (e.g. by adapting the reduction rate and data transfers based on network availability and coordinating the I/O of multiple applications to decrease the congestion) without having to implement application-specific solutions.

References

[1] Rajarsi Gupta, et al. Characterizing immune responses in whole slide images of cancer with digital pathology and pathomics. Current Pathobiology Reports, 8(4):133–148, Dec 2020.

[2] Diogo R. Ferreira. Applications of deep learning to nuclear fusion research, 2018.

[3] Jian Zhou and Olga G. Troyanskaya. Deep supervised and convolutional generative stochastic network for protein secondary structure prediction, 2014.

[4] Robert M Patton, et al. Exascale Deep Learning to Accelerate Cancer Research. 2019 IEEE International Conference on Big Data, 2019

[5] H. Yoon, et al. Model-based Hyperparameter Optimization of Convolutional Neural Networks for Information Extraction from Cancer Pathology Reports on HPC, *2019 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, 2019, pp. 1-4

[6] R. M. Patton, et al. 167-PFlops Deep Learning for Electron Microscopy: From Learning Physics to Atomic Manipulation, High Performance Computing, Networking, Storage and Analysis, 2018, pp. 638-648

IMPROVING DATA COLLECTION, MANAGEMENT, ACCESS, AND REPRODUCIBILITY THROUGH ENHANCED SYSTEM DESIGN

AUTHORS:

Angela Norbeck, Pacific Northwest National Laboratory, angela.norbeck@pnnl.gov
Chitra Sivaraman, Pacific Northwest National Laboratory, chitra.sivaraman@pnnl.gov

TOPICS:

Metadata management infrastructure to support FAIR principles; Capturing provenance information; Data-management support for AI and complex workflows, Model validation

CHALLENGE:

(*Identify aspects of current systems for scientific data management and storage that show the limitations of state-of-the-art practice with examples)*

Collecting and collating data with associated metadata in a centralized repository is challenging for several reasons: distributed storage, evolving technology, inadequately captured metadata, lack of community data standards and fear around data sharing and protection of intellectual property. PNNL is home to two DOE user facilities 1) the Environmental and Molecular Sciences Laboratory[1] that offers capabilities ranging from mass spectrometry of biological samples to chemical catalysis and more and 2) Atmospheric Radiation Measurement (ARM)[2] that offers capabilities to collect data from remote sensing observatories to improve the understanding of aerosols and clouds and their representation in models. Open and restricted data are stored and disseminated from individually managed repositories and portals. Data has diverse needs and its users range from instrument owners to scientists advancing their research to gaining insights into the data and to modelers who are eager to reduce the uncertainties in models.

Although many domain communities have made efforts to develop metadata standards, progress still must be made for analytical data to be FAIR. Elements that make the collection of metadata difficult include different data ontologies regarding what data are captured for experiments, observations, scientific and mathematical models, machine learning models, etc. In all these cases, there are concerns around versioning and model specifics that have a major effect on the reliability of results.

Another challenge is the ever-increasing experiment and human time required to gain insights that lead to answering complex scientific questions. Rapid, accurate, and adaptable systems are needed to facilitate data integration beyond what a human can do. All of these challenges highlight the need to focus efforts on smart data and metadata management systems and make investments in data integration, automation, and classification.

OPPORTUNITY:

(*Describe how the identified challenges may be addressed, whether it is through new tools and techniques, new technologies, or new groups collaborating in the codesign process)*

Metadata collection across domains could be enhanced using automated methods of data injection and metadata creation that do not rely on a standard consensus master identifier list, and do not require researchers to conform to any standard in their analysis or submission. Such a process would address the need for generating metadata and capturing raw data without human intervention and introducing errors. Metadata generation could also utilize machine learning and natural language processing to complete the metadata.

For machine learning provenance tracking, graphs may be a solution to explore. Graph-like databases may be a way to connect versioning for models, outputs, and lifecycle provenance without the limitations of SQL data storage, and it may also be a means to connect metadata across institutions.

TIMELINESS *(or maturity: Why now? What breakthrough or change makes progress possible now where it wasn't possible before? What will be the impact of success?)*

ASCR has long supported cutting-edge research in computing that supports DOE mission areas. Continued expansion of high-performance computing and scientific workflow development necessitates that there be research into effective, secure metadata management systems. In addition, advances in AI and the growing use of machine learning and distributed data sources requires focused effort in unified metadata infrastructure that cross cuts domains across the DOE facilities.

Other organizations are already moving in the direction of centralized metadata storage. The CEDAR group[2], for examples, collects metadata for medical record cross-reference and is accessible by anyone.

REFERENCES

1) EMSL: https://www.emsl.pnnl.gov/
2) ARM: https://www.arm.gov
3) Other work: CEDAR metadata repository: Open science and fair data, https://metadatacenter.org

Title: ASCR Compute Facilities Should Promote Sharing of FAIR Research Data by Publishing Data Catalogs

Authors: Annette Greiner (NERSC, amgreiner@lbl.gov), Lisa Gerhardt (NERSC, lgerhardt@lbl.gov), Gabor Torok (NERSC, gtorok@lbl.gov), Nicholas Balthaser (NERSC, nabalthaser@lbl.gov), Kirill Lozinskiy (NERSC, klozinskiy@lbl.gov), Shreyas Cholia (NERSC, scholia@lbl.gov)

Topic: Metadata management infrastructure to support FAIR principles

Challenge: DOE computing centers host petabytes and petabytes of data that could be reused by researchers both within the DOE complex and outside it, if it were only readily findable. That data sits in multiple storage tiers reachable only by its owners, or in disparate science gateways, each known only by a small community. Much of this data the centers themselves know little about besides permissions, timestamps, and its footprint on the file system; and thus DOE itself knows less than it might about this important work product of the research it funds. Data management plans [1] and the FAIR principles [2] begin to address these issues, but making data available to the public in a way that others will actually use requires extra effort on the part of investigators that they may not see as the best use of their time.

Opportunity: We need to tip the balance of how researchers value the FAIRification of data. ASCR compute facilities can raise the perceived value and lower the work barriers by publishing FAIR data catalogs that rely on shared metadata standards. Appropriate metadata standards and online data catalog software already exist, but some innovation would be needed to be successful. New developments would include automated collection of metadata and provenance information, APIs for FAIR-data-related services, and file transfer tools to enable non-HPC users to access large datasets.

Standardization facilitates automation, enabling the creation of tools that minimize human effort to prepare data. Standards also make data harvestable by popular data search engines. Enabling sharing at the center level, if done with attention to good metadata and standardized data sharing techniques, can feed information upward to services such as Data.Gov, Google Datasets, and DOE Data Explorer, increasing the incentive to make data shareable by giving researchers a broader audience.

Since the computing center is the point of interaction with researchers and their data, the center is the logical unit to promote and facilitate sharing of data by providing a center-wide data catalog. Center-level automation can make the task of contributing much easier, lowering the barrier of work for researchers. User accounting systems can be leveraged to provide a simple tagging mechanism to move a dataset into the published state, automating addition of metadata to a publicly accessible, searchable, and harvestable catalog. The system could include assigning persistent identifiers, automatically checking for dead links, enabling the tracking of downloads, and collecting actionable feedback from reusers (i.e., what made the dataset more or less usable, how it could be improved).

Though much can be automated, we recognize that some tasks can only be accomplished by the researchers whose data is in question. Only they know which sets of their data merit public sharing, and only they know the full context of the data. NERSC has been exploring additional options for incentivizing users to curate their data and provide good metadata, including starter templates for data management plans and for dataset metadata, quota exceptions for FAIR datasets, and enhancing the perceived value of data by tracking downloads and treating datasets like publications.

The file systems that store shared data also need to support public file sharing and simple, rapid downloads. The primary storage tier for most centers is not optimized for public consumption, and

probably shouldn't be. We need to find ways of simplifying or obviating file movement to publicly available file systems. Also, allowing users who are unfamiliar with the HPC ecosystem to download data without installing special software or having access to a center's command line would facilitate sharing across disciplines.

In addition to agreement on metadata standards among computing centers, success in creating these data catalogs will depend on ASCR support to facilities for long-term dataset storage, development of automated metadata-generating systems, and development of FAIR-data-handling services (e.g., researcher portals for curating data and public tools for downloading datasets). Through suitable policies and allocations, DOE can leverage FAIR data to maximize the value of the computational work it funds.

Timeliness: The inclusion in this call for papers of a facet mentioning FAIR data principles, along with an expansion of the abbreviation for those unfamiliar with it, testifies at once to both the importance that DOE now places on FAIR data and the relative newness of the concept to the community. The principles were first articulated in 2016 [3], but the most recent years since then have seen an uptick in interest in the DOE scientific computing community. For example, just over a year ago, the DOE announced funding for FAIR data for artificial intelligence [4]. Earlier this year, the Office of Science announced the PuRe data initiative, which centers FAIR data principles [5,6]. Clearly DOE has FAIR data on its mind. So do scientists, who are interested in ensuring the reproducibility of research as well as reusing data from previous research in new ways. At the same time, we are seeing increasing use of microservices at HPC centers, such as the Petrel data service [7], myOLCF [8], and the Superfacility API [9]. Thus, the building blocks of such a system can readily be integrated into existing service platforms.

References:

[1] U.S. Department of Energy, *DOE Policy for Digital Research Data Management*. https://www.energy.gov/datamanagement/doe-policy-digital-research-data-management, accessed 10 December 2021.

[2] GoFAIR Initiative, FAIR Principles. https://www.go-fair.org/fair-principles/, accessed 7 December 2021.

[3] Wilkinson, M., Dumontier, M., Aalbersberg, I. et al. "The FAIR Guiding Principles for scientific data management and stewardship." *Sci Data 3*, 160018 (2016). https://doi.org/10.1038/sdata.2016.18.

[4] U.S. Department of Energy, *Department of Energy Announces $8.5 Million for FAIR Data to Advance Artificial Intelligence for Science*, 10 August 2020. https://www.energy.gov/articles/department-energy-announces-85-million-fair-data-advance-artificial-intelligence-science.

[5] U.S. Department of Energy, "Introducing SC Public Reusable Research (PuRe) data resources," *US Department of Energy Science News*, 13 April 2021. https://ekaprdweb01.eurekalert.org/features/doe/2021-04/ddoe-isp041321.php.

[6] U.S. Department of Energy, *Public Reusable Research (PuRe) Data*. https://science.osti.gov/Initiatives/PuRe-Data, Accessed 7 December 2021.

[7] Allcock, William E. et al., Petrel: "A Programmatically Accessible Research Data Service." PEARC '19, July 28-August 1, 2019, Chicago, IL, USA. https://www.globusworld.org/files/2019/190728_Petrel_Programmatically_Accessible_Research_Data_Service.pdf.

[8] McDowell, Rachel, *OLCF Launches New User Portal, MyOLCF,* 10 November 2020. https://www.olcf.ornl.gov/2020/11/10/olcf-launches-new-user-portal-myolcf/.

[9] Bard D.J. et al. (2021) "Automation for Data-Driven Research with the NERSC Superfacility API." In: Jagode H., Anzt H., Ltaief H., Luszczek P. (eds) *High Performance Computing*. ISC High Performance 2021. Lecture Notes in Computer Science, vol 12761. Springer, Cham. https://doi.org/10.1007/978-3-030-90539-2_22.

Accelerating Data Processing at the Edge with Extreme Specialization

Data-management support for AI and complex workflows

ANTONINO TUMEO, MARCO MINUTOLI, VITO GIOVANNI CASTELLANA, ANKUR LIMAYE, CHENG TAN, ISMET DAGLI, NICOLAS BOHM AGOSTINI, SERENA CURZEL, VINAY AMATYA, and JOSEPH MANZANO, Pacific Northwest National Laboratory, USA

Challenges and Current Approaches. The US Department of Energy (DOE) operates the largest collection of experimental scientific instruments in the world for a variety of scientific disciplines. Each discipline's specific instruments and computing requirements introduce unique design challenges for addressing the data management and processing required by the application. The vision for the next generation of scientific instruments is that they will produce higher volumes of data at a higher velocity [1] and are expected to perform computing and learning tasks at the edge in order to automate experiments and/or assist domain scientists. Redesigning the computer hardware and software systems that are capable of supporting the data-management aspects required to build tightly integrated and intelligent scientific instruments will be fundamental to maintain DOE's leadership in scientific discovery.

The advent of the Internet-of-Things (IoT) has led the microelectronics research community to explore efficient, fast, scalable, reliable, and secure edge computing devices that are able to process data in real time at the source or near to the user. However, only the high-volume commercial edge computing devices can justify the high non-recurrent engineering costs for their development. These devices also have to collect and manage significantly lower amount of data, and are typically specialized for one or just a few specific use cases. Experimental instruments, such as electron microscopes, particle accelerators, mass spectrometers, and more, may need to set up a variety of specific experimental workflows to collect different types of data, at different times, and in some cases with limited opportunities to repeat the experiments. The sheer amount of data collected may not allow in situ processing, so methods to efficiently store, reduce, and move the data to a large-scale system for analysis and simulation are required. While some partial solutions may be achieved by combining existing high-performance embedded devices with reconfigurable hardware, they may not be able to address all the unique requirements of scientific instruments.

Opportunities. We believe the future of compute within DOE experimental facilities lies in a division of processing. At the experimental instrument or sensor, compute/data analysis will be present, providing high-speed online processing, conditioning, and extraction of features as the data is captured. Experimental steering will be possible through results learned during online processing leveraging custom accelerators and by providing feedback to sensors on where best to focus their data gathering. Such an approach opens new research opportunities – most notably, how best to partition the processing between edge-based and data center-based devices, how best to design and optimize each device for performance, power and cost, and finally, how to interconnect these elements to deliver the highest possible aggregate performance to instrument users while retaining flexibility.

To enable this vision, we believe that an opensource, modular, extensible, multilevel compiler toolchain to enable hardware/software codesign and agile end-to-end generation of domain specific system is required. Enabling domain scientists to move form novel algorithmic formulation to the implementation of a system with dedicated accelerators, exploiting either reconfigurable logic (e.g. Field Programmable Gate Arrays) or application specific integrated circuits (ASICs) without the assistance of a team of hardware designers, offers unique opportunities to accelerate the data processing and management pipelines from the scientific instruments to scientific discovery. A modular and multilevel compiler infrastructure [2] allows interfacing with the productivity tools adopted by domain scientists. They are

Corresponding Author: Antonino Tumeo; Authors' addresses: name.surname@pnnl.gov.

critical to initiate architecture independent optimizations and design space exploration of the generated systems as early as possible, to maximize the benefit of user-provided information. Dedicated hardware generation engines based on existing opensource compiler technologies can today leverage a richness of algorithmic solutions to generate highly optimized circuit designs, especially in the case of Finite State Machines with Datapath (FSMD). Interfacing with novel higher-level compiler infrastructures with their natural support for hierarchy and (task level, coarse grained) parallelism, opens further opportunities in generating and composing hierarchical hardware systems. The hardware generations process benefits from the availability of opensource or licensable hardware intellectural properties (IPs), which can beccome part of the resource library for such compiler-based toolchains, enabling algorithmic and hierarchical system-level design. This not only accounts for opensource instructions sets (such as RISC-V) but also templated accelerators [3] or even functional units. Compiler-based generators enable exploring the design space and setting parameters for these components (e.g., precision). Hence, they directly tie to the configurability of templated components. Additionally, they provide a path to supporting dynamic reconfigurability, for example, leveraging just-in-time compilation, where the intermediate representation (bit code) can be lowered to slightly different machine code depending on the overall system status and adapt dynamically to the experimental workflows.

Timeliness. DOE's scientific instruments present unique challenges. Some target very specific edge use cases, others are connected to large-scale instruments. In all cases, the ability to process multi-modal high-bandwidth data streams in real time, perform data compression and management, and identify key data points of actual value, is critical. As sensors evolve, the amount of collected data will explode. Only highly specialized, custom systems might satisfy these unique, contrasting requirements, but the complexity, and ultimately the costs, associated with their design, from software to hardware implementation, are too high. Despite the advances in tools and architectures, complexities and costs continue to rise. Industry alone cannot satisfy the needs of DOE's scientific instruments. Addressing these needs can only happen by leveraging community efforts to build adequate, end-to-end tools to enable the automated generation of specialized systems. The emergence of opensource compiler-based design automation tools, opensource hardware, chiplet based designs, establishes a unique context ripe for new research and investments to empower domain scientists with methods to support complex, data intensive experimental workflows.

Conclusion. We have identified needs for the co-design of efficient edge-computing and high-performance data processing facilities for DOE's intelligent scientific instruments. We have argued that to address these unique needs of these instruments we need new end-to-end design automation tools able to generate application-specific accelerators from high-level productive programming frameworks. In particular, we have highlighted the impact of these tools to generate hyper-specialized systems able to deal with the volume, velocity, variety, veracity, and value (big 5 Vs) of data provided by new sensors in scientific instruments. We believe that it will be possible to address these key data management needs only by leveraging a modular, end-to-end, compiler-based agile hardware design toolchain able to generate highly specialized data analytics and artificial intelligence accelerators.

REFERENCES

[1] E. Wes Bethel, Martin Greenwald, Kerstin Kleese van Dam, Manish Parashar, Stefan M. Wild, and H. Steven Wiley. 2016. Management, analysis, and visualization of experimental and observational data - The convergence of data and computing. In *12th IEEE International Conference on e-Science, e-Science 2016, Baltimore, MD, USA, October 23-27, 2016*. IEEE Computer Society, 213–222. https://doi.org/10.1109/eScience.2016.7870902

[2] Chris Lattner, Jacques A. Pienaar, Mehdi Amini, Uday Bondhugula, River Riddle, Albert Cohen, Tatiana Shpeisman, Andy Davis, Nicolas Vasilache, and Oleksandr Zinenko. 2020. MLIR: A Compiler Infrastructure for the End of Moore's Law. *CoRR* abs/2002.11054 (2020). arXiv:2002.11054 https://arxiv.org/abs/2002.11054

[3] Antonino Tumeo, Marco Minutoli, Vito Giovanni Castellana, Joseph B. Manzano, Vinay Amatya, David Brooks, and Gu-Yeon Wei. 2020. Invited: Software Defined Accelerators From Learning Tools Environment. In *57th ACM/IEEE Design Automation Conference, DAC 2020, San Francisco, CA, USA, July 20-24, 2020*. IEEE, 1–6. https://doi.org/10.1109/DAC18072.2020.9218489

Title: Minimizing Latency in High Performance Computers

Authors: (1) Branislav Radovanovic, Balex Technologies, LLC, bradovanovic@balextech.com and (2) Dr. Martin Perlmutter, Balex Technologies, LLC, map@balextech.com.

Topic: Storage-System Architecture Design.

Challenge: Latency is the single biggest performance inhibitor for high performance computing, effectively limiting the advance of scientific and technical discovery. Current storage systems rely on hard disk drives (HDDs) and solid-state drives (SSDs), and while these components have worked well over the last few decades, they have reached the limit of their capacity to handle modern I/O intensive applications, resulting in latency becoming untenable.

We believe that the minimization of latency will require the greater utilization of DRAM in storage system architecture design. DRAM is orders of magnitude faster than either HDDs or SSDs. The greater utilization of DRAM, however, creates challenges that need to be overcome including persistence, data recoverability, scalability, distributed caching, efficient data transfer, compatibility, and affordability.

Opportunity: Utilize DRAM-based storage system architecture for data transfer. Such new storage-system architecture may include one or more of the following elements:

- Persistence – The incorporation of battery back-ups to enable the orderly transfer of data from DRAM to persistent storage in the event of power loss.
- Data Recovery – The mirroring of data to prevent data loss in the event of component failure.
- Scalability – The linear increase in performance as the number of nodes in the system increases.
- Distributed Caching – The incorporation of distributed “RAM Disks” that can be accessed by all the processors in a multi-core computer system.
- Efficient Data Transfer Methods – The incorporation of efficient data transfer methods such as remote direct memory access (RDMA) and block storage.
- Compatibility – The compatibility with existing SCSI and NVMe compliant computer hardware and software protocols to avoid the need to rewrite existing software.
- Affordability – The reduction of cost as measured on an Input/Output Operations Per Second (IOPS) basis.

Timeliness/Maturity: The minimization of latency is a much higher priority now than in the past because of exponential growth of the data volume needing to be processed in a timely and cost-effective manner to solve an ever-expanding list of complex problems. Latency has grown into a significant problem because improvements in processor speed, parallelization and bandwidth have far outpaced investment and improvements in the data transfer systems between processors and storage. Now, without an improvement in data transfer rates, further improvements in processor speeds will, at best, only have a marginal impact on computer performance.

The impact of minimizing latency will be broad. A solution is critical to most scientific, commercial, and national security fields. In fact, the HPC user communities in these fields have been asking for these improvements for many years, recognizing that they must simplify code, dumb down input parameters, and reduce resolution for their software to run in a timely and useful manner. Indeed, the Department of Energy DOE has recognized that achieving exascale performance will be a multi-faceted effort encompassing, among other things, applications, system software, hardware technologies and architectures. A critical step to achieving exascale performance is increasing I/O speeds and reducing latency.

References:

The following are several articles discussing the importance of I/O latency:

Hill, Vince. “What’s Needed for High Fidelity, Low Latency HPC Network Monitoring at 100Gbps.” Inside HPC. 2021.

Grider, Gary. “HPC Storage and I/O Trends and Workflows.” Open Fabrics Alliance 12th Annual Workshop. 2016.

Luu, H., et. al. “A Multiplatform Study of I/O Behavior on Petascale Supercomputers.” The ACM International Symposium on High-Performance Parallel and Distributed Computing (HPDC), 2015.

Taneja, Arun. “I/O Performance in Need of a Fix.” *SearchStorage*, TechTarget, 8 Nov. 2011.

Lang, Samuel, et al. “I/O Performance Challenges at Leadership Scale.” Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis - SC '09, 2009.

Storage System Performance Improvement Using Science and System Metadata

Bryan Hess, bhess@jlab.org

Thomas Jefferson National Accelerator Facility

12000 Jefferson Avenue, Newport News VA 23606

Topic

Storage System performance improvement using AI with science-domain metadata and system performance information

Challenge

Storage Systems for scientific data storage are internet-scale, spanning institutional boundaries to deliver data to endpoints that are both batch and interactive. Policy-driven configuration is frequently used to automate data and metadata replication.

In addition to data storage itself, metadata is often stored in catalogs that are distinct from the filesystem. Information about the performance of file movement, network health, and storage systems performance are found in software systems including workload managers, file transfer clients, and the network control plane itself.

Characterizing system performance and understanding performance problems can require data from all these software components. Expertise in the science domain, system administration, network engineering, and software architecture are combined to draw conclusions and system performance trends.

Opportunity

There is an opportunity for an AI-based storage system capability that integrates information from these diverse software components to inform decisions about data movement beyond policy driven mechanisms. Such a capability would use AI models to identify anomalous file movement that reveal new usage patterns, suggest system degradation, or reveal overcommitment of resources. The incorporation of science metadata *and* system performance data combines expertise from multiple domains.

Examples of software components that can furnish performance information (such as latency, data rates, patterns) include Storage Elements, file systems, and tape storage systems. Time series data from Network Monitoring and measurement tools, such as perfSONAR, and network management systems

quantify network capabilities, performance, and health. Examining performance data from all aspects of file movement characterize the end-to-end performance. Inclusion of metadata has as its aim to make inferences from the science-driven patterns.

Opportunities for an AI system that incorporates system and domain metadata information might include:

1. Detection of poor performance or threats to system performance prior to human notice. Actions to test the model might include applying QoS limits to under-performing workloads to limit drag on the system.
2. Anticipate workloads, and schedule proactive data movement. This pre-staging would exercise systems in advance of the workload, highlighting system and network problems before they are consequential.
3. increase overall availability and protection of data that it rated as important based on metadata, access patterns, and overall performance by proactively creating replicas.
4. Suggest adaptive possibilities, *e.g.* is it better to stream vs stage a file for a particular workload? Is it quicker to replicate from B to C rather than from A to C based on end-to-end performance observation?

Timeliness

The maturity of community-supported software tools like XRootD, CVMFS, Rucio, LibreNMS, and PerfSONAR provide a mature toolset for building storage systems and instrumenting their performance at all layers, from the filesystem to the network. Each of these tools is a source of high-quality data and metadata that can provide deep insight into complex workflows.

References

1. https://xrootd.slac.stanford.edu/
2. https://rucio.cern.ch/
3. https://eos-web.web.cern.ch/eos-web/
4. https://www.perfsonar.net/
5. https://www.librenms.org/

Interfaces Supporting Data Management in Complex In transit Processing Workflows on Heterogeneous Systems with Deep Memory Hierarchies

B. Loring*, E.W. Bethel, K.J. Wu. Lawrence Berkeley National Lab

1 Topics

Data-management support for AI and complex workflows.

2 Introduction

As our ability to produce data has and continues to out pace our ability to store it, analysis has moved from the realm of post-processing into the realms of *in situ* and *in transit* processing where data is processed as it is produced. Figure 1 shows an *in transit*, workflow that is being used with the WarpX laser plasma accelerator(LPA) simulation code[10]. The analysis detects the laser envelope, a feature which encloses the particles undergoing acceleration that will ultimately form the beam. A data reduction is realized by only writing the accelerated particles to disk. *In transit* methods are employed here in order to decouple the simulation from the analysis code which depends on a parallel global FFT, which requires the data to be laid out in memory in a different decomposition than is used by the simulation, and has different scaling characteristics than the WarpX particle in cell(PIC) solver[2]. Once the laser envelope has been identified a representation is sent back to the simulation process group where it is used to down select the particles for tracking and to write to disk for further analysis.

This analysis use case is emblematic of some of the challenges inherent in complex *in transit* workflows where simulation data is moved to a separate process group or application for analysis. Often the analysis code needs to reorganize the data in flight to account for different sized jobs or specific domain decomposition requirements. Metadata and control information need to be exchanged prior to the movement of simulation data in order to make decisions about how to execute the data reorganization and orchestrate the buffering and communication of the data. However, existing I/O library interfaces lack means for the bidirectional exchange of control and metadata requiring external implementations. Additionally, specific addressability of the full memory hierarchy is needed in order to put the data directly into the most effective location for analysis. For instance, the data could be moved directly into the memory bank of an attached accelerator assigned to the analysis job via RDMA transfer.

3 Background

Often *in situ* processing through techniques such as dynamic steering and compression realizes data reductions enabling the storage of higher temporal fidelity data than would be possible before. A related approach called in-transit processing, where data is moved as it is produced to a separate application running simultaneously on a distinct set of hardware resources, is often used to couple simulation and analysis codes with different scaling characteristics and/or to couple analysis codes with specific data access and execution requirements that differ from each other[5, 6]. *In transit* processing requires the movement of large amounts of data between applications running on distinct sets of nodes. In addition to purely network based, "over the wire", approaches[9] for moving data from the producer to the consumer, high level I/O libraries with support for self describing layouts and rich metadata have added explicit in memory data movement capabilities[3].

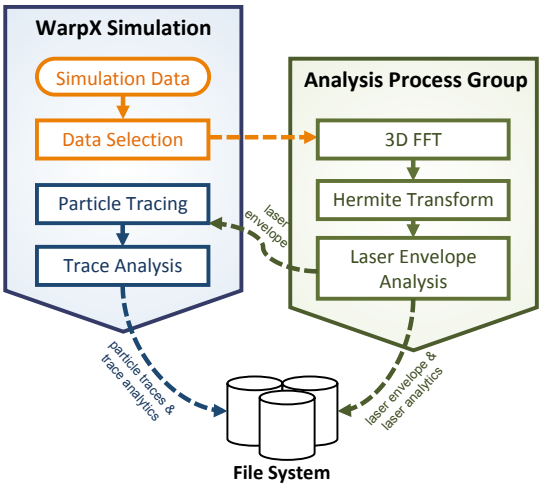


Figure 1: A complex bidirectional *in transit* workflow. Data is moved from the WarpX LPA simulation to a separate process group where the laser envelop is detected. A geometric representation of the envelop is sent back to the simulation where it is used to identify particles being accelerated. A large data reduction is realized when only accelerated particles are written to disk for post processing enabling higher fidelity output than there otherwise would be I/O budget for.

Generic *in situ* seeks to couple any data producer to any data consumer through a single API such that consumers of data may be swapped at run time via a simple configuration mechanism [1]. Generic *in transit* leverages metadata rich, self describing data streams to improve interoperability between codes and enable application specific dynamic steering and routing of data improving overall efficiency[6].

4 Challenges

The proliferation in accelerator based systems and the deeper memory hierarchy that result pose significant challenges and opportunities for effective *in transit* processing. In addition to a number of CPUs and memory banks, modern HPC systems are equipped with a number of additional on node accelerators, each with its own memory hierarchy. Computational systems are also being deployed with ever deeper memory hierarchies incorporating technologies such as local SSDs, burst buffers, and run time configurable NVM. Recent efforts have shown that directly addressing GPU memory in I/O operations can improve performance [7] and the ability to directly address SSD based burst buffers can outperform in memory data transfers [4].

*corresponding author: bloring@lbl.gov

However, the interfaces provided by traditional I/O libraries, even those that enable configurable "over the wire" data movement have not kept pace with the rapidly evolving technological landscape. The traditional path based interfaces for addressing the memory hierarchy is cumbersome and incomplete. The approach of treating new memory levels in the hierarchy as file system caches is useful but not sufficient to meet the needs of the complex work flows that arise in *in transit* processing scenarios. Ideally I/O interfaces would be expanded to provide fine grained point to point addressing enabling *in transit* work flows to move data directly to and from specific locations in each level, in each memory hierarchy, attached to each node.

In transit workflows often need to orchestrate complex data reorganizations as data is moved from the simulation to the analysis application either to accommodate different scaling characteristics in the analysis or to accommodate specific domain decomposition requirements of the analysis, or both. Efficiencies in the dynamic repartitioning of simulation data can be realized through the bi-directional inter application exchange of metadata and control information to prepare for and execute large bulk inter application data movement [6]. However, existing high level I/O and data movement libraries lack explicit interfaces for the the bidirectional exchange of control information and metadata leaving practitioners to implement ad hoc solutions.

5 Opportunities

Redesigned and expanded interfaces to high level I/O libraries could provide better access to all levels and locations in the system memory hierarchy for the purpose of moving data between codes running on distinct hardware resources on the system. The interfaces could build upon the existing implementation for the creation and use of metadata rich self describing streams, but could be redesigned and expanded to provide more support for inter application point to point communication between specific resource locations, and expanded to include support for bi-directional inter application communication of metadata for steering, and execution and flow control.

Such a revisions to I/O library interfaces could be the foundation for the more effective use of *in transit* processing on current and future systems with accelerators and deeper memory hierarchies. The improved interfaces would facilitate run time decisions about how best to dynamically allocate and use the extended memory hierarchy and orchestrate movement of data directly to the desired location leading to improved throughput and enabling better science.

6 Timeliness

The proliferation of accelerator based systems, and accompanying higher computational throughput, has further increased the growing gap in our ability to produce data relative to our ability to store it leading to the continued need for *in situ* and *in transit* processing. The addition of accelerators to HPC systems has made their on node memory hierarchies more complex. At the same time the on node and system wide memory hierarchy has been expanded to include local SSDs, burst buffers, and NVM making both the on node and system wide memory hierarchy more complex. As we approach the limits of Moore's law modern HPC systems equipped with accelerators and deeper memory hierarchies will likely continue to play an important role in HPC[8]

New and expanded interfaces to high level I/O libraries that include more inter application communications capabilities such as fine grained addressability of the available memory resources and locations in the system; and new and expanded interfaces for the exchange of control information and metadata are needed to fully realize the potential for *in situ* and *in transit* processing on current and future systems.

References

[1] U. Ayachit, B. Whitlock, M. Wolf, B. Loring, B. Geveci, D. Lonie, and E. W. Bethel. The SENSEI Generic In Situ Interface. In *Proceedings of In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization (ISAV 2016)*, Nov. 2016. LBNL-1007263.

[2] L. Dalcin, M. Mortensen, and D. E. Keyes. Fast parallel multidimensional fft using advanced mpi, 2018.

[3] W. F. Godoy, N. Podhorszki, R. Wang, C. Atkins, G. Eisenhauer, J. Gu, P. Davis, J. Choi, K. Germaschewski, K. Huck, A. Huebl, M. Kim, J. Kress, T. Kurc, Q. Liu, J. Logan, K. Mehta, G. Ostrouchov, M. Parashar, F. Poeschel, D. Pugmire, E. Suchyta, K. Takahashi, N. Thompson, S. Tsutsumi, L. Wan, M. Wolf, K. Wu, and S. Klasky. Adios 2: The adaptable input output system. a framework for high-performance data management. *SoftwareX*, 12:100561, 2020.

[4] J. Gu, B. Loring, K. Wu, and E. W. Bethel. Hdf5 as a vehicle for in transit data movement. In *Proceedings of the Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization*, ISAV '19, p. 39–43. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3364228.3364237

[5] J. Kress, M. Larsen, J. Choi, M. Kim, M. Wolf, N. Podhorszki, S. Klasky, H. Childs, and D. Pugmire. Comparing the efficiency of in situ visualization paradigms at scale. In M. Weiland, G. Juckeland, C. Trinitis, and P. Sadayappan, eds., *High Performance Computing*, pp. 99–117. Springer International Publishing, Cham, 2019.

[6] B. Loring, J. Gu, N. Ferrier, S. Rizzi, S. Shudler, J. Kress, J. Logan, M. Wolf, and E. W. Bethel. Improving performance of m-to-n processing and data redistribution in in transit analysis and visualization. In *EuroGraphics Symposium on Parallel Graphics and Visualization (EGPGV)*. Norrköping, Sweden, May 2020.

[7] J. Ravi, S. Byna, and Q. Koziol. Gpu direct i/o with hdf5. In *2020 IEEE/ACM Fifth International Parallel Data Systems Workshop (PDSW)*, pp. 28–33, 2020.

[8] J. Shalf. The future of computing beyond moore's law. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 378:20190061, 03 2020.

[9] W. Usher, S. Rizzi, I. Wald, J. Amstutz, J. Insley, V. Vishwanath, N. Ferrier, M. E. Papka, and V. Pascucci. Libis: A lightweight library for flexible in transit visualization. In *Proceedings of the Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization*, ISAV '18, p. 33–38. Association for Computing Machinery, New York, NY, USA, 2018.

[10] J.-L. Vay, A. Huebl, A. Almgren, L. D. Amorim, J. Bell, L. Fedeli, L. Ge, K. Gott, D. P. Grote, M. Hogan, R. Jambunathan, R. Lehe, A. Myers, C. Ng, M. Rowan, O. Shapoval, M. Thévenet, H. Vincenti, E. Yang, N. Zaïm, W. Zhang, Y. Zhao, and E. Zoni. Modeling of a chain of three plasma accelerator stages with the warpx electromagnetic pic code on gpus. *Physics of Plasmas*, 28(2):023105, 2021.

Revisiting Storage Programming Models

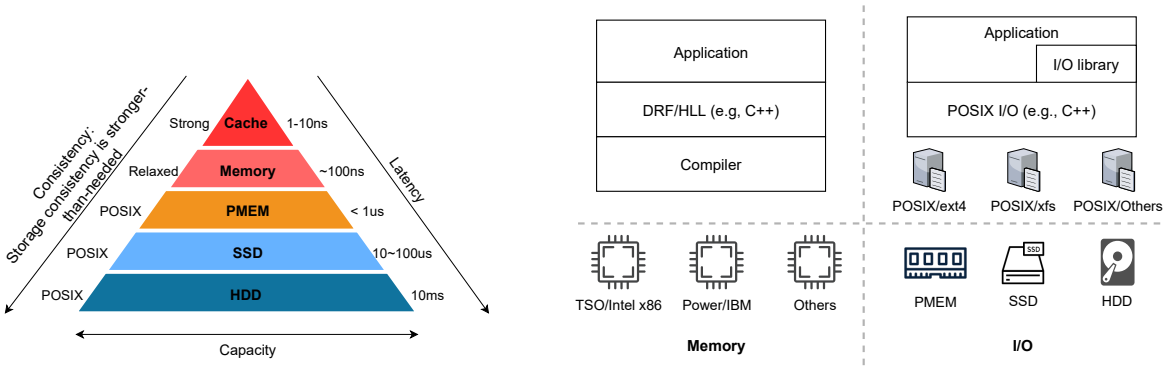
Chen Wang (chenw5@illinois.edu) and Marc Snir (snir@illinois.edu)
Department of Computer Science
University of Illinois at Urbana-Champaign

Topic

The use of the POSIX consistency model for I/O has plagued the HPC community for many years, but it is becoming more problematic due to two key reasons: (1) the rapid increase in the scale of HPC systems; (2) the emergence of the new storage techniques such as persistent memory. This problem can no longer be ignored especially as we move toward the exascale era. Even though POSIX consistency is the issue, the solution is not finding a better consistency model. The right solution is a paradigm shift away from the current consistency-centric I/O programming model to a synchronization-centric I/O programming model.

Challenge

There are two fundamental issues with the use of POSIX I/O for HPC: (1) It is overused; and (2) Its strict consistency model is a major bottleneck. The left edge of the pyramid below depicts the strictness of the consistency models provided at each storage level. One would expect that from top to bottom, the consistency model should be weaker and weaker as sharing becomes less frequent and access patterns become simpler. However, most storage systems only utilize the strong consistency model imposed by POSIX. Our previous study [4] has shown that HPC applications do not require the POSIX consistency model. A weaker consistency model can be used to improve performance without sacrificing programmability or portability.



(a) The most commonly used storage techniques

(b) Memory vs. I/O programming model

Figure 1: Memory vs. I/O

When it comes to the domain of memory, an important factor in the adoption of relaxed memory models is that compilers can hide the complexity. As shown in Figure 1(b), programmers target a single consistency model specified by the high-level programming language (e.g., C++ and Java) without the knowledge of underlying consistency models provided by the CPUs. In comparison, there is no corresponding "compiler" layer in the I/O programming model. General file systems (local or parallel) must provide the same consistency model (POSIX) regardless of the underlying storage hardware, which leads to unnecessarily reduced performance. Besides, the POSIX consistency is very expensive to maintain especially in distributed systems.

Opportunity

Several efforts [1, 2, 3, 5] have been made to alleviate the bottleneck caused by POSIX. These efforts propose to replace the POSIX PFSs with relaxed-semantics or tunable consistency PFSs as shown in Figure 2(a) and (b). Although these approaches have shown performance improvement, they are not long-term solutions to

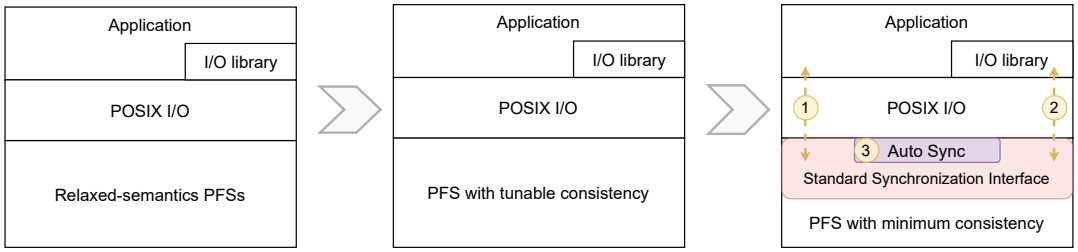


Figure 2: Both (a) and (b) are based on the current consistency-centric programming model, where PFSs are unaware of the application's synchronization logic and thus conservative consistency models have to be used. The proposed synchronization-centric programming model is shown in (c).

the fundamental issues of using the POSIX I/O model. The fact remains that there is no single consistency model that is optimal for all applications and hardware. Fundamentally, HPC I/O performance is hindered by the current consistency-centric programming model and we must change our approach.

We propose a shift away from the consistency-centric I/O model we use today in HPC to a synchronization-based I/O model, shown in Figure 2 (c). The core component is a standard synchronization interface. In contrast to the current programming model where file systems implement a standard consistency model (POSIX), here they implement a standard synchronization interface. File systems can provide whatever consistency models work the best for the underlying hardware, and HPC application programmers will utilize synchronization operations to ensure I/O correctness. We envision three ways to achieve the "proper synchronization" as denoted by the circled numbers in the figure:

1. Applications directly make use of the APIs provided by the standard synchronization interface. This should provide the best performance since application users know exactly where and when synchronization operations are needed. The drawback is that it requires modifications to the existing applications.
2. High-level I/O libraries or special synchronization libraries can provide an abstract layer to simplify the process of inserting synchronization operations. For example, they may allow annotations to existing function calls, e.g., marking a `close` function call as a synchronization operation. This method should be easier to use and can provide comparable performance to the first method.
3. Synchronization middleware can be designed to perform automatic synchronizations. The simplest implementation is to synchronize at every I/O operation. Both applications and high-level libraries can utilize this middleware. Since the application's I/O logic is unknown to the middleware, unnecessary synchronizations may be inserted which will lead to reduced performance. The advantage of this method is it requires little or no modification to the existing code.

We believe this new I/O programming model will have a significant impact on HPC systems and applications. We anticipate that many HPC applications will gain great performance improvement without any code changes. More importantly, the I/O bottleneck caused by the overuse of POSIX consistency can be addressed by this synchronization-centric programming model. With proper and accurate synchronizations, HPC applications should be ready to take advantage of future exascale storage systems.

References

[1] L. L. N. Laboratory. UnifyFS: A File System for Burst Buffers. `https://github.com/LLNL/UnifyFS`, Dec. 2021.

[2] O. Tatebe, S. Moriwake, and Y. Oyama. Gfarm/BB—Gfarm File System for Node-Local Burst Buffer. *Journal of Computer Science and Technology*, 35(1):61–71, 2020.

[3] M.-A. Vef, N. Moti, T. Süß, T. Tocci, R. Nou, A. Miranda, T. Cortes, and A. Brinkmann. GekkoFS: A Temporary Distributed File System for HPC Applications. In *2018 IEEE International Conference on Cluster Computing (CLUSTER)*, pages 319–324. IEEE, 2018.

[4] C. Wang, K. Mohror, and M. Snir. File System Semantics Requirements of HPC Applications. In *Proceedings of the 30th International Symposium on High-Performance Parallel and Distributed Computing (HPDC)*, pages 19–30, 2020.

[5] T. Wang, K. Mohror, A. Moody, W. Yu, and K. Sato. BurstFS: A Distributed Burst Buffer File System for Scientific Applications. In *The International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*, 2015.

Title: Disaggregating Storage to Meet the Needs of Integrated Facilities

Authors: Christopher Zimmer (zimmercj@ornl.gov), Sarp Oral (oralhs@ornl.gov), Christopher Brumgard (brumgardcd@ornl.gov) Oak Ridge National Laboratory

Topic:
Interfaces for accessing data. Data management and support for complex workflows

Challenge:
DOE user facilities house the world's largest supercomputers, billion-dollar scientific instruments such as the Spallation Neutron Source, and 3D manufacturing facilities. The growing demand to extract more science and increase the sum of the total value from these facilities has led to the desire to interconnect and use them in concert. An example of this is digital twins, modeling an experiment in real-time guiding manufacturing or directing an electron microscope.

In connecting facilities, the storage system is the natural intermediary. However, the parallel file systems of today are insufficient for this task. They are too rigid to meet the requirements for interoperability of emerging workflows due to their static nature of being pools of hardware and software. The motivation behind parallel file system development has primarily been to increase capacity and I/O performance while still maintaining a POSIX interface for applications. This has incurred not just significant fiscal costs, but comes at a cost to flexibility. These file systems fail to provide guarantees for interoperability, predictable performance, use case-derived and persistent capacity, security isolation, data reduction, filtering, compression, and visualization mechanisms. While the list of requirements is not exhaustive as we are still learning about these challenges, connecting billion-dollar facilities without accommodating these needs increases the risk of experimental failure that is both costly and time-consuming.

We argue that file and storage systems will have to change to meet our new demands. Instead, they will need to dynamically carve out and partition individual software and hardware components down to the individual storage elements used. In essence, we are arguing for dynamically composing exclusive file and storage systems out of shared resources. This presents several challenges, such as how to migrate data to free up resources, how does the shared file and storage manage data in such a way that best allows for the addition or removal of disks and servers, how to automate creating, managing, and tearing down project allocations via policy.

Opportunity:
As the term disaggregated computing has begun making its rounds within the DOE HPC compute cluster space, we argue that the concept of disaggregated storage should as well. We define disaggregated storage as an on-demand mechanism providing the constituent parts necessary to create dynamic composable file and storage systems tailored to the needs of integrated facility experiments. In disaggregated storage, the file and storage system is a pool of hardware resources managed by sophisticated and dynamic software that enables the ability to

carve use case-specific file and storage systems out of the pool of hardware resources. Disaggregated storage differs from commercial storage as a service (SaaS) concept in that both the vertical and horizontal stacks should be completely configurable. It should offer a provisionable software abstraction layer over a set of hardware resources and an entire isolated I/O subsystem including all necessary software and hardware pieces that are composable. In addition to reducing the OpEx and CapEx costs, we aim to provide a performance-driven and guaranteed file and storage system for interoperated scientific user facilities.

Timeliness and Maturity:
We are at a technology inflection point with the advent of DPUs, ARM server CPUs, NVMe over Fabric, NVIDIA DGX, and other supporting technologies. This view will change the composition of a parallel file system to include computation within a pool of servers and discs to support advanced data filtering and compression. Pools of disks usually dedicated to a single server or pair of servers are too expensive of a commodity to keep in reserve for partitioning. Instead, advanced headless storage systems managing disaggregated disks with network interfaces may become the fine-grained approach to support this work. The COVID pandemic has demonstrated the necessity of connected experimental ecosystems to enable rapid science progress during global crises. Flexible storage on demand is one part of the development of this ecosystem necessary to accommodate the varying requirements of a more connected world of science.

References:
Y. Zhu, W. Yu, B. Jiao, K. Mohror, A. Moody and F. Chowdhury, "Efficient User-Level Storage Disaggregation for Deep Learning," 2019 IEEE International Conference on Cluster Computing (CLUSTER), 2019, pp. 1-12, doi: 10.1109/CLUSTER.2019.8891023.

Vamsee Reddy Kommareddy, Simon David Hammond, Clayton Hughes, Ahmad Samih, and Amro Awad. 2019. Page migration support for disaggregated non-volatile memories. In Proceedings of the International Symposium on Memory Systems (MEMSYS '19). Association for Computing Machinery, New York, NY, USA, 417–427.
DOI:https://doi.org/10.1145/3357526.3357543

Shan, Yizhou, et al. "Legoos: A disseminated, distributed {OS} for hardware resource disaggregation." 13th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 18). 2018.

Nanavati, Mihir, Jake Wires, and Andrew Warfield. "Decibel: Isolation and sharing in disaggregated rack-scale storage." 14th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 17). 2017.

A Standards-Based Data Framework for Scalable, High Performance, Cross-Organizational Science

Dale V. Stansberry (stansberrydv@ornl.gov), Sean R. Wilkinson (wilkinsonsr@ornl.gov),
Olga A. Kuchar (kucharoa@ornl.gov)
Oak Ridge National Laboratory, Oak Ridge, TN

Topic

Definition and standardization of data-access interfaces and security models for cross-facility research.

Challenge

The current trends within modern scientific research of increasing data volume, variety, and velocity, coupled with a lack of standardized high-performance data access solutions, have led to significant burdens on researchers engaged in data-centric, cross-facility, and/or collaborative research efforts. While large-scale observational and computational facilities generally provide data infrastructure sufficient for localized research, there is a notable lack of support for researchers that need to share or utilize data and data streams across facility or organizational boundaries. Without uniform, performant, and secure data access technologies, researchers are forced to spend significant time contending with basic "data plumbing" issues instead of focusing on actual scientific research. While there are existing technologies and solutions that individually address subsets of the data access problem space, there are no comprehensive stakeholder-driven solutions that offer uniformity, performance, security, or maintainability.

GridFTP is a point-to-point data transfer protocol standardized by the Open Grid Forum [1] to specifically address the needs for reliable and efficient data transport for grid computing environments. Globus and the Globus Toolkit (GTK) [2] are specific GridFTP implementations developed by the Globus group within the University of Chicago which have become the de facto standard for data transport within the scientific community. While Globus has provided invaluable services to the scientific community for many years, ultimately DOE and ASCR have no direct control over this specific implementation, resulting in significant operational risks to future integrated facilities. For example, Globus dropped support for GTK (now the Grid Community Toolkit [3]) which was, and still is, widely used by the HPC community. From a technical perspective, Globus has evolved toward a web-centric model that does not directly support the requirements of most facilities running HPC or analytics environments (i.e. command line interfaces, scriptable APIs, unsupervised processes). In addition, GridFTP does not support data streaming or low-latency data transport for live experiments that require near-real-time analytics on remote computational resources.

Security is a critical aspect of data access, and with the push towards cross-facility and collaborative research, it becomes increasingly challenging to achieve system-wide security from a technical perspective. Existing security models either do not scale across many organizations or user accounts, or do not accommodate the unique constraints of HPC and scientific environments. Federated identity management is a step in the right direction, but the current approach is not suitable for higher security enclaves or for long-running unsupervised processes.

Opportunity

ASCR is in a unique position to support research into a data access framework that would be an enabling technology for large-scale, data-oriented, cross-facility science. Important research topics, such as ML/DL, are currently hindered by data rather than compute, and this problem is likely to become worse over time due to the current focus on computational performance and capacity over data storage and transport. While data access is only a part of the overall data problem, a scalable data-access framework would enable uniform, efficient, and secure transport and/or streaming of large volumes of data across organizational boundaries. Such a capability would thus enable the development of transformative distributed data systems for creating cross-facility workflows, data pipelines, and large-scale, FAIR-principled [4] scientific data management systems. The result of this data framework research would be a living, stakeholder-driven standard that would specify the architecture, requirements, and interfaces of a scalable data transport and streaming system. Future developers and vendors could then create compliant solutions for integration into the DOE/ASCR network of laboratories and facilities.

Research into a framework rather than specific technologies permits a degree of flexibility while still ensuring interoperability. A data access framework would define an abstract ecosystem of interfaces, security models, and quality of

service attributes for a variety of data utilization patterns, such as point-to-point transfers, streaming, and/or publish-subscribe. Such a data framework would provide uniform, high-level data access semantics while encapsulating lower-level technologies and protocols - enabling the adoption of future technology improvements without requiring rework to applications or the framework itself. A framework would also enable low-level protocol negotiation to ensure optimal and secure data exchange between sites with varying capabilities as well as providing a window of backward compatibility for long-term maintainability. In general, the key design objectives of an ideal data access framework would be interoperability, security, performance, scalability, and maintainability; however, DOE/ASCR oversight of the research and standardization process is essential given the significant investment required to develop and integrate new technologies into the large and highly complex systems managed by DOE/ASCR.

There are a number of modern technologies that could be evaluated for applicability, or inspiration, regarding specific technical aspects of a data framework. For example, it may be possible to adapt or tailor current federated identity management models to better work with the security and interface requirements of ASCR facilities. There are also a number of popular data transport, streaming, and pub/sub technologies that could be evaluated for use in a data framework, or as technical guidance for the development of new data technologies.

Timeliness

Heterogeneous models for HPC that employ accelerators such as GPUs and TPUs in tandem with traditional multicore CPUs have influenced not only the algorithms of HPC, but also the future design and locations of the computers themselves. Scientific campaigns can now collect such large amounts of data from observational facilities as to necessitate some level of processing at the data source itself – at the edge – before transferring the data to traditional HPC centers. This move to the edge will be accompanied by changes in the way scientific campaigns use automation and networks to get their work done. The current practice of utilizing different compute and data resources from different facilities will become increasingly common, making this the perfect time to begin creation of a framework to support those needs.

Moreover, modern technologies such as blockchain and smart certificates enable improved designs that were not possible when GridFTP and Globus were first designed. A promising approach lies within the Web 3.0 technologies augmented with newer networking methodologies. For example, BitTorrent was benchmarked by Sandia [5] and shown to be performant but could be more user-friendly. Random linear network coding technologies [6] associated with WiMAX protocols could overcome some of the networking overhead associated with TCP network acknowledgements resulting in greater throughput and improved error correction of transmissions. Similarly, technologies like IPFS [7] offer secure and distributed storage of resources over peer-to-peer networks. Building from the distributed nature of these synthesized technologies and hybridizing with mesh networking technologies such as Batman [8], multiple concurrent data transmission paths may be coordinated to route massive amounts of data to a destination. When security is a vital concern, authentication and resource discovery can be delegated to smart contracts on the blockchain. This capability would provide redundancy, authentication, transaction records (provenance), improved automation, and secure resource sharing.

References

[1] "Open Grid Forum." https://ogf.org

[2] "Globus - Research Data Simplified." https://www.globus.org

[3] "Grid Community Forum." https://gridcf.org

[4] "FAIR Principles." https://www.go-fair.org/fair-principles

[5] "An Evaluation of BitTorrent's Performance In HPC Environments." https://www.osti.gov/servlets/purl/1115054

[6] "Random Linear Network Coding on Programmable Switches." 2019 ACM/IEEE Symposium on Architectures for Networking and Communications Systems, ANCS 2019. Version: Original manuscript

[7] "IPFS powers the Distributed Web." https://ipfs.io

[8] "B.A.T.M.A.N. protocol concept." https://www.open-mesh.org/projects/open-mesh/wiki/BATMANConcept

Portable Persistent Services for Data Management in coupled HPC + ML/AI Workflows

Author: Daniel Laney, LLNL, laney1@llnl.gov
Topic: Portable HPC + Cloud infrastructures for data management

Challenge: persistent services to wrangle huge data volumes from coupled HPC + ML/AI workflows could be massive force multipliers yet create barriers to portability. We should invest in the continuing merging of traditional HPC with Cloud-based technologies, and work to enable a portability layer for workflows to leverage future merged compute environments.

The coupling of traditional High Performance Computing (HPC) with new simulation, analysis, and data science approaches provides unprecedented opportunities for discovery but also creates new application and infrastructure challenges. These new applications combining HPC and data science will require sophisticated infrastructures to manage data, including maintaining workflow state, cataloging simulation outputs, supporting dynamic and adaptive workflows, hosting training data for machine learning processes, and enabling data curation. Additional complexity arises if these workflows are coupled to experimental facilities. A key issue is that these infrastructures are often placed firmly in the hands of the application developers and are not supported by the system software stack or workflow management software. In these cases, application developers often adopt a model where all services are deployed within the same batch allocation as the compute and data science components of a workflow [1]. A downside of this approach is that it is explicitly tide to the batch system and its limitations, and the ability to elastically leverage compute resources is limited. For cases where a workflow system *does* handle data management and movement, there are additional barriers to installing and configuring the required services and servers leading to barriers to adoption, and a workflow being portable only to data centers where these barriers have been overcome.

It is likely, as we have seen in commercial ML/Analytics applications, that *persistent services* to manage these complex data-driven workflows will be crucial to ensure they are tractable to maintain and deploy, and scalable across a range of hardware. Enabling portability and performance of these services in blended HPC + ML/AI environments, particularly as HPC and Cloud technologies continue to merge, will greatly enhance the ability of teams to incorporate data management capabilities and build or leverage sophisticated workflow systems and infrastructures.

Opportunity: creating portable interfaces for persistent services in blended HPC + Cloud environments to support high performance exascale+ workflows that combine HPC, ML/AI, and experimental data sources will accelerate the ability of scientists to leverage these new hardware environments.

Several facilities are working toward programmable interfaces to HPC / data center resources (e.g., the NERSC Superfacility effort [2]). There exists at least one open-source effort to provide portable API's for cloud technologies [3]. A research effort focused on understanding how to

build on these ongoing efforts to create a portability layer for performant coupled workflows is needed. Key challenges for such an effort are:

- a systematic approach is needed to understand performance and bandwidth requirements of cloud + ML/Experimental workflows
- current approaches (e.g., on-prem Kubernetes environments) may not provide scale or bandwidth needed for tightly coupled HPC + ML/AI workloads and research may be required on enhancing these technologies or developing new approaches
- the ongoing blending of traditional HPC and Cloud technologies, and the rate of change in each space is high, orchestrating jobs and services is a key area of research and innovation
- security postures at facilities vary widely, and network capabilities and configurations are heterogeneous

Nevertheless, providing workflow and applications developers with portable API's to data management services, orchestration and messaging services, and related capabilities would enable adoption of these capabilities at a higher rate than we are seeing currently, and enhance the capability and portability of future HPC + ML/AI workflows.

Timeliness: workflows that integrate HPC, ML/AI, and/or experimental data streams are increasing in number and size, as evidenced by complex workflows deployed for the Gordon Bell Covid-19 competition at Supercomputing 2020 [4]. Workflows of this complexity will probably require site-specific configuration and optimization in the absence of good performing portable APIs to interface with persistent services providing Data Management, incurring an opportunity cost on application teams building and leveraging these workflows. Efforts to merge Cloud and traditional HPC are ongoing, and several facilities now boast on-prem Cloud environments in their data centers which provides fertile opportunities for co-design and experimentation. An effort towards portable services for data management and orchestration can leverage representative applications and the emerging environments at facilities to make forward progress and engage in co-design activities.

References:

1. H.Bhatia, F. D. Natale, J. Y. Moonet al., "Generalizable coordination of large multiscale workflow: Challenges and learnings at scale," The International Conference for High Performance Computing, 2021.
2. https://www.nersc.gov/research-and-development/superfacility/
3. https://libcloud.apache.org/
4. https://www.exascaleproject.org/workflow-technologies-impact-sc20-gordon-bell-covid-19-award-winner-and-two-of-the-three-finalists/

This work was performed under the auspices of the U.S. Depart-ment of Energy by Lawrence Livermore National Laboratory under ContractDE-AC52-07NA27344 (LLNL-CONF-826133)

Storage-system architecture design
A Time based Streaming Data Storage and Management

David A. Bader, Qing Liu and Zhihui Du
New Jersey Institute of Technology
Newark, New Jersey 07102
Email: {bader, qliu, zhihui.du}@njit.edu

1. Challenges

Many large scientific facilities will constantly generate huge amounts of streaming data. For example, LSST (Legacy Survey of Space and Time) [1], [2] is the successor to a long tradition of sky surveys and its camera is expected to take over 200,000 pictures (1.28 petabytes uncompressed) per year, far more than can be reviewed by humans. Managing and effectively analyzing the enormous output of the telescope is expected to be the most technically difficult part of the project. At the same time, more and more emerging applications, such as social networks, cybersecurity and bioinformatics, also have an increasing amount of streaming data [3]. These applications motivate the challenging problem of designing a novel storage-system architecture to efficiently support different scientific workflows [4], [5], [6].

The scientific workflows can be divided into three types, streaming workflow for the current in-motion data, learning and mining workflow for historical data, and archiving workflow for long time archived data. The challenges for such streaming data storage and management architecture are as follows: (1) the basic data block management should be flexible because scientific workflows often have different data access patterns; (2) the data storage and management should support all workflows well instead of just one kind of workflow. Since time is a critical factor to design and develop the storage and management architecture for the broad set of scientific workflows, we propose a novel *time based storage and management architecture* to attack the challenges of a lifecycle data management [7], [8].

2. Opportunity

The basic idea of the proposed storage and management architecture is given in Fig. 1. It is a time based evolving architecture. This means that how data are organized and managed is highly related with the timestamp of the data. The data block size, the data storage position and data access method will change with time.

For any streaming workflow, the real-time performance is a critical requirement so the data management should provide fine grained random data access and only a small number of data in a limited time window are available. Online query applications can use an in-memory database [9] to improve the data access performance. To further improve the distributed processing performance, the streaming data with the same timestamp can often be partitioned into distributed memory so they can be processed in a parallel fashion.

However, the online streaming workflow often needs models and statistical data generated from historical data. The learning and mining workflows can provide such information. The historical data block will often be organized based on hours, days, weeks or even months. At the same time, the data with the same or close position will be stored closely. In such workflows, we often find that more recent data will often be accessed with high frequency. So the data closer to current time will often be organized as smaller blocks. At the same time, the metadata about different blocks will also be generated. Taking advantage of the metadata and the variable size data block, machine learning and data mining workflows often can achieve high data access throughput and high performance.

Archiving the data and related programs are necessary for scientific applications. They are often IO intensive applications and data compression/reduction are often needed to improve the performance.

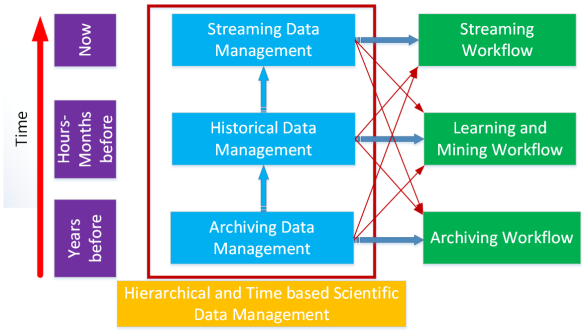


Figure 1: A lifecycle scientific streaming data storage and management architecture

Fig. 1 shows that when we build a data storage and management architecture and it can evolve with time and can support different kinds of workflows well during the lifecycle of given streaming data. This architecture will cover the complete hierarchical storage levels. The size of data block and management methods at different levels can be

very different to support different kinds of workflows. The novel storage media, such as SSD devices [10], non-volatile memory (NVM) cards [11], provide us the opportunity to design a novel hierarchical and integrated storage system.

3. Timeliness or maturity

When large scientific facilities, sensors and network are widely deployed, more applications will generate streaming data. At the same time, the novel storage devices also provide us the media to store streaming data at their different lifecycle stages. So a novel data storage and management system is necessary to efficiently support different kinds of streaming workflows. The success of the novel streaming data storage and management architecture will directly support different kinds of scientific workflows and explore the value of streaming data in their lifecycle.

References

[1] R. L. Jones, M. T. Bannister, B. T. Bolin, C. O. Chandler, S. R. Chesley, S. Eggl, S. Greenstreet, T. R. Holt, H. H. Hsieh, Z. Ivezić *et al.*, "The scientific impact of the Vera C. Rubin observatory's legacy survey of space and time (LSST) for solar system science," *arXiv preprint arXiv:2009.07653*, 2020.

[2] A. Siraj and A. Loeb, "Searching for black holes in the outer solar system with LSST," *The Astrophysical Journal Letters*, vol. 898, no. 1, p. L4, 2020.

[3] A. McGregor, "Graph stream algorithms: a survey," *ACM SIGMOD Record*, vol. 43, no. 1, pp. 9–20, 2014.

[4] S. Bharathi, A. Chervenak, E. Deelman, G. Mehta, M.-H. Su, and K. Vahi, "Characterization of scientific workflows," in *2008 third workshop on workflows in support of large-scale science*. IEEE, 2008, pp. 1–10.

[5] Y. Gil, E. Deelman, M. Ellisman, T. Fahringer, G. Fox, D. Gannon, C. Goble, M. Livny, L. Moreau, and J. Myers, "Examining the challenges of scientific workflows," *Computer*, vol. 40, no. 12, pp. 24–32, 2007.

[6] E. Deelman, T. Peterka, I. Altintas, C. D. Carothers, K. K. van Dam, K. Moreland, M. Parashar, L. Ramakrishnan, M. Taufer, and J. Vetter, "The future of scientific workflows," *The International Journal of High Performance Computing Applications*, vol. 32, no. 1, pp. 159–175, 2018.

[7] S. Higgins *et al.*, "The lifecycle of data management," *Managing research data*, pp. 17–45, 2012.

[8] A. Ball, *Review of data management lifecycle models*. Citeseer, 2012.

[9] H. Garcia-Molina and K. Salem, "Main memory database systems: An overview," *IEEE Transactions on knowledge and data engineering*, vol. 4, no. 6, pp. 509–516, 1992.

[10] J. Wang, D. Park, Y. Papakonstantinou, and S. Swanson, "SSD in-storage computing for search engines," *IEEE Transactions on Computers*, 2016.

[11] A. Chen, "A review of emerging non-volatile memory (NVM) technologies and applications," *Solid-State Electronics*, vol. 125, pp. 25–38, 2016.

The challenge of capturing and converting primary to secondary and summary datasets

David M. Rogers (rogersdm@ornl.gov, ORNL)
Kathryn Knight (knightke@ornl.gov, ORNL)
Dec. 15, 2021

Topic: Interfaces and Storage Architecture.

Challenge

Research programs in both experimental and computational sciences are generating primary data at ever-increasing rates. This data is often very difficult to capture and utilize, and as a consequence, loses value over time. Nevertheless, there is a growing demand for high-level insights from comparative studies performed over secondary, summary datasets. For a concrete example, the collection, over time, of photographs from a single microscope or output trajectories from time-integration performed by a single simulation program create streams of primary data. To advance science and guide the direction of research programs, we would like to gather data from multiple instruments pertaining to a single physical phenomenon, make question-specific summaries, and compare. In the field of materials sciences, there are several well-known examples of highly successful data compilations created in this way [1]. Nevertheless, these represent only a small fraction of the data resources now available.

The challenge here is to create policies for recognizing, capturing, and encouraging the processing and sharing of primary and secondary datasets. Can consortiums organized within fields of research create tools that lower the barriers for researchers to safely and effectively store and advertise their results? Can institutions contribute to the value of data by implementing policies for internal validation and publication? How can we encourage the formation and growth of communities producing and maintaining secondary data?

Opportunity

Materials-Project started as a collection of computed energies for periodic solid materials, and became a hub for a community of materials researchers interested in diverse properties. Wikipedia, Github, and Kaggle have similar stories. These are all examples of what is possible with online communities organized around the production and curation of data. Scientific data can benefit in similar ways from adapting these models. There is an opportunity to create a forum for publishing annotated datasets – for example to summarize an entire instrumental history or to gather disparate experiments on a selected set of materials. To prevent data storage explosion, these primary datasets might be given limited lifetimes or expire if they remain unused. The key outcome from such a forum would be opportunities for creating higher-level, summarized datasets. Summary datasets can then be used for a variety of novel and exciting research. Can an AI predict phase diagrams from lattice data? What is the “materials space” of compressibility, cost, hydrogen storage capacity, grain size dispersity, etc.? What types of alloys have the widest variation in damage resistance to ionizing

radiation? What past experiments have been performed that would help predict the outcome of a proposed new experiment?

Timeliness

There is an increasing, global awareness about the importance of comprehensive, publicly accessible, reference databases. Part of the driver is AI approaches, which require well-typed, formatted, and comprehensive datasets in order to make useful predictions. Another is the continuing need for maintaining a single source of reference data to prevent unnecessary duplication of effort. National-level efforts are currently being directed toward collecting and curating data libraries in multiple domains of science, arts, and humanities [2]. At the same time, there is already a critical mass of existing databases and consortia which are available to assist in the process of developing data ontologies and defining best practices [3]. A concerted effort right now toward providing a platform, storage mechanism, and method for facilitating discussions between these groups could have an outsized impact on the global state of data. The DOE could potentially benefit both by having a platform for capturing its internal data (aiding data producers) as well as taking the lead to shape a data access platform (aiding data consumers).

References

[1] Gen-IV Materials Handbook, https://www.osti.gov/biblio/1110968/
CRC Handbook of Chem. & Physics, https://hbcp.chemnetbase.com
NIST Chemistry Webbook, https://webbook.nist.gov/
NIH Pubchem, https://pubchem.ncbi.nlm.nih.gov/
Granta Material Property Charts, https://www.grantadesign.com/education/students/charts/
Japan National Institute for Materials Science Creep Dataset, https://www.nims.go.jp/eng/news/press/2021/04/202104010.html

[2] NIST Materials Registry, https://materials.registry.nist.gov
DOE Code database, https://www.osti.gov/doecode/
LUNA online media library https://luna.folger.edu
Nature - Scientific Data https://www.nature.com/sdata/
https://www.micropublication.org/
https://think.f1000research.com/materialsopenresearch/

[3] Materials Research Data Alliance, https://github.com/marda-dd
NIST Data ontologies https://data.nist.gov/od/dm/nmrr/vocab/
Research-Object Crate Project https://www.researchobject.org/ro-crate/
Data Carpentry, https://datacarpentry.org

Intelligent HPC Storage Systems for Scientific Workflows

Devarshi Ghoshal, Drew Paine, Lavanya Ramakrishnan
Lawrence Berkeley National Lab
[dghoshal, pained, lramakrishnan]@lbl.gov

Topic: Data management support for complex workflows

Today, scientists largely manage their workflows and storage systems separately relying on ad-hoc scripts, manual steps, or rarely, workflow tools. This has resulted in a fundamental disconnect between user workflows and the storage systems at high-performance computing (HPC) facilities since workflow tools treat the storage as a blackbox and the storage systems have little or no knowledge about the workflows. For example, past studies showed that adding a fast “burst buffer” tier need not necessarily improve the performance of all workflows running on HPC systems [1,2]. There is a need for future storage systems to be intelligent such that they learn and adapt based on the workflows and data they handle and inform decisions at the workflow-level.

Challenges

One of the major challenges in designing an intelligent storage system for the future is the inability to identify and predict the myriad workflow and data access patterns that exist on HPC systems. Past research proposed abstractions and policies for efficiently managing workflows and data on hierarchical storage systems [2,3]. However, there is a lack of appropriate design patterns that could inform users about the different possibilities and trade-offs of managing workflow data across the different storage systems. In addition to that, users and/or workflows manage their data and workflows without any feedback from the storage system, which results in suboptimal I/O and workflow performance. It is challenging to predict the optimal workflow and data management strategy without experimenting across different systems, using different configurations. Additionally, users and system administrators have to manually plan for storage capacity and usage. The interplay of data life cycles, persistence models of storage systems, and workflows are often ignored. This becomes even more challenging when workflows need to manage data across edge, cloud and HPC resources.

Opportunities

Detecting and predicting patterns. The next-generation of storage systems can be more aware of the workflows and data. Past research has shown the importance of detecting workflow patterns by analyzing job and I/O logs in identifying the needs for future systems [4,5]. Next-generation storage systems need to be able to identify and classify different workflow and data access patterns automatically for efficiently managing data. There is an opportunity to use AI/ML methods to derive patterns across different workflows where simple log-based analysis might fail. These methods might be able to correlate and identify workflow patterns across different science domains. Additionally, an AI-enabled storage system can build predictive and analytical models iteratively to continuously improve the understanding and management of workflows and associated data. Such methods can also be used to correlate performance bottlenecks to different workflow and data access patterns.

Using AI for optimizing data management. As data gets distributed and moved across different storage systems by the workflows, there is an opportunity to provide feedback from the storage systems to the users. This feedback can be used not only for efficient data distribution

and management, but also for optimizing the use of different storage systems. There is an opportunity for AI-based recommendation systems to provide users with the necessary information required to optimize workflow and data management performance. Such recommendations can be used to alleviate I/O bottlenecks, minimize performance variability, and identify trade-offs of different storage options. Past research has shown that data management strategies need to evolve based on workflow and data needs [2,3,5]. Instead of static and/or ad hoc policies for storing, moving and managing data for complex scientific workflows, there is an opportunity for AI-based recommendation and automation systems to manage the storage resources more effectively and dynamically. These recommendations will influence the design of optimal data management strategies, and also allow users to better understand their data and workflows across different storage systems.

Enabling elastic data management on federated storage systems. The growth in the volume of and evolving scientific software architectures that include edge and cloud computing have made data management for complex workflows across distributed heterogeneous storage resources a new reality. While existing models of allocating storage resources to workflows are mostly static, distributed infrastructures open up the possibility of using storage resources on-demand across the HPC and cloud infrastructures. AI can play a critical role in predicting the allocation and usage of appropriate storage resources for users and workflows, and drive the decisions about where, when and how to move data. This provides an opportunity for future storage systems to be more dynamic and elastic in nature, instead of being a static resource. The use of workflow patterns and AI-based recommendations can help users, system admins and workflow managers to automatically and efficiently distribute data and allocate storage resources. Users will be able to dynamically grow and shrink storage resources and manage their data based on execution, data usage and I/O patterns of the workflows.

Timeliness

Data from scientific experiments, observations and simulations is increasing at a rapid rate. It is necessary that such large amounts of data are managed efficiently and effectively by the users and their workflows. Novel methods are required that would allow storage systems to interact intelligently with the workflows. With the advancements in AI, methods can be developed and integrated into the next-generation storage systems that will allow them to learn, adapt and provide feedback for achieving optimal workflow performance.

References

[1] Daley, Christopher S. et al. "Performance characterization of scientific workflows for the optimal use of burst buffers." Future Generation Computer Systems 110, 2020.

[2] Ghoshal, Devarshi, and Lavanya Ramakrishnan. "Madats: Managing data on tiered storage for scientific workflows." In Proceedings of the 26th International Symposium on High-Performance Parallel and Distributed Computing, pp. 41-52. 2017.

[3] Shin, Woong, et al. "Data Jockey: Automatic data management for HPC multi-tiered storage systems." In 2019 IEEE International Parallel and Distributed Processing Symposium, 2019.

[4] Ghoshal, Devarshi, et al. "Characterizing Scientific Workflows on HPC Systems using Logs." In 2020 IEEE/ACM Workflows in Support of Large-Scale Science (WORKS), 2020.

[5] Paul, Arnab K., et al. "Characterizing Machine Learning I/O Workloads on Leadership Scale HPC Systems." In 2021 29th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems (MASCOTS), 2021.

Incentive-driven I/O Resource Management and Data Management for HPC Users

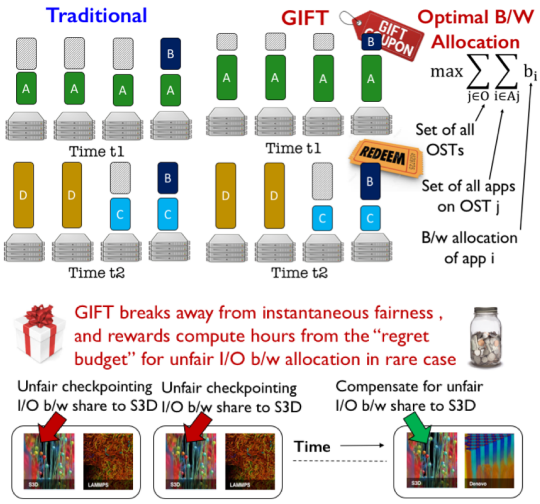
Devesh Tiwari (d.tiwari@northeastern.edu) Northeastern University
Goodwill Computing Lab: https://goodwillcomputinglab.github.io/

Problem Background and Motivation

HPC data management and I/O resource management are tightly coupled. Our users produce data via running complex AI and science workflows, they store this data and then, wish to access it at a later point – hopefully, *expecting FAIR principles* (Find-ability, Accessibility, Interoperability, and Reusability). Unfortunately, at every single step, users need to access the I/O resources and deal with the consequences of the I/O resource management policies. With the growing number of users and the complexity of AI workflows, our data management and I/O resource management policies are getting severely challenging. The traditional *"selfish" approach of I/O performance tuning and better resource allocation for oneself will not be sustainable* – this is what we have been doing for decades. Determining a few "one size fits all" and "constant at all times" policies will become ineffective with growing AI workflow complexity and diversity in user characteristics. *Instead, we argue that data management and I/O resource management should become* ***"incentive-driven" where users are "helping" each other to maximize the quality of everyone's observed experience at times when it matters to them*** *– this will be key to bridging the SSIO and data management (DM).*

Key Ideas of the Approach

This whole approach is inspired by two recent works on the use of incentives for HPC resource management [1,2]. The first work is directly demonstrated in the context of data management and I/O resources. In particular, this work, GIFT [1], shows that it is possible to provide both competing goals -- much better overall system throughput (system-level goal) and per-application execution time (user-level goal), when a coupon-based system is used for I/O bandwidth allocation (depicted via a visual representation below). The system obfuscates the coupon management details from the users, but provides both fairness and efficiency in the long-term using coupons (throttle now and reward at a later point). It also shows that one can apply this concept to trade-off I/O performance with compute time (barter system) to minimize the overall regret, when appropriate. The second work, Shiraz, demonstrates similar concepts but in the context of I/O (checkpointing) and reliability – how some applications can be incentivized to perform I/O at relatively less-reliable times[2].



Opportunities and Areas of Investment

Multiple interesting and challenging opportunities exist for us to make incentive-driven systems operational and production-ready. We need to invest staff time and effort to develop a system software stack to specific incentive-driven strategies – this will essentially involve enriching I/O and user accounting stack with multiple game-theoretic kernels and testing it out for different user behavior, access patterns, and reuse patterns.

Opportunities involve identifying areas where the incentive is likely to be most effective. *For example, incentives around reducing I/O contention during high peak load are likely to be*

successful where users can be given coupons, and later they can be redeemed for better performance. Similarly, data management policies could benefit from incentives where different users could use different storage hierarchy levels (burst buffer vs. PFS) at different times for opportunistic performance. *Users can be incentivized to purge their data quickly for better I/O performance or job wait time.*

A key opportunity is to develop systems and techniques that promote users to participate in such incentive-driven I/O management systems – essentially, the system should specify how users can benefit from such an approach and what their expectations should be in the short-term and long-term? From a system's point of view, ***it is important to develop techniques that can provide provable guarantees about I/O performance in a control-theoretic manner***. This control-theoretic core can ***leverage AI to learn user behavior (from data life cycle provenance) to design more effective incentive-drive user interfaces for scientists.***

Finally, as a systems' core building block, we need to implement a mechanism that ensures that users cannot "game" such systems – that is, the system needs to remain strategy-proof. This is directly related to SSIO and DM's *focus on user access interface and metadata management using FAIR principles*. This aspect requires careful consideration and robust design to ensure that incentive are being managed at a level that cannot be easily gamed, but at the same time users are reasonably happy and understand that FAIR principles are being followed.

Timeline and Anticipated Challenges

The expected timeline is approximately 3-5 years. Most of the timeline would be in developing system software stack which have configurable incentive-driven data management and I/O resource management policies. Experimental prototype creation, implementation and validation at large scale will require significant time and staff investment. The major challenge would be buy-in from users and social aspects of this proposed. It's an unorthodox approach, from a computer science / computer systems point of view, because we have traditionally been taught to design deterministic systems. In that sense, incentive-driven approach might appear non-intuitive at first, but they have the potential to be transformative – almost all breakthrough scientific discoveries are a result of collaborative efforts among scientists, then, it only makes that we can utilize our systems better and make it effective if we can build a collaborative framework around our users!

References

[1] Tirthak Patel, Rohan Garg, and Devesh Tiwari, "GIFT: A Coupon Based Throttle-and-Reward Mechanism for Fair and Efficient I/O Bandwidth Management on Parallel Storage Systems", USENIX FAST 2020.
[2] Rohan Garg, Tirthak Patel, Gene Cooperman, and Devesh Tiwari, "Shiraz: Exploiting System Reliability and Application Resilience Characteristics to Improve Large Scale System Throughput", DSN 2018

Towards Unified Intelligent High Performance Computing Storage Systems

Dong Dai, University of North Carolina at Charlotte, ddai@uncc.edu

Topic: This position paper focuses on a new storage-system architecture design.

Challenge: High-Performance Computing (HPC) faces significant I/O challenges. The applications' data volume and their needs for high-speed data accessing are growing fast. HPC storage is also expected to deliver low latency and high IOPS for emerging machine learning and artificial intelligence workloads [3].

To fulfill such requirements, modern HPC systems deploy multi-tiered storage stacks, which may include client-side on-demand file systems (e.g., GeKkoFS or BeeOND), burst buffers (e.g., DataWarp or IME), together with the traditional parallel file systems. These storage layers obtain distinct latency, bandwidth, capability, data visibility, and data durability attributes. As a result, how to leverage them most effectively and to obtain the best performance becomes a critical job to the end-users.

This places significant burdens on end-users as they need to know available storage options in the HPC systems, performance, capacity, data visibility, and durability. In addition to the usability challenge, letting users manually configure the multi-tiered HPC storage systems often misses significant runtime optimization opportunities [5]. For instance, without deep understandings of the I/O patterns, users may select sub-optimal I/O configurations (e.g., data size, location, or durability); without a global view of the entire storage system, multiple applications and users may compete for the same storage resource, resulting in a limited performance [7]; without a real-time view of the storage resource statuses, users can not dynamically schedule or tune I/O operations in runtime to improve the performance [2].

As the HPC storage becomes more complex and heterogeneous, effectively and productively using it will simply become more challenging; the gained performance will be unsatisfactory as well. This problem roots in the isolated design of current HPC storage systems and their high dependencies on users' manual tuning and configurations. Therefore, We see a need to shift its design from *isolated multi-tired storage* to *unified storage*; from *users' manual configuration* to *intelligent automatic management*.

Opportunity: The key to addressing the previously described challenges is to build *a unified, intelligent HPC storage system* that automatically delivers high I/O performance to end-users.

If we consider the heterogeneous storage devices are just running a slim layer of software to store data objects and communicate with each others, then to achieve such a unified and intelligent storage, we will need an intelligent data management system to conduct most of the configuration and tuning work, such as determining data locations, directing data buffering and moving, scheduling I/O requests, without any users' hints or manual configurations. We expect such data management decisions would be made by machine learning components based on historical patterns and real-time storage system status, effectively delivered by an extremely high-performance metadata layer. The machine learning components will also dynamically tune the I/O operations based on the real-time system status, together with achieving optimal performance.

To enable such a unified and intelligent storage system in HPC, we need to address multiple challenging questions that are barely touched in previous work.

- First, what metadata is necessary, adequate, and accurate for data management tasks, such as data location selection, visibility and consistency tuning, and dynamic IO scheduling? Many of these metadata are about the runtime status of other HPC components, such as batch job schedulers. How can they be integrated with the POSIX namespace to enable unified and easy metadata access?
- Second, how to support the common metadata operations and the data management functionalities efficiently to match the scale and speed requirements of modern HPC systems? The upcoming Exascale machines may deliver IO maximally at billions of random-read IOPS [1], which far exceed the IOPS

that SSDs and HDDs. Persistent memory seems promising. However, can it deliver such a speed to work with future systems?

- Third, how to improve the intelligence of data management such that it can coherently manage various IO tasks to achieve extreme performance? Existing studies touched a small part of these IO tasks such as data location, pre-fetching, or asynchronous data movement but lack understanding of how they work together and miss runtime I/O tuning. So, how the runtime I/O tuning can be done by machine learning models?

Timeliness or maturity: Building such a unified, intelligent HPC storage system becomes feasible now for two reasons: 1) the availability of new persistent memory (PMEM) devices to support needed fast metadata operations; 2) the progress in deep learning to accurately capture I/O patterns. Here we focus on explaining the maturity of persistent memory, while the machine learning part has been proven in many recent studies.

Feasibility of persistent memory. Persistent memory (PMEM) (particularly, Intel Optane DC Persistent Memory [4]) is a new kind of memory device that provides near-DRAM data access latency, higher capacity, lower-price, and data persistence. Its higher density, lower cost, and near-zero standby power cost make it a perfect choice for implementing our proposed global metadata management layer. For instance, for 128-byte small random reads, a single fully populated PMEM server can deliver 60 million IOPS [6]. We know that DOE's upcoming Exascale machine Frontier will deliver around 5TB/s read/write bandwidth and 2 million random-read IOPS. Its on-demand file systems on compute nodes will maximally deliver 75TB/s read bandwidth and 15 billion random-read IOPS. Then theoretically, we just need 25 PMEM servers to deliver 15B IOPS. Although this is just a theoretical calculation, it still shows the feasibility of persistent memory to deliver needed fast metadata management.

References

[1] OLCF ANNOUNCES STORAGE SPECIFICATIONS FOR FRONTIER EXASCALE SYSTEM. https://www.olcf.ornl.gov/2021/05/20/olcf-announces-storage-specifications-for-frontier-exascale-system/.

[2] D. Dai, Y. Chen, D. Kimpe, and R. Ross. Two-Choice Randomized Dynamic I/O Scheduler for Object Storage Systems. In *International Conference for High Performance Computing, Networking, Storage and Analysis, SC*, volume 2015-Janua, pages 635–646, 11 2014.

[3] G. K. Lockwood, D. Hazen, Q. Koziol, R. S. Canon, K. Antypas, J. Balewski, N. Balthaser, W. Bhimji, J. Botts, J. Broughton, et al. Storage 2020: A vision for the future of hpc storage. 2017.

[4] Optane. Intel Optane Persistent Memory. https://www.intel.com/content/www/us/en/products/docs/memory-storage/optane-persistent-memory/optane-dc-persistent-memory-brief.html, 2019. Accessed: 2019-11.

[5] Y. Qian, X. Li, S. Ihara, A. Dilger, C. Thomaz, S. Wang, W. Cheng, C. Li, L. Zeng, F. Wang, et al. Lpcc: Hierarchical persistent client caching for lustre. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pages 1–14, 2019.

[6] T. Tristian and L. Travis. Analyzing the performance of intel optane dc persistent memory in app direct mode in lenovo thinksystem servers, 2019.

[7] M. R. Wyatt, S. Herbein, K. Shoga, T. Gamblin, and M. Taufer. Canario: Sounding the alarm on io-related performance degradation. In *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 73–83. IEEE, 2020.

Scientific Data Management in Disaggregated Heterogeneous Memory

Dong Li*, Ivy Bo Peng, and Maya Gokhale
University of California, Merced* Lawrence Livermore National Lab
dli35@ucmerced.edu {peng8,gokhale2}@llnl.gov

This article is with a release number LLNL-ABS-830181

The growing needs of experimental and observational science generate vast amount of data, and the volume of generated data is growing exponentially at an unprecedented rate. Meanwhile, memory is increasingly heterogeneous based upon the advances of hardware technologies, such as 3D XPoint, through silicon via (TSV) and fast interconnect. The traditional assumptions about data management (including architecture, memory profiling methods, memory allocation, interfaces for accessing data, and supporting of AI and complex workflows) must be revisited. This position paper discuss the impact of emerging disaggregated heterogeneous memory (DHM) on scientific data management. DHM uses network attached memory that is distinct from the memory in the compute nodes. This approach allows the disaggregated memory to scale independently of the system's computing or storage capacity, and removes the need to over-provision one resource to scale another.

1 Challenges

Underutilization of memory resource in production HPC. We performed a large-scale study to understand how exiting workloads on production HPC systems at Lawrence Livermore National Lab (LLNL) utilize the memory resources [1]. For this, we analyzed more than two million jobs on four HPC clusters. Our results show that most jobs only utilize a small fraction of memory resources, and for more than 90% time, a node utilizes less than 35% memory capacity. How to utilize memory resource efficiently is a challenge.

Data management challenges on multi-tiered memory architecture. The memory system is becoming more and more heterogeneous. The traditional NUMA system in HPC is adding new memory technologies, such as high-bandwidth memory (HBM) and high-density persistent memory (PM), which easily exceeds two memory tiers in the traditional design. Intel's Optane DC persistent memory-based machine is such an example where we see persistent memory plus DRAM create two tiers and NUMA effects create another two. Disaggregated memory system, far away from the compute nodes, will add another tier. Different tiers have different latency, bandwidth, and capacity, creating challenges on data management. Figure 1 depicts such a memory architecture with rich memory heterogeneity.

The data management in the multi-tiered memory architecture includes at least three problems: (1) deciding where scientific data (including raw data and metadata) should be allocated, (2) deciding how data should be migrated between memory tiers to make the best use of fast memory tiers for high performance, and (3) deciding how data objects should be co-located in the same page or in the same memory banks to avoid page-level false sharing or bank conflicts.

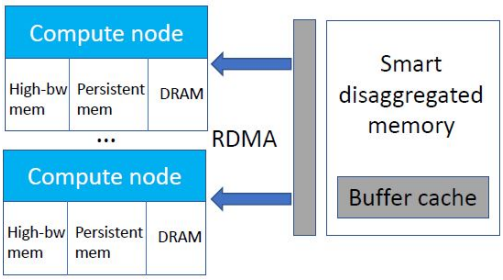


Figure 1: Architecture of disaggregated heterogeneous memory.

The complexity of data management on multi-tiered heterogeneous memory creates a major programming challenge for domain scientists. There are a large amount of data objects with different lifetime, access frequency, and memory footprint. Mapping them to rich memory tiers introduces an extremely large design space. Workload knowledge can help leverage such memory/storage architecture. however, current system-level solutions, e.g., file systems and operating systems, cannot be tailored quickly for the rapid changes on architecture or for specific workflows/workloads because their privilege access and system-wide impact. Relying solely on developers to optimize at per-application basis is non-scalable/portable. Thus, there is a gap between system software and applications, which can be filled with novel approaches for understanding (profiling) and learning workloads on heterogeneous architecture and knowledge transfer.

2 Opportunities

The above challenges provide many opportunities to study new data management frameworks. Software-based data management on heterogeneous memory consists of at least three components – a profiling

mechanism, a migration policy, and a migration mechanism. A profiling mechanism is critical for identifying performance-critical data in applications and is often realized through tracking page accesses. A migration policy chooses candidate pages to be moved to top tiers. Finally, the effectiveness of a data management solution directly depends on whether its migration mechanism can move pages across tiers at low overhead.

Scalable profiling. Existing memory profiling mechanisms manipulate specific bits in page table entries (PTEs) to track memory accesses at a per-page granularity. The profiling overhead scales linearly with the number of tracked pages. Some solutions only profile a small set of randomly-chosen pages based on PTE manipulation or performance counters, or heavily rely on the user to configure the profiling method to reduce profiling overhead. However, such a strategy compromises profiling quality and may miss frequently-accessed pages and time-changing access patterns.

Formal metrics that quantify performance impact from pages need to be established to guide page selection for profiling. By only tracking the most performance-critical pages at a time, a solution can guarantee adaptiveness to time-changing patterns and controlled overhead.

Learned data placement on disaggregated heterogeneous memory aims to use machine learning models as a migration policy to decide data allocation and migration on DHM. Existing migration policies either rely on a memory profiling mechanism or use domain knowledge to decide data access frequency and lifetime. Using the memory profiling is *reactive* in nature, because it is based on the assumption that the future data access patterns bear similarity to the recent ones measured by a memory profiler. However, this assumption ignores data semantics and is often not held for irregular data access patterns. As a result, the memory profiling-based approach cannot effectively direct data placement for complex workflows or scientific applications with irregular data access patterns. Using domain knowledge, we can have a good knowledge (sometimes parameterized performance modeling) on access patterns and lifetime of data objects, and hence take action early to migrate data objects for high performance or memory saving [2]. However, using domain knowledge demands deep knowledge on scientific applications and domain science, which are not always available to scientists.

Machine learning can learn implicit data semantics and even domain knowledge, and hence can address the limitation of the above two approaches. For example, machine learning can learn data correlation within a sequence of data access, and hence enable data prefetcing for high performance. This is especially useful for those workloads with irregular (or even random) but still follow some distributions in data accesses. Using machine learning to guide data migration for disaggregated memory is especially useful, because data movement across nodes is relatively slow (compared with data movement with a node) even with emerging fast interconnect, and using machine learning can proactively trigger data movement to hide the migration cost. However, using the learned data placement must address a set of challenges, such as reducing inference time, deciding input features, and handing mis-prediction cases.

Data access interface. The remote disaggregated memory and local memory can build a global address space. Data access will be based on load/store instructions with potential page faults intercepted and redirected to the remote memory to fetch pages. Based on the emerging cache coherence protocol (such as CXL), the data access across the compute node and the remote memory pool can also happen at a finer granularity (e.g., cache block level), which allows tracking of data dirtiness and reduce unnecessary data movement. The data allocation will be based on a new allocation interface allowing the user to give hints on data semantics (e.g., metadata or observational scientific data). The traditional data format (such as HDF) can be significantly simplified to benefit from the memory-centric data access method.

3 Timeliness and Maturity

The emerging disaggregated heterogeneous memory is based upon the recent progress in the interconnect technologies. For example, high-end infiniband-based deployment (FDR/EDR) can offer low latency at the order of microseconds, close to local memory latency. The emerging optical interconnect technology based on the recent advances in integrated photonics can further enable scalable rack-distance and energy-efficient interconnects. Built upon the fast interconnection, GenZ, CCIX, OpenCAPI, and CXL provide memory-semantic access to data with possible cache coherence in place.

References

[1] I. Peng, R. Pearce, and M. Gokhale, "On the Memory Underutilization: Exploring Disaggregated Memory on HPC Systems," in *IEEE International Symposium on Computer Architecture and High Performance Computing*, 2020.

[2] J. Ren, J. Luo, I. Peng, K. Wu, and D. Li, "Optimizing Large-Scale Plasma Simulations on Persistent Memory-based Heterogeneous Memory with Effective Data Placement Across Memory Hierarchy," in *International Conference on Supercomputing (ICS)*, 2021.

Storage system design tools for asynchronous, non-uniform, hybrid, highly distributed I/Os

Franck Cappello (cappello@anl.gov), Bogdan Nicolae (ANL)

Topics: This white paper is addressing the following topics of the call: Storage-system architecture design. It is also related to: Data-management support for AI and complex workflows.

Challenge: Scientific applications users, I/O runtime designers and storage systems developers are facing a massive I/O and storage paradigm shift through the combination of 5 almost simultaneous changes in the I/O patterns of supercomputing facilities. The first change is the adoption by all I/O libraries of **asynchronous** I/Os permitted by the inclusion in HPC systems of high-performance local storage such as NVme [1]. The second change is the rapid adoption by users of **lossy data reduction** [2] that leads to non-uniform I/Os from nodes running parallel executions. The third change is the introduction of AI to accelerate computation and to perform data analytics. **AI exhibits specific I/O patterns different from HPC simulations** [3]. The fourth change is the integration in the HPC systems of latency critical tasks that must be executed as soon as possible. This new need coming from the integration of edge computing (facility instruments, sensor networks) and HPC systems break the traditional batch scheduling model and requires **fast I/Os to implement effective preemption** [4]. The fifth potential change (not yet widely adopted in the HPC community but widely adopted in other domains) **is the transition toward highly distributed object stores** [5] in replacement of the classic centralized parallel file systems **and the adoption of computational storage** [6] that allows to move some computations directly on the storage systems. Researchers in I/Os and storage systems are currently mostly considering the impacts of these 5 changes independently on existing infrastructures. We argue that because these changes will happen in a short time frame, they must be considered together through a holistic approach for the design of the next generation storage systems and I/O runtimes.

Opportunity: Research and development of tools that would identify favorable pathways to optimize the storage infrastructure and I/O runtime environments in the context of the combination of these 5 changes. The community needs characterization studies, performance models and simulations (and potentially emulations) to understand the impact of the I/O patterns and optimization opportunities in the design space of holistic storage solutions that combine future hardware technologies and runtimes in various configurations, both from the perspective of performance (read/write latency, throughput) and reliability. Examples of questions that such tools would address: 1) How to handle the metadata and data? Shall they be stored together or separately? If separately, what storage design would optimize the performance for both? Can computational storage help accelerate metadata processing? 2) What size should each storage tier be? Should the storage system be rather flat and centralized, fully hierarchical and distributed or any configuration in between? 3) How separated should the system traffic and the storage traffic be? Should they be physically separated by using different interconnects and servers or should they use the same physical infrastructure possibly augmented with software mechanisms to isolate them? 4) What impact does the differences in data representation and access model have across the storage hierarchy? For example, if persistent memories use a memory-oriented access model (byte-level access using pointers), what is the cost of emulating this model on top of external repositories that support put/get or POSIX models?

Case study example: One example of convergence between HPC and ML workloads is HPC ensemble simulations driven by DNN models, which learn the patterns exhibited by the simulations and decide how to select the next simulation parameters. The DNN models are trained on-the-fly with the outputs of the simulations, which are curated and labeled. This is a write-intensive I/O scenario that involves both data and metadata. On the DNN training side, the same data is read-intensive and deterministic (i.e., order of reads is known in advance). Write operations could take advantage of the deterministic nature of reads to implement I/O optimizations. However, the complexity of this scenario raises questions at all levels: is it better to share I/O resources between the readers and the writers (and potentially risk conflicting optimization goals), or is it better to dedicate separate I/O resources and handle data movements between readers and writers explicitly? Should the metadata be stored together with the data or separately? On what storage tiers? Are optimization decisions consistent across various I/O access models or does the optimal strategy need to change (e.g. when moving from POSIX to object stores)? Without rigorous characterization, performance modeling and simulation efforts, it is easy to miss a large number of optimization opportunities, let alone explore the design space that implements them.

Timeliness or maturity: The five discussed changes will happen within the next 2-3 years on systems like Aurora at Argonne. Aurora will be a platform of choice to capture the new I/O workload because it will run applications using asynchronous I/O, compression and a mix of AI and HPC tasks. It will also feature a distributed object store that can be used as an alternative to parallel file systems and that could be monitored to understand how the mixed I/O workload translates to low-level storage operations on a distributed object store. ALCF is also exploring ways to implement effective preemption for latency sensitive tasks. Tools exist to support this research such as Darshan that characterize I/Os and storage system simulators like CODES. The community will need to perform a gap analysis to identify the new tools to develop or the new features to augment existing tools to enable the exploration of the large new parameter space defined by the 5 changes in the I/O workload and storage software and hardware.

Potential scientific impact. The scientific impact will first be about the design of the next generation storage systems for HPC and AI workloads featuring asynchronous I/O, latency sensitive tasks, compressed data, object store interfaces and computational storage devices. The broader scientific impact will be on I/O runtime design and optimization for the next generation storage systems. Impact is also expected at the facility level: the availability of tools and research studies on storage system design considering the 5 mentioned changes will inform and help identify relevant designs for procurements.

References:

[1] B. Dong et al., Data Elevator: Low-Contention Data Movement in Hierarchical Storage System, 2016 IEEE 23rd International Conference on High Performance Computing (HiPC), 2016

[2] F. Cappello, S. Di, S. Li, X. Liang, A. Murat Gok, D. Tao, X.-C. Wu, Y. Alexeev, F. T. Chong, Use-cases of lossy compression for floating-point data in scientific datasets, IJHPCA, Vol 33, Issue 6, 2019.

[3] S. Chien et al. Characterizing Deep-Learning I/O Workloads in TensorFlow. IEEE/ACM PDSW-DISCS workshop, 2018.

[4] M. Agung, Y. Watanabe, H. Weber, R. Egawa, H. Takizawa, Preemptive Parallel Job Scheduling for Heterogeneous Systems Supporting Urgent Computing, IEEE Access, vol. 9, pp. 17557-17571, 2021

[5] J. Lofstead, et al. DAOS and Friends: A Proposal for an Exascale Storage System, SC '16, 2016

[6] M. Torabzadehkashi, et al. Accelerating HPC Applications Using Computational Storage Devices, IEEE 21st International Conference on High Performance Computing and Communications, 2019

Making SSIO FAIR

Galen Shipman (corresponding author)
Los Alamos National Laboratory
gshipman@lanl.gov

Topics: Metadata management infrastructure to support FAIR principle, The overlap between traditional storage systems and I/O (SSIO) efforts and data management.

Challenge

The Department of Energy (DOE) Office of Science (SC) provides the world's most capable scientific instruments, large-scale computational facilities, and observational networks. These scientific resources have enabled significant advancements in fields spanning Nuclear Physics to Earth System Sciences. Today many of these scientific advancements are made by small teams of researchers working in an extremely close (and agile) collaboration. These collaborations are often deeply embedded in a particular domain and carry with them a tailored set of practices and technologies in data management. Unfortunately, these practices and the technologies that support them can miss critical information necessary to support Findability, Accessibility, Interoperability, and Reusability (FAIR) [1] beyond the small collaboration.

Technologies across these collaborations are often developed independently and without significant engagement with the SSIO community. This can result in a fragmented set of tooling with poor interoperability with state-of-the-art SSIO technologies and suboptimal functional reuse. On the SSIO side, a lack of understanding and visibility into FAIR data management requirements can result in technologies that provide poor support for these requirements, reinforcing the need to develop stand-alone technologies among established teams.

Opportunity

Many of the technical requirements brought by FAIR data management could benefit from focused SSIO R&D. A few examples of these opportunities include:

- Findability could benefit from SSIO technologies for efficient provenance and metadata capture such as property graphs [2] which allow properties and relationships of data to be efficiently represented for large-scale simulation campaigns. Recent work in scalable metadata services [3] have demonstrated significant performance gains in advanced metadata operations relative to MongoDB and SciDB systems.

- Accessibility could benefit from SSIO technologies for more scalable and seamless access to multiple tiers of storage within an HPC facility, from high-performance flash storage tiers, disk based parallel file system tiers built upon GPFS or Lustre, lower performance but high-capacity campaign storage built on object storage, or even large-scale archival storage systems. This tiering strategy is an important design point enabling each tier to be optimized for different performance and capacity requirements but can often form a bulwark to accessibility.

- Interoperability could benefit from a focus on high-performance and scalable support for highly descriptive and structured data formats that can be used across large-scale simulation, scientific

instruments, and observation facilities or campaigns. Research that bridges the performance and scalability gap between highly descriptive formats such as Resource Description Frameworks (RDFs) and highly "relaxed" structures used in HPC is sorely needed.

- Reusability often requires significant data engineering, a process of data transformation and reorganization which traditional HPC storage system technologies were not designed for. Understanding these workloads and their underlying requirements could significantly influence future SSIO architecture and design.

Timeliness

While the entire breadth of capabilities required to support FAIR data management is daunting, focusing on core mechanisms that could be provided through targeted SSIO R&D could have transformative benefits for DOE, providing foundational technologies for making SC data, "AI Science Ready".

AI techniques have been adopted across many HPC science campaigns and facilities (ALCF, OLCF, and NERSC) have rapidly deployed deep-learning and other AI technologies to facilitate these workloads. This presents a timely opportunity for SSIO R&D to address FAIR data management requirements across traditional scientific simulation and AI workloads in an integrated fashion.

Recent advances in storage system software composition [4] provide a methodology and technology platform on which services to support FAIR data management could be rapidly prototyped, evaluated, and deployed. Hardware innovations such as computational storage and networking occurring across multiple technology vendors provide a unique opportunity to co-design these services across data-management, SSIO, and hardware.

References

[1] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, et al. The fair guiding principles for scientific data management and stewardship. Scientific data, 3, 2016.

[2] H. Tang, S. Byna, B. Dong, J. Liu, and Q. Koziol. Someta: Scalable object-centric metadata management for high performance computing. In 2017 IEEE International Conference on Cluster Computing (CLUSTER), pages 359–369. IEEE, 2017.

[3] D. Dai, Y. Chen, P. Carns, J. Jenkins, W. Zhang, and R. Ross. Graphmeta: a graph-based engine for managing large-scale hpc rich metadata. In 2016 IEEE International Conference on Cluster Computing (CLUSTER), pages 298–307. IEEE, 2016.

[4] Robert B. Ross et al. Mochi: Composing Data Services for High-Performance Computing Environments. Journal of Computer Science and Technology. 35, 121–144 (2020).

Harnessing Hierarchy

Gerd Heber[1], Chris Hogan[1], Anthony Kougkas[2]

Topics:

- Storage-system architecture design that supports scientific workflows on varied hierarchical storage and networking devices
- Utilizing AI to learn I/O patterns of emerging workloads for efficient data management

Challenge

To achieve good performance, today's users of HPC systems are being asked to know a lot about the system, its hardware and software composition, and a long list of dos and don'ts. And, of course, this changes from system to system. From an I/O perspective, perhaps only RAM and parallel file systems (PFS) have been present with some predictability. Whenever "shiny" go-between hardware showed up, the job of making changes fell to application developers. Perhaps the main reason for this was the lack of support for these "outliers" in the underlying middleware.

In principle there are four categories of escape routes:

1. Stop interspersing new hardware
2. Make the existing middleware software hierarchy-aware
3. Create a uniform (in appearance!) hierarchical aggregate
4. Create a new middleware capable of handling the complexity of modern hierarchical storage

Option 1. is perhaps only theoretical. Option 2. seem appealing because it would shield applications from change. However, we believe that the success of retrofitting middleware layers might be limited due to complexity, the cross-cutting nature of hierarchy-awareness, and application changes being necessary in the end to achieve performance gains that would make the whole effort seem worthwhile. Option 3. would achieve the same (shielding applications from change), but it is likely that, in practice, its non-uniform access characteristics will bleed back into applications without any ability to control it, aside from application code changes. Options 1-3. also share a common problem, which is that they implicitly perpetuate an abstraction, the notion of filesystems and files, which is perhaps not the direction that high-performance storage or cloud storage are headed, at the moment.

The challenge is to design something that achieves the following:

a. It is hierarchy-aware.
b. It is transparent to applications.
c. It plays well with existing middleware.
d. It is neutral with respect to I/O abstractions.
e. It runs in user space without the need for elevated privileges.

Item a. means that it shows demonstrably better performance by utilizing hierarchies over the PFS baseline. Item e. is not strictly necessary, but a practical consideration.

Opportunity

We believe that there is an opportunity to create a new middleware along the lines of option 4. by taking advantage of the full spectrum of new storage (e.g., PMEM, NVMe) and network hardware (e.g., IB RDMA, RoCE, NVMeoF), and by targeting new storage interfaces such as object stores. The new middleware

1 The HDF Group, gheber@hadfgroup.org, chogan@hdfgroup.org

2 Illinois Tech, akougkas@iit.edu

should not attempt to appear as a unified storage layer which exposes a certain storage abstraction. [4] That would be a mere return to option 3. Instead, it would appear as a distributed buffering layer with the following characteristics:

- A set of adapters for popular middleware (e.g., UNIX STDIO, POSIX, MPI-IO, HDF5) would extend existing applications w/o the need for code changes and make them hierarchy-aware.
- Buffering behavior at different levels (global default, per-call, etc.) would be governed by policies, which would shape the way data is distributed throughout the hierarchy and across nodes [1,2]. There would be default policies for common workloads, but also support for user-defined policies. This would also be a natural place for utilizing AI to learn I/O patterns [3] of emerging workloads for efficient data placement, routing, compression, etc.
- Buffering resources could be discovered and used dynamically, but also configured statically for well-understood scenarios and predictable resource utilization for buffering.
- This buffering middleware would appear as a distributed application extension. In order to support cooperating applications (e.g., “data hand-over”) a user space daemon would take custody of in-transition data or to support publisher/subscriber patterns.
- Finally, occupying a fairly central position at the middleware crossroads, we believe that there are interesting codesign opportunities, for example, with Mochi-style data services.

Timeliness and Impact

The deployment of a new breed of storage devices and ever more capable fabric in multiple DOE sites and the breathtaking speed of hardware turnover in public clouds pushes the question of how we are going to fully utilize that hardware to the forefront. Let’s create something that builds on what we have, but also supports complex scientific workflows that so far had to make due with improvisations and difficult compromises.

We believe that the proposed buffering layer will finally enable a less forceful convergence for hybrid BigData and HPC workloads, and dramatically accelerate workflows that hitherto had to use the PFS for communication between stages. Traditional write-heavy workloads such as checkpointing will benefit, and so will applications that have chosen to implement custom buffering such as LOFS [5]. Because buffering would occur in “multiple dimensions“ (e.g., vertically across hierarchy layers, and horizontally between nodes), we believe that seemingly pattern-free, read-heavy workloads as found in ML training [3] will see I/O performance gains. Finally, scratch-heavy read-after-write workloads such as reverse time migration might get a boost as well.

References

[1] Kougkas, Anthony, Hariharan Devarajan, and Xian-He Sun. "Hermes: a heterogeneous-aware multi-tiered distributed I/O buffering system." In Proceedings of the 27th International Symposium on High-Performance Parallel and Distributed Computing, pp. 219-230. 2018.

[2] Devarajan, Hariharan, Anthony Kougkas, Luke Logan, and Xian-He Sun. "Hcompress: Hierarchical data compression for multi-tiered storage environments." In 2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS), pp. 557-566. IEEE, 2020.

[3] Devarajan, Hariharan, Huihuo Zheng, Anthony Kougkas, Xian-He Sun, and Venkatram Vishwanath. "DLIO: A Data-Centric Benchmark for Scientific Deep Learning Applications." In 2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid), pp. 81-91. IEEE, 2021.

[4] Moody, Adam, Danielle Sikich, Ned Bass, Michael J. Brim, Cameron Stanavige, Hyogi Sim, Joseph Moore et al. UnifyFS: A Distributed Burst Buffer File System-0.1. 0. No. UnifyFS. Lawrence Livermore National Lab.(LLNL), Livermore, CA (United States), 2017.

[5] Lack-of-File-System (LOFS), accessed online on Dec 15th, “http://lofs.io/”

Meeting the demands of all I/O workloads all the time through dynamic reconfiguration

Glenn K. Lockwood
National Energy Research Scientific Computing Center
Lawrence Berkeley National Laboratory
glock@lbl.gov

I. Introduction

The relationship between scientific computing and data is being redefined by the confluence of two factors: experimental and observational instruments capable of generating tremendous amounts of data are coming online, and AI-enabled methods of deriving insight from tremendous amounts of data are emerging at an unprecedented pace. Whereas traditional modeling and simulation (hereafter "HPC") **produce** large datasets to fuel new insights, extreme-scale data analysis (hereafter "AI") **consume** large datasets to generate new insights. Unlike the well-structured I/O workloads of HPC, extreme-scale AI workloads also read data in random patterns that are intrinsic to the statistical methods being applied to that data.

Fortunately, technologies exist to meet the I/O challenges of AI. High demand from AI in industry has made solid-state storage an economical option for high random access performance, and this affordable flash has given rise to new breeds of parallel storage systems designed to deliver high random access performance (but not high bandwidth) at scale [1]. That said, there is no one-size-fits-all solution, and trade-offs must be made that either optimize for HPC *or* AI.

In practice, HPC facilities now deploy single, monolithic storage systems that are imperfect for both HPC and AI, making all users equally unhappy. Worse yet, this monolithic model will erode overall productivity as scientific discovery moves towards workflow-driven modes where a single study may require both HPC-like and AI-like steps. In such a future, deploying monolithic storage systems not only makes all users equally unhappy, but precludes any one workflow from achieving optimal performance at every step.

II. An I/O subsystem for the future

In an utopian I/O world, each workflow step would use an I/O subsystem specifically optimized its needs. In reality, giving each workflow step a bespoke storage system is not economically tractable under the traditional monolithic storage system model. However, common hardware technologies–solid-state storage and high-speed networks–can deliver outstanding performance to both HPC and AI workloads. Thus, the matter of delivering bespoke storage systems to each workflow step is actually a matter of software reconfiguration, not hardware economics. As such, we envision *reconfigurability, dynamicism, and software-defined approaches* to be key elements of all future workflow-optimized storage systems.

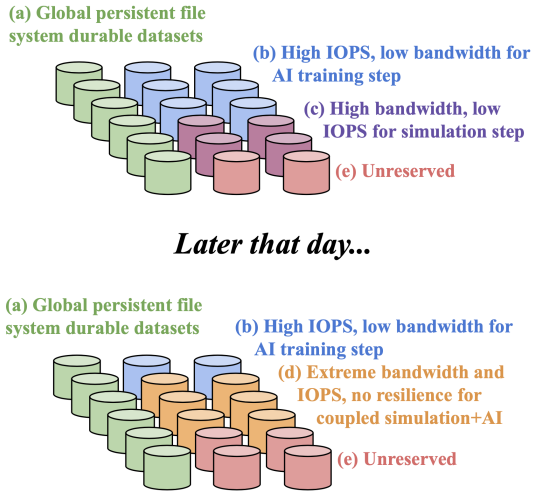


Fig. 1. Notional reconfigurable storage system comprised of (a) a familiar, monolithic parallel file system for unsophisticated users (green) and dynamically reconfigurable elements (b-d) for advanced users. Possible configurations may include (b) IOPS-optimized storage for random-read-heavy AI training, (c) bandwidth-optimized storage for traditional checkpoint-restart, or (d) extreme bandwidth and IOPS storage for exchanging data between AI and simulation within a workflow. The resilience characteristics of each configuration govern their performance optima, and a pool of unreserved storage (e) shrinks and grows as different workflows request different resources.

Such a system is depicted in Figure 1 and is composed of homogeneous storage elements (e.g., servers filled with flash). As jobs enter and exit the compute subsystem, they request ephemeral storage that has the performance characteristics best suited for their I/O patterns. For example, the AI training step of a workflow may be best served by a file system that duplicates data across storage elements to increase read IOPS and decrease tail latency by effectively halving write bandwidth. If the next step of that workflow requires simulation based on the surrogate model that was just trained, it could then request storage that is configured using traditional Reed-Solomon coding to optimize for the streaming read and write patterns common to bulk-synchronous simulation. The physical storage elements remains unchanged throughout, obviating the economic infeasibility of providing workflows with bespoke storage. Instead, software reconfiguration is used to achieve different ratios of bandwidth, IOPS, and resilience based on the requirements of each step in users' workflows.

HPC systems do not implement storage in this way to-

day, but reconfigurable, software-defined storage has a long history in commercial computing. For example, Amazon Web Services offers dynamically provisionable networked file systems through its FSx product, and all cloud platforms offer dynamically provisioned block devices that attach to compute nodes over a high-speed network. Even within the HPC community, components of software-defined storage have been demonstrated. Cray DataWarp allowed users to allocate private parallel file systems to their compute nodes on-demand and choose whether they wanted to trade random write performance for consistency guarantees at job launch [2]. The Mochi project takes this reconfigurability farther by providing a framework for workflows to implement highly optimized storage systems that can be dynamically created and destroyed in userspace [3].

Between the extensive existing commercial and hyperscale technologies and the HPC-specific software-defined technologies built upon them, a rich assortment of configurations for a workflow-optimized reconfigurable storage system already exists. However, these tools and technologies were developed for expert users– software engineers building infrastructure for businesses or sophisticated scientists in full control of their entire workflow's source code. For the average user, this high degree of dynamism, reconfigurability, and optimization adds tremendous complexity.

III. Challenges and opportunities

For a reconfigurable storage system to deliver on the promise of optimal performance for each workflow step, the complexity of the underlying system–dynamically optimizing and moving data–cannot be foisted on end-users. Thus, we foresee several challenges and opportunities.

A. Choice of configurations

Making optimal use of a reconfigurable storage system requires that each user understands both the I/O needs of their entire workflow and the optimal storage configuration to match those needs; this is unrealistic in most cases. Fortunately AI can reduce this complexity by providing sensible defaults and recommendations for non-expert users. For example, a user could simply request a certain capacity of storage along with their compute resources, and an inference engine trained on that user's past jobs would be able to decide the optimal storage configuration for the workflow steps.

Machine learning has been applied to infer areas for optimization for applications and workflows [4]–[6], but more work is required to improve the recall rate of these models and understand what factors cause them to fail. This may require extending I/O profiling tools to capture more information about how data is accessed–not just I/O patterns–which will allow models to better reflect end-to-end workflow behavior. Developing new data fusion methods to integrate user- and system-level telemetry with workflow metadata will also be required to ensure that the storage configurations applied for each workflow step holistically optimize time to solution rather than individual metrics of I/O performance.

B. Data placement

Data placement becomes much more complex in the world of reconfigurable storage because reconfigurability implies ephemerality, and data will have to be moved in order to be persisted. Very much like economics drove HPC centers to embrace tiered storage, the diverging requirements of HPC and AI are driving the fragmentation of the topmost storage tier. Requiring users to track data across ephemeral storage configurations themselves is unduly onerous.

Fortunately, there are several ways in which this problem can be addressed. Many challenges around managing data in ephemeral storage systems were partially addressed during the development of burst buffers; for example, job control language was extended to allow users to declaratively state what data needed to be in what tier at the beginning and end of each job step [2]. This could be extended to allow workflows to declare, for example, that a training dataset must exist in random-read-optimized storage as a condition of job launch, and the resulting model weights must exist in persistent storage as a condition of job completion.

This declarative approach to expressing the data requirements of a workflow will require much richer interfaces for data access [7]. Work performed towards semantically rich data management interfaces must now be rethought in the context of scheduling workflows on dynamic storage systems to ensure that the complexity of ephemeral storage does not hamper scientific productivity. Developing semantically rich interfaces for data access also enables data to be optimally reorganized in transit to match the performance characteristics of the underlying storage configuration with its intended access modes, both reducing complexity and increasing performance. Finally, new semantic interfaces to data bring the additional benefit of enabling passive capture of detailed provenance information for other data management systems as well, underscoring the broad value of developing these capabilities.

IV. Outlook

Many core technologies required by a reconfigurable storage system for HPC and AI already exist–programmable infrastructure, scalable microservices, and software-defined storage are all well utilized outside of HPC. It is therefore important that the HPC community does *not* reinvent what industry has already done. Instead, focusing our efforts on adapting technologies meant for software engineers at tech companies to scientists at ASCR facilities will be the shortest path to advancing data management and accelerating scientific discovery. The future success of modernizing I/O for scientific computing lies in aligning our tools, methods, and infrastructure with the greater currents in the open-source community where possible. Industry has surpassed HPC in leading innovation in many applications of extreme scale computing and AI, and the reconfigurable storage concept outlined here is an adaptation of commercial innovation to suit the needs of HPC. The most critical work ahead lies in ensuring that this adaptation effectively bridges the gap between advanced technologies and the unique needs of scientific users.

REFERENCES

[1] G. K. Lockwood, A. Chiusole, and N. J. Wright, "New challenges of benchmarking all-flash storage for HPC," in *2021 IEEE/ACM Sixth International Parallel Data Systems Workshop (PDSW)*, Lockwood2021, 2021.

[2] D. Henseler, B. Landsteiner, D. Petesch, C. Wright, and N. J. Wright, "Architecture and Design of Cray DataWarp," in *Proceedings of the 2016 Cray User Group*, London, 2016.

[3] R. B. Ross, G. Amvrosiadis, P. Carns, C. D. Cranor, M. Dorier, K. Harms, G. Ganger, G. Gibson, S. K. Gutierrez, R. Latham, B. Robey, D. Robinson, B. Settlemyer, G. Shipman, S. Snyder, J. Soumagne, and Q. Zheng, "Mochi: Composing Data Services for High-Performance Computing Environments," *Journal of Computer Science and Technology*, vol. 35, no. 1, pp. 121–144, jan 2020. [Online]. Available: http://link.springer.com/10.1007/s11390-020-9802-0

[4] J. Lüttgau, S. Snyder, P. Carns, J. M. Wozniak, J. M. Kunkel, and T. Ludwig, "Toward Understanding I / O Behavior in HPC Workflows," in *Proceedings of the 2018 IEEE/ACM 3rd International Workshop on Parallel Data Storage & Data Intensive Scalable Computing Systems (PDSW-DISCS)*, 2018, pp. 64–75.

[5] S. Madireddy, P. Balaprakash, P. Carns, R. Latham, R. Ross, S. Snyder, and S. M. Wild, *Machine Learning Based Parallel I/O Predictive Modeling: A Case Study on Lustre File Systems*, ser. Lecture Notes in Computer Science, R. Yokota, M. Weiland, D. Keyes, and C. Trinitis, Eds. Frankfurt: Springer International Publishing, 2018, vol. 10876.

[6] M. Isakov, E. d. Rosario, S. Madireddy, P. Balaprakash, P. Carns, R. B. Ross, and M. A. Kinsy, "Hpc i/o throughput bottleneck analysis with explainable local models," in *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*, 2020, pp. 1–13.

[7] M. Dorier, M. Dreher, T. Peterka, and R. Ross, "CoSS: Proposing a Contract-Based Storage System for HPC," in *Proceedings of the 2nd Joint International Workshop on Parallel Data Storage & Data Intensive Scalable Computing Systems - PDSW-DISCS '17*. New York, New York, USA: ACM Press, 2017, pp. 13–18. [Online]. Available: http://dl.acm.org/citation.cfm?doid=3149393.3149396

Title: New User Abstractions for Scientific Data Management

Authors: Greg Eisenhauer (Georgia Tech, eisen@gatech.edu, corresponding author), Jeremy Logan (ORNL, lot@ornl.gov), Patrick Widener (Sandia, pwidene@sandia.gov), Matthew Wolf (ORNL, wolfmd@ornl.gov)

Topic: User Data Management Abstractions

Challenge: There is a key need to scale up and scale out scientific data access to match the scaling of computational science applications. Industry has focused its data scaling solutions on large collections of small and mostly independent (log or image) items. Scientific data, conversely, is generally highly interdependent and rich in not only current but future connections. Data storage, transmission and retrieval is of course a common need in all types of computing, but it's particularly critical in the HPC space where inefficient I/O is a significant impediment to efficient utilization of HPC resources. There has been a lot of prior work that attempted to create data models that were broadly useful, ranging from simply serializing data into POSIX directory and file structures, using HDF5's hierarchical models[2], ADIOS's PGAS-like model[3], or even the tuple space solutions proposed by Linda[1], DataSpaces[4], or more recently Unity[5]. These are all good working solutions to address specific short-term issues in scientists' workflows, but none are a complete solution.

It has become critical to rethink the user-level abstractions and storage and I/O technologies we use for scientific data. As we look forward to large and complex datasets being fed into larger and more complex workflows that contain ML, simulation, and even experimental control aspects, operations at the level of "read" and "write" are at an unsatisfyingly low level of abstraction for interacting with scientific data. As a result, modern scientific data management solutions have been forced into a number of uncomfortable trade-offs. At the performance level, the POSIX tape-based access patterns give high performance only to specific read patterns. This serialization for performance forces libraries and end users to adopt data structure and metadata tagging on "write" that may be unclear, poorly maintainable, fragile, and even pathological for particular later access patterns, such as AI/ML training for digital twins.

Opportunity: Extending the publish/subscribe design pattern provides an opportunity to match the abstractions we use for interacting with scientific data. Starting from publish/subscribe rather than POSIX read/write, we can extend the abstraction to surface more of the lifecycle of data, exposing publication, use, reuse, deletion, and validation phases which have mostly been hidden from users. Furthermore, exposing that lifecycle allows metadata, which is subject to significantly different usage patterns, to be managed using the same abstractions but with tailored strategies. Scientific data should be considered as an archive of high performance data events, with generation, annotation, optimization, annotation, curation, and consumption of those events representing key stages of the data lifecycle.

While the details of future data models and APIs that might support this type of data management would be the subject of research, there are signposts that may indicate useful directions. For example, model-based program analysis can help us understand how data is produced and used, capturing significant semantic and provenance information, but that

information is difficult to represent as part of existing metadata representations. Directories and files, while otherwise restrictive, can provide useful context (separating this version of data from that, this user's runs from others) and protection (via file and group permissions) that isn't intrinsic in all other models. Some features of publish/subscribe approaches, like reader-specified data filtering, are a natural fit for emerging hardware capabilities like fabric-attached computing that lack a semantic foothold in read/write models. Taken together, such snippets of useful features from different approaches are at least indicative of functionality that has proven valuable and might be replicated in a new approach.

We believe that there is a significant opportunity in this space to move toward a fusion of storage, database, and internet-scale data management techniques by refocusing on making the data elements inherit explicit tags and contexts through channels, sub-channel designations, perhaps even literal hashtags. In some ways, this vision is more metadata-focused than data focused, because derived and implicit metadata capture upon write and the use of metadata to determine data routing and delivery would be of primary importance. However, ML-based learning techniques should be useful to improve the use of that metadata for optimizing and customizing data delivery.

Timeliness: Scientific data is already facing pressure to scale up and scale out, serving as the glue in workflows which use simulation and analytics in interchangeable ways. The space of large-scale data storage solutions has become increasingly diverse as cloud-based object storage and NVM approaches are increasingly integrated into mainstream computational science. This collision of storage and computational diversity has exposed assumptions, relied upon by current data management approaches, whose cost-benefit implications have become unfavorable. Now is an excellent time to revisit those assumptions and develop data management abstractions which reflect the semantics of how data is used rather than how it is stored.

Bibliography:

1. D. Gelernter. 1985. Generative communication in Linda. *ACM Trans. Program. Lang. Syst.* 7, 1 (Jan. 1985), 80–112.
2. M. Folk, G. Heber, Q. Koziol, E. Pourmal, and D. Robinson. 2011. An overview of the HDF5 technology suite and its applications. In *Proceedings of the EDBT/ICDT 2011 Workshop on Array Databases* (*AD '11*). Association for Computing Machinery, New York, NY, USA, 36–47.
3. Q. Liu, J. Logan, Y. Tian, H. Abbasi, N. Podhorszki, J.Y. Choi, S. Klasky, R. Tchoua, J. Lofstead, R. Oldfield, M. Parashar, N. Samatova, K. Schwan, A. Shoshani, M. Wolf, K. Wu, W. Yu. 2014. Hello ADIOS: the challenges and lessons of developing leadership class I/O frameworks. *Concurrency and Computation: Practice and Experience*, *26*(7), pp.1453-1473.
4. C. Docan, M. Parashar, and S. Klasky. 2010. DataSpaces: an interaction and coordination framework for coupled simulation workflows. In *Proc. of the 19th ACM International Symposium on High Performance Distributed Computing* (*HPDC '10*).
5. T. Jones, M. J. Brim, G. Vallee, B. Mayer, A. Welch, T. Li, M. Lang, L. Ionkov, D. Otstott, A.Gavrilovska, G. Eisenhauer, T. Doudali, and P. Fernando. "UNITY: Unified Memory and File Space" In Proceedings of the 7th International Workshop on Runtime and Operating Systems for Supercomputers (ROSS'17).

Intelligent data subsystems for converged AI and HPC workloads

Guojing Cong, Steven Young, Robert Patton, ORNL {congg,youngsr,pattronrm}@ornl.gov

Topics: AI enhanced data subsystem, data system connecting HPC, cloud, and edge systems

Challenge: AI techniques are being employed in large-scale simulations to accelerate scientific discovery. As a result, high-performance computing (HPC) simulations increasingly take on characteristics of big data analytics. With this emerging paradigm where intensive computing produces massive amount of data and in turn the data is mined to accelerate computing, traditional parallel file systems can be easily overwhelmed with dynamic data access patterns resulted not only from simulations but also from on-line training and inferencing.

Figure 1 shows the average epoch time to train ResNet50 with ImageNet-1K with a 256-node system. Each node has 4 GPUs, and the dataset is stored on the parallel file system. As the number of GPUs increase from 2x4 to 8x4, training exhibits near linear scaling. As the number of GPUs increases to 64x4 and beyond, training time is dominated by waiting for data as parallel I/O becomes the bottleneck. At 1024 GPUs, almost all epoch time is spent on waiting for data.

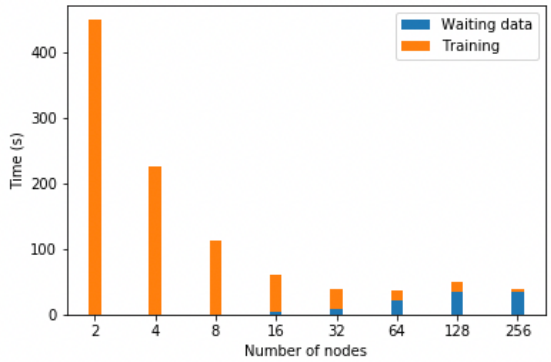


Figure 1. Average epoch time to train ResNet50 with Imagenet-1K dataset in different scales on an HPC system. The epoch time stopped decreasing when the data loading overhead stopped scaling [1]

Opportunity: The architectures of current and next generation supercomputers are designed for both HPC and AI applications. Opportunities abound in effectively leveraging such platforms to develop the next generation intelligent data subsystem that provides: 1. efficient I/O support such as data prefetching, staging, and exchanging for classic simulations, AI applications, and AI-augmented simulations, and 2. additional data services through modern AI techniques for efficient data organization, compression, retrieval, regeneration, sampling, archiving as well as state cataloguing and rare event detection.

Supporting converged HPC and AI workloads: Classic simulations tend to exhibit regular data access patterns that can be learned by AI models. Predictions in advance can help optimize I/O performance. Deep learning applications in the training phase exhibits irregular data access patterns that are challenging to learn and predict, but optimizations specific to these applications are still possible [1] through higher-level interface between application and data subsystem. For example, with stochastic gradient descent, only some kind of guarantee of

random sampling of data [2] is needed, and the application does not necessarily need to specify a complete ordering of data items. AI-augmented simulations bring the challenges of both classic simulations and deep learning applications. A powerful data subsystem is necessary to meet the needs of a diverse landscape of applications, and at the same time it needs to take full advantage of technologies such as RDMA, NVMe, user-level file systems, GPU-direct on leadership computing facilities.

Enhanced intelligent data services: In light of increasingly "intelligent" applications [3], new functionalities are desired of the data subsystem. Many will be made possible by the rapidly evolving AI techniques. For example, auto-encoders and generative models bring new answers to questions such as what data to archive, how to archive them, how to retrieve them (or regenerate them), and how to combine archived data with on-line data. With such techniques, not only data can be compressed and regenerated, it is also possible to generate new data that did not exist before. We expect compression, dimensionality reduction, state cataloguing and rare event detection become standard services offered by the next generation intelligent data subsystems. Standardized services will greatly improve the productivity of practitioners and performance of large-scale applications. To provide intelligent services, AI algorithms that handle high-dimensional, feature-rich scientific data are necessary. Neural network models for scientific problems and auxiliary data from scientific simulations demands research attention and investment to realize the full potential of AI to science.

Connecting edge systems and cloud: As more computing moves to the cloud and edge systems, large-scale simulations produce data that benefit other similar applications, and at the same time, it can benefit from insights gained from applications that run on public cloud or dedicated clusters. An intelligent, flexible, and scalable data subsystem is key to the success of future computing paradigms.

Timeliness: Progress in this area lays foundations for converged HPC workflows and data analytics that are urgently needed by several research efforts tackling grand-challenge problems with tremendous social, economic, and environmental impact. Taking a systematic approach towards the new paradigm is necessary to produce long-lasting software utilities are meet the common needs of these projects. It brings rigorousness and efficiency that is otherwise lacking in ad-hoc per project effort.

References

1. "*Accelerating Data Loading in Deep Neural Network Training*", Jamie Yang and Guojing Cong, IEEE International Conference on High Performance Computing, Data, and Analytics, HiPC 2019
2. "*Partial data permutation for training deep neural networks*", Guojing Cong and Li Zhang, 20th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGRID20
3. Enabling AI-Accelerated Multiscale Modeling at Millisecond and Molecular Resolutions on Supercomputers", YiCong Zhu, Peng Zhang, et al., International Supercomputing Conference (ISC 2021)

Direct Support for Indexing Metadata

H. Lee Ward
lee@sandia.gov
Sandia National Laboratories, dept. 1423
Related topics; Storage-system architecture design, Metadata management infrastructure to support FAIR principles, Capturing provenance information, Providing data management support for AI and complex workflows

Machine learning, data mining, and other applications often bring a requirement that metadata be present in some index service or arrange their own. This can be, often is, accomplished using a database of some form. At a minimum, seemingly, a key-value store is required as a basic building block.

Using such databases are awkward and cannot by themselves maintain relationships between the (meta)data they maintain and the related source data, for instance. Any tool we might contemplate to extract or derive metadata from source data needs to connect to and interact with a database. This can be awkward as it ties the tool to the database choice at some level and because the source might not also reside within the database. It becomes a kind of bridging service, then, between a bulk-store storage system and the metadata storage. An unreliable bridge, though, in that it constrains, but cannot itself enforce how the source data is organized within the name space of the storage system but must utilize that name space. For instance, it is all too easy to consider a scenario where the source data is perused and the interesting metadata captured within a database then later, through mistake or ignorance the source data is moved, or removed, so removing the implicit or explicit link between the derived metadata collection and the actual data it describes.

These issues motivate storage system support of, first, the ability to link existing storage system API providing user-level metadata function to an external indexing service and, second, the ability to expose storage system namespace changes as events to that, or other, services.

The ability to link existing storage system API metadata function to an external service would allow tools to leverage that API to deposit, maintain, and manage the interesting metadata in a way that is completely divorced from the indexing service, or database, choice. For instance, existing POSIX file API provides such metadata capabilities that tools can, and do, use. What such a choice fails to provide, though, is the ability to mine all the metadata on the storage system without also enumerating the POSIX namespace. If the storage system exposed the existence, content, and related authorization information to an external indexing service then this metadata could be gathered, maintained, and managed in a way that enables fast, system-wide, search.

Such a change also enables reliable maintenance of the semantic link between the derived metadata and the source data. For instance, if a user modifies, removes, or alters the source data it would be possible for the external service to react. It could react to these events by one or more of capturing, intact, a previous version of the data via copy, versioning, or snapshot, or removing the associated key-value pairs within the global index, or reassociating the associated collection with the new location of the source data.

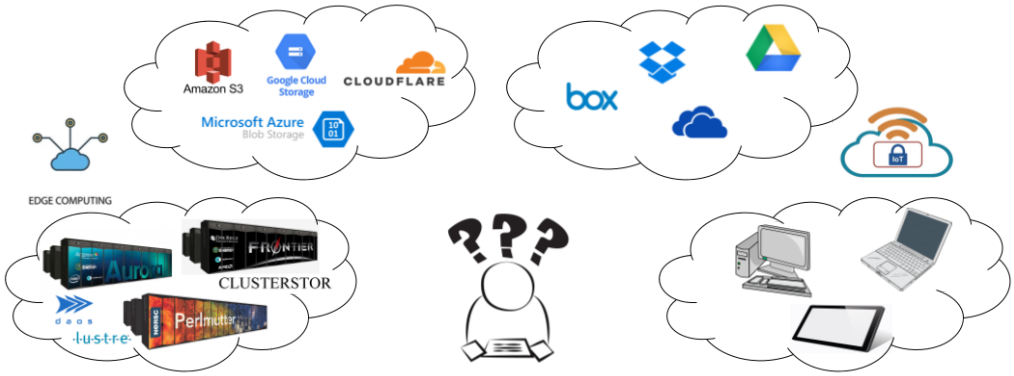
Such a system would provide for, at least, automated maintenance and management of a volume-wide metadata index and remove the need for tools to be aware of how that index is maintained. Capturing and accurately maintaining metadata has long been a problem and this idea seems ripe, on it's face, to make a decent stab at beginning to address that.

Scientific Data Access Without Borders

Houjun Tang (htang4@lbl.gov), Jean Luca Bez (jlbez@lbl.gov), Qiao Kang (qkang@lbl.gov)
Topic: Data access interfaces, FAIR principles, Provenance, Data movement performance and tuning

Challenge

Scientific data management continues to be a challenge today and has been on track to become even more complex with the ever-increasing heterogeneity of storage systems, as shown in the figure below. Traditional scientific applications are designed with a single storage location in mind, interacting only with the file systems collocated on the computing system where data is generated. However, as mentioned in the MSSD pre-workshop document [1], there has been a growing trend that scientific data resides not only in where it is generated, but also in various cloud services, workstations, laptops, and even edge and Internet of Things (IoT) devices (as shown in the figure below). To analyze the data, users often have to manually move the data from multiple data sources to a single location before accessing it. For instance, a typical scientific workflow using HPC resources is as follows: data are collected by edge devices or generated by simulation codes in an HPC center, scientists then analyze the data on a different (often smaller) cluster or their workstations, and then publish or share the processed data as well as analysis results to the community on public cloud storage. Data sharing can be a barrier as it can take a significant amount of time when the data is large and/or there are a large number of files involved. In certain cases, it may also be impractical due to storage space constraints. In addition, it could also result in a loss of metadata information when the data is transferred multiple times, making it challenging to maintain the principles of findability, accessibility, interoperability, and reusability (FAIR).



Opportunity

Future scientific data management systems must focus on operating in “virtual data facility” scenarios that provide seamless data access across heterogeneous storage systems that are located in various computing environments (edge, cloud, supercomputing, laptops, etc.) while maintaining the FAIR principles. A novel data management system should provide simple and optimized data and metadata access methods to ingest and retrieve data, select the most appropriate storage location automatically, and maintain machine-readable and query-able rich metadata. Such a system should be capable of adding new data and metadata, and sharing the data along with their metadata among users. These capabilities remove the barriers of data access and sharing, allowing scientists to focus

more on their scientific discovery instead of spending time to organize, memorize, and transfer their data manually. By hiding all the unnecessary details about the storage locations and device characteristics, it could significantly increase scientific research productivity. We propose three research directions for managing scientific data without borders:

1. Seamless data access APIs for scientific productivity - Scientific data users should only care about the data at an abstract level (i.e., structured and unstructured grids, multi-dimensional arrays, tensors, etc.) itself instead of where the data is located as long as they have permission to access. A "virtual facility" views data as objects and abstracts away their storage locations across multiple environments and the low-level details of the storage devices. Objects can be stored in a flat namespace and be located with a unique identifier or through querying on metadata. Once access credentials are provided and verified, users can perform various operations, such as "post/create", "put/update", "get/read", and "delete", on one or a group of objects with popular programming languages, such as C/C++, Python, R, etc., and the data management system will determine where to put or retrieve the data.

2. FAIR metadata management for fast discovery of data and trustworthy data - Metadata and provenance play a major role in helping scientists locate data they wish to analyze. The "virtual facility" manages the rich metadata along with the data to help scientists deal with the increasing volume, complexity, and creation speed of data. It also automatically records and maintains the provenance of the data as part of the metadata management process. All the metadata is easily searchable and can be automatically shared with their associated data objects, making it understandable and reusable by the community.

3. Runtime system with active monitoring and optimization for efficient data movement - Since efficient data movement is critical to the performance of data ingestion and retrieval, understanding data accesses and their performance in the different stages of data management in a "virtual facility" is crucial to automatically and transparently detect bottlenecks and to guide data movement optimizations without end-user involvement. This requires defining data movement and I/O metrics across multiple environments, monitoring these metrics actively with low overhead, and guiding optimization methods proactively. Existing solutions for monitoring and optimization should be accessible with an API and account for the data ingestion, management, and retrieval stages, including data placement and movement, and also recording contextual information, with low overhead, regardless of the physical location of the data.

Timeliness

The efforts to provide seamless data access methods and maintain FAIR principles allow scientists to focus more on their science by getting the data to where it is needed in an efficient, timely, and easy-to-manage way in the era of "scientific big data".

Reference

1. Kathryn Mohror, et. al. "ASCR Workshop on the Management and Storage of Scientific Data", https://www.orau.gov/MgmtStgeonScDataVersion

Data Management for Scientific Artificial Intelligence Workloads

Huihuo Zheng and Venkatram Vishwanath
huihuo.zheng@anl.gov
Argonne Leadership Computing Facility, Argonne National Laboratory

December 15, 2021

1 Data management support for AI and complex workflows

Artificial intelligence (AI) and Big data analytics are two pillars working closely with the modeling and simulation pillar for scientific discoveries. Leadership computing facilities have also been actively embracing these emergent workloads for upcoming and future systems. We expect these workloads to occupy a significant fraction of the time on these systems and effective data management to support AI training and inference will be needed.

We are witnessing several novel AI chips being architected to accelerate workloads. These include systems such as Cerebras, Groq, and SambaNova, and are being evaluated for science by leadership computing facilities [2]. These architectures makes the HPC systems more heterogeneous and the workflows more complex - One involving challenging data coupling between AI and traditional scientific simulation [4] and on heterogenous systems.

The data management and I/O patterns of AI workloads are different from traditional simulation workloads. In the exascale era together with novel AI hardware, we will need to systematically reconsider the data management and storage landscape to meet the science needs. In this position paper, we will first discuss two key challenges: **(1) AI training at exascale**; **(2) AI and simulation coupled workloads**. We will also discuss research thrusts spanning I/O and storage libraries, profiling needs and benchmarking efforts needed in the next 5 - 10 years in order to address the challenges.

2 Challenges

2.1 Data management challenge for AI training at exascale

- **Parallel I/O Interfaces**: AI Framework libraries such as Tensorflow (TF) or PyTorch (PT) use POSIX I/O for reading and writing datasets. For example, TensorFlow data pipeline is optimized for its native tfrecord format, and this does not support parallel I/O. Given the challenges with POSIX I/O semantics, we expect scaling challenges for applications using this API. There is a need to design new interafces for AI frameworks to fully utilize parallel I/O to meet the data scales expected. Additionally, scientific datasets in formats such as HDF5, ADIOS and NetCDF are not natively supported by AI frameworks
- **Metadata intensive, small and sparse I/O operations**: Traditional parallel file systems such as Lustre and GPFS do an excellent job at supporting large bulk I/O. AI workloads typically involve intensive metadata operations with a lot of small, sparse and random I/O. On Lustre, the metadata performance is a key bottleneck. There is a need to re-architect file systems to support scalable metadata operations together with small and sparse I/O.
- **Complex data format**: AI datasets are in various kinds of file formats, including json, text files, image files, unstructured data formats such as key-value store database. There are no effective parallel I/O APIs to access these datasets at scale.
- **Software support for storage hierarchy**: Node-local storage and burst buffers have been proposed to bridge the performance gap between memory and the parallel file system. There is a need to develop I/O libraries or methods to efficiently incorporate these novel hardware architectures into the data movement pipeline, for example, utilizing intermediate storage layers for caching / pre-staging the datasets to liberate the I/O pressure on the parallel file system. There are other layers in the entire storage hierarchy, such as, CPU memory, GPU memory. There is a need to develop a unified API to manage data movement across different tiers in a transparent manner.

2.2 Data management challenge for AI and simulation coupled workloads

The key challenge for these workloads is the data coupling and movement between AI training and inference stages with an HPC simulation. The data exchange can be done through the storage. For example, SmartSim uses a database [1], where new samples are committed to the database on the fly from a simulation. One can run an AI inference or training on samples in the database similar to how we deal with streaming databases. However, future workflow are expected to much more complicated. The AI and simulation may be executed completely on two computing systems, perhaps at two different computing centers.

There is a lack of infrastructure and software support for data movement and streaming. In particularly, there is a need for integrating / interfacing different hardware and software components involved in the data movement, such as the workflow manager, AI framework data pipeline libraries, scientific I/O libraries (HDF5, NetCDF, ADIOS) used by the simulation for storing data, as well as inter-facility data transfer infrastructure such as Globus. The data movement and streaming may involve different types of storage, such as memory, node-local storage, and global file systems; therefore, there is a need to develop high level abstractions to simplify the data management.

3 Opportunity

We identify the following three research areas to focus on to overcome these challenges:

- **Optimizing data movement to support complex coupled workflows**: Data movement will need to account for the deep memory hierarchies with diverse capacity and capabilities, together with the underlying interconnect topology interconnecting various subsystems. At the same time, another dimension we need to consider is the compute and memory requirements of the coupled processes, the rates at which they produce and consume data, the availability of leveraging in-network compute, among others. We will need a programming model wherein these application requirements can be expressed in a declarative manner and have a runtime that optimizes this multi-objective optimization taking into account both the application needs and the underlying system characteristics and, thus, removing this burden from the application developer.
- **New parallel I/O for support AI frameworks**: For the AI training and inference at scale, we need to tackle this again in two separate though synergistic dimensions. First, we will need to augment the underlying deep-learning frameworks, such as Tensorflow(TF) and PyTorch, to fully exploit parallel I/O. For instance, we will need to augment the TF dataset API to go beyond its support for POSIX I/O and embrace parallel I/O. Next, given that several scientific datasets are in formats such as HDF5, NetCDF, and ADIOS, we will need to augment the frameworks to support these formats natively.
- **Holistic Profiling and Benchmarking new workloads**: As the workflow become increasingly complex, it is crucial to develop tools and benchmarks that can profile data movement to understand the bottleneck and identify avenues for improvement. These include: (1) Developing holistic tools to profile the data movement at different levels spanning AI, Memory, Compute, Storage, and Interconnect. VaniDL is one such example [3]. (2) Profiling and benchmarking existing AI workloads running on different leadership facilities will help us identify I/O patterns for different kinds of scientific workloads. DLIO [3] is one early work toward this vision. (3) Study the I/O performance of different file systems / storage solutions for diverse science, and automate the organization and layout of these datasets on existing and aid in the design of future systems.

4 Timeliness

Given the increased adoption of AI for Scientific computing on leadership systems, we are currently faced with a challenge to support AI Data management. The time is ripe to make significant investments for science at scale.

References

[1] SmartSim documentation. https://www.craylabs.org/docs/overview.html.

[2] ALCF. AI testbeds, 2021. https://www.alcf.anl.gov/alcf-ai-testbed.

[3] Hariharan Devarajan, Huihuo Zheng, Anthony Kougkas, Xian-He Sun, and Venkatram Vishwanath. Dlio: A data-centric benchmark for scientific deep learning applications. page 81–91, May 2021.

[4] Anda Trifan et. al. Intelligent resolution: Integrating cryo-em with ai-driven multi-resolution simulations to observe the sars-cov-2 replication-transcription machinery in action. Oct 2021.

Next Generation Indexing and Search in Large-Scale Scientific Storage Systems

Ioan Raicu[1], Alexandru Orhean[1], Kyle Chard[2], Lavanya Ramakrishnan[3], Anna Giannakou[3]
[1]Illinois Institute of Technology, [2]University of Chicago, [3]Lawrence Berkeley National Laboratory
iraicu@cs.iit.edu, aorhean@hawk.iit.edu, chard@uchicago.edu, lramakrishnan@lbl.gov,
agiannakou@lbl.gov

Topic: Scalable Data Indexing and Search

Challenge

Rapid advances in digital sensors, networks, storage, and computation coupled with decreasing costs is leading to the creation of huge collections of data—commonly referred to as "Big Data." This data has the potential to enable new insights and discoveries that can change the way business, science, and governments deliver services to their consumers and can impact society as a whole. Increasing data volumes, particularly in science and engineering, has resulted in the widespread adoption of parallel and distributed filesystems for storing and accessing data efficiently. However, as filesystem sizes and the amount of data "owned" by users increase, it is increasingly difficult to discover and locate data amongst the petabytes of accessible data, with exabytes of storage capacity on the horizon. While much research effort has focused on the methods to efficiently store and process data, there has been relatively little focus on methods to efficiently explore, index, and search data using the same high-performance storage and compute systems.

One of the most significant burdens faced by the scientific community is the lack of efficient tools that enable targeted search and exploration of large file systems. While it is now trivial to quickly find websites from the billions of websites in existence, it is difficult for researchers to search data in their scientific data stored on large-scale storage systems. Google has pioneered much of the information retrieval and search engine research; however, its area of focus is large-scale distributed search over web data rather than searching over scientific data stored in high-performance file systems—two areas with significantly different data, storage, processing, and query models. In the enterprise search domain there are several tools that are commonly used to enable scalable search, such as Apache Lucene [1], Apache Solr [6], and ElasticSearch [4]. While these solutions were designed to work well on commodity hardware, they are not designed to make use of the advanced features of HPC systems, and are typically tightly coupled with some sort of distributed file system, such as Apache Hadoop File System [7] not commonly found in HPC. Many of these Apache projects have been implemented in Java, which also has not garnered wide adoption in HPC systems. Other existing works from the HPC domain have also aimed to tackle this problem, however they typically have focused on indexing and search of metadata [2, 5] as opposed to the scientific data itself.

Opportunity

In the absence of better options, scientists often fall back to the state of the art methods for finding data in single, centralized systems. That is, they use traditional Linux tools: ls and grep, or find. However, these utilities are not designed for large file systems. For example, listing all files (a common operation when searching for a specific file name) in a production parallel file system commonly found on large computing clusters could take many weeks to complete (given that it contains billions of files and metadata performance is typically limited to 1000 operations per second). Further, this does not consider the time to read the data itself, a task that could compound the search time by several orders of magnitude. Searching through a 10 petabyte file system (the size of the persistent storage system on the Theta supercomputer at Argonne) by reading through the entire data could take over 3 years at a modest 100MB/sec read rate.

Classical tools are not suitable in the context of large-scale storage systems. We believe that tools which allow data and metadata stored on today's HPC storage systems (e.g. Lustre, GPFS, Ceph, Globus [3]) should be index-able and search-able in a transparent effortless way while not impacting the performance of the storage system for I/O intensive workloads. Scientific data comes in many flavors, from free-text data (e.g. logs in text files), to numerical data (e.g. matrices in HDF5, time-series data), to image data (e.g. medical images in DICOM format), to video data (e.g. videos from biology studying organisms behavior). Each data type might need specialized indexing and search methods, further complicating an already difficult problem at scale. Due to the sheer amount of data found in today's HPC storage systems, any solution must be distributed, be parallel in nature, and use recent advancements in non-volatile memory.

Timeliness

This work is transformative due to its radical distributed architecture for data organization, indexing, and search, enabling fundamental long-missing functionality to large-scale scientific computing storage systems. There are unique challenges in indexing and searching large-scale scientific storage systems, as they can hold petabytes to exabytes of data spread out over billions of files, and the computing systems attached to these storage systems can produce an avalanche of data (e.g. terabytes per second). In parallel and distributed file systems, and more specifically in scientific computing, there is no scalable search system that is able to efficiently and automatically index and organize data. Scientists can explicitly read or write specific data by using file names, but when file systems contain millions to billions of files in a complex and deep directory hierarchy, finding data is analogous to looking for a needle in a haystack. Storage systems in high-performance computing have become so large that many scientists have to spend a significant part of their work to organize and catalog their scientific datasets in order for them to find their data at a later time. This departs from the traditional brute force data searching (saving a significant amount of time) or the explicit a priori creation of specialized catalogs (saving a significant amount of money).

While it is now routine to search for data on a personal computer or discover data on the Internet at the click of a button, there is no such equivalent method for discovering data on large parallel and distributed storage systems. There is an urgent need to develop new methods to support search in large storage repositories in scientific computing systems. While some dedicated scientific communities have developed specialized catalogs and tools that aid discovery, such approaches are limited in terms of generalizability, and they are often cumbersome to use due to imposed schemes and the need for manual data wrangling.

Storage technologies have seen rapid improvements in the past decade as non-volatile memory has increased per device throughput and latencies by orders of magnitude through the introduction of NVMe and DDR4 interfaces. High-end computing hardware that uses many-core architectures with multiple storage devices with fast network interconnects along with open-source storage systems such as Lustre, Ceph, and Globus, have made the timing the development and deployment of such tools that could be deployed as infrastructure services much more likely to succeed than at any point in the past. These tools will revolutionize the heroic effort invested by the scientific community to organize large data repositories and make them accessible to the scientific community at large, touching every branch of computing in high-end computing. These advancements will strengthen a wide range of research activities enabling efficient access, processing, and sharing of valuable scientific data from many disciplines.

References

[1] BIAŁECKI, A., MUIR, R., INGERSOLL, G., AND IMAGINATION, L. Apache lucene 4. In *SIGIR 2012 workshop on open source information retrieval* (2012), p. 17.

[2] BONNIE, D. J. Gufi overview. Tech. rep., Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2018.

[3] CHARD, K., FOSTER, I., AND TUECKE, S. Globus: Research data management as service and platform. In *Proceedings of the Practice and Experience in Advanced Research Computing 2017 on Sustainability, Success and Impact*. 2017, pp. 1–5.

[4] GORMLEY, C., AND TONG, Z. *Elasticsearch: the definitive guide: a distributed real-time search and analytics engine*. " O'Reilly Media, Inc.", 2015.

[5] PAUL, A. K., WANG, B., RUTMAN, N., SPITZ, C., AND BUTT, A. R. Efficient metadata indexing for hpc storage systems. In *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)* (2020), IEEE, pp. 162–171.

[6] SHAHI, D. Apache solr: an introduction. In *Apache Solr*. Springer, 2015, pp. 1–9.

[7] SHVACHKO, K., KUANG, H., RADIA, S., AND CHANSLER, R. The hadoop distributed file system. In *2010 IEEE 26th symposium on mass storage systems and technologies (MSST)* (2010), Ieee, pp. 1–10.

Intelligent Discovery and Delivery of Scientific Data using Knowledge Networks

Ivan Rodero, Daniel Balouek-Thomert, Manish Parashar
SCI Institute, University of Utah
{ivan.rodero, daniel.balouek, manish.parashar}@utah.edu

Related topics: Interfaces for accessing data;
Metadata management infrastructure to support FAIR principles

Challenge

The rapidly increasing variety, scales, resolutions, and availability of observational, experimental, and computational data, such as that provided by DOE large facilities and observatories, provides the potential for new insights and new opportunities for addressing the scientific and societal challenges of the 21st century. This availability of data is leading to complex data-driven workflow formulations targeting discoveries across domains and datasets, such as those envisioned by the NERSC Superfacility. These workflows aim to compose and analyze heterogenous data from multiple sources, disciplines, and domains, for example, multi-user scientific observatories, instruments, and experimental platforms. They are increasingly leveraging computational capabilities across the continuum, from high-end systems to the edge.

This growth in the scale and variety of the data coupled with a growth in the number of users accessing this data is resulting in new challenges associated with ensuring that the data can be discovered, broadly accessed, integrated, and analyzed in a timely manner. For example, facilities often provide data-download portals for data and data-product discovery and access. However, these tend to be highly domain-specific and can limit discoverability, integration into multidisciplinary workflows, and overall access to these resources by the broader science community.

Recent years have also seen significant advances in capabilities aimed at increasing access to commercial data and data services as well as the federation or resources (e.g., NERSC Superfacility, Open Science Grid). However, current solutions still require users to have certain domain knowledge about the facility to be able to effectively find the data they need and do not provide capabilities to help users discover new data and products that they do not know a priori. Furthermore, these solutions are not equipped with intelligent discovery and anticipatory delivery of data and data products from large facilities to the users anytime, anywhere. Consequently, more effective and timely data discovery and access for composing workflows from multiple sources are urgently needed for large facility-enabled science. These capabilities can be built leveraging existing technical solutions as described below, addressing the urgency of connecting data sources, computing environments, and interdisciplinary researchers to unleash the full power of data and AI to support science workflows across sectors with significant societal impacts.

Opportunity

The collective potential of the emerging data and computational ecosystem for scientific advances, the associated challenges outlined above, and current trends in intelligent data-science, present opportunities to rethink how data is discovered and accessed. Specifically, one can envision an intelligent data discovery and delivery framework supporting the broad discovery, timely delivery and FAIR use of science data. Such a framework would be composed of user query analysis techniques to model access patterns and associated localities and affinities, optimized data caching, data prefetching, and data steaming mechanisms to support optimized push-based data delivery and a data recommendation framework based on the knowledge network model described above to facilitate data discovery (see Figure 1).

Furthermore, as integrating knowledge from many sources can quickly grow in scale, it can leverage the distributed nature of the knowledge network to develop a parallel implementation along the computing continuum based on artificial intelligence techniques. Therefore, such a framework can be exploited from the edge (e.g., data quality from instrumentation) to high-end system resources (e.g., real-time data assimilation for digital twins that support critical management decisions). It revolves around the key innovations illustrated in Figure 1 and has the potential to deliver a new federation technology to enable data-driven scientific workflows. For example, leveraging concepts such as open knowledge networks, knowledge extraction, and

graph theory [1], coupled with CI components and data management techniques, can be used to implement intelligent data discovery and push-based data delivery. It goes beyond existing history-based methods that are the most commonly used approach for web prefetching [2].

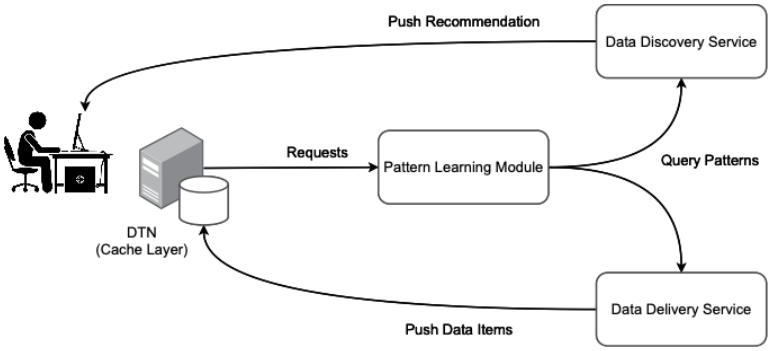


Figure 1. Overarching architecture of the envisioned intelligent data service framework

The path forward to improve data access and delivery methods include distilling knowledge exploring the connection between the facilities' data and their associated research, which is natural in human learning. For example, the concepts underlying recommender systems, which are extremely effective in e-commerce, can be leveraged in connection with science projects and domain-specific research. The co-design of the software framework with a focus on service composability in support to end-to-end application workflows can enable the integration and orchestration of a wide range of existing CI components for optimizing data management in support of anticipatory delivery of data and data products thus reducing data transfer, latency and time to solution.

Timelines or maturity

The growth of complex, distributed computing ecosystems that couple simulation, observation, and simulation will provide significantly new frontiers for science. Large facilities are already in production (e.g., Large Hadron Collider (LHC), Spallation Neutron Source (SNS), Advanced Light Source(ALS)), and others are under construction (e.g., ITER, Second Target Station at SNS, upgrades to ALS, etc.). In addition, exascale systems are nearing deployment (Frontier and Aurora) that will become key pieces in a scientific computing ecosystem.

The emergence of a distributed ecosystem composed of rich, diverse, and connected data sources (such as instruments, experimental facilities, and observatories) presents new opportunities for data-driven scientific discovery and innovation. An intelligent data discovery and delivery framework that leverages the state of the art in intelligent data science can ensure the data is FAIR and broadly accessible and that these opportunities are effectively and scalably realized. Furthermore, preliminary work based on a collaborative knowledge graph to represent the auxiliary information extracted from facility metadata, domain knowledge, and history of user requests [3] has demonstrated the feasibility (and benefits) of this approach leading to a significant reduction in data access and data transfers from facilities as compared to current approaches.

References

[1] X. Wang, et al., “KGAT: Knowledge Graph Attention Network for Recommendation,” 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, pp. 950-958, 2019.
[2] L. Xiong, Z. Xu, H. Wang, S. Jia, L. Zhu, “Prefetching scheme for massive spatiotemporal data in a smart city,” Int. J. Distrib. Sens. Netw. 12 (1), 2016..
[3] Y. Qin, I. Rodero, M. Parashar, “Toward Democratizing Access to Facilities Data: A Framework for Intelligent Data Discovery and Delivery,” arXiv:2112.06479, 2021.

The Need for Portability: Unified Interfaces to Access Scientific Data on Blurred Memory and Storage Tiers

Ivy Peng (peng8@llnl.gov), Roger Pearce (pearce7@llnl.gov) and Maya Gokhale (gokhale2@llnl.gov)
Lawrence Livermore National Laboratory
Topics: Interfaces for accessing data

Challenge:
The dichotomy between memory and storage, originating with the historical orders of magnitude performance gap between the two, is manifested in a dichotomy of representation between in-memory and in-storage data structures. While non-volatile memory technology advances combined with sophisticated hardware and software cache algorithms has greatly reduced the gap, advances in scientific data management software libraries have lagged.

On current systems, frequent conversions between in-memory representation and storage-level format are needed in scientific data management. The state-of-the-art practice stores in-memory data from scientific simulations into persistent storage and retrieves data from storage to memory for processing. In practice, when data sizes in storage often exceed main memory, more conversions may be needed for out-of-scale processing. For instance, querying and processing scientific data could benefit from domain-specific metadata management, custom information indexing in object storage, such as key-value stores and application-tailored databases. However, infrastructures for building such object stores are not widely available, and their implementations are mostly based on file storage, which requires conversions between in-memory representation and storage-level formats.

Meanwhile, emerging memory and storage architecture has blurred boundaries. For example, fast NVMe SSD and terabyte-scale persistent memory are filling the gap between the conventional DRAM-based memory tier and HDD-based storage tier. The most common practice today is that application developers must use different interfaces to explicitly determine which tier of memory/storage to store and retrieve data. For instance, on Summit-like systems, node-level NVMe, network-attached burst buffer, and global storage are all accessible to applications through file systems, but users need to manage their data stores across these tiers.

We advocate unifying storage and memory by creating storage interfaces tailored to memory-like access. A unified interface for in-memory representation and storage-level format of object store could eliminate the overhead of the abovementioned conversions. A unified interface could also improve programming productivity and performance portability across different memory/storage architectures. Such unified interfaces need to have a self-identifying format that augments global information in existing memory-pointer based interfaces because the data is likely to persist beyond the application's lifespan, and decentralizes the metadata management because terabyte-scale data volumes are commonplace on today's systems.

Opportunity:
A unified interface should support domain scientists to describe high-level data requirements and data views for processing and analysis. Based on the data requirements, system software could potentially eliminate redundant data conversions between in-memory formats and storage formats, or relax the management on data consistency and reliability, which often incurs high overhead due to serialization of metadata. Today, database constructs have provided a standard way of describing data views. However, since databases are heavy weight, new research is needed to develop lightweight interfaces that can meet the performance requirements of in-situ workflows. Potential tools and techniques include standard APIs such as JSON for domain scientists to describe high-level semantics and compiler-based analysis on the data analytics code to derive proper semantics. The APIs and new algorithms must be created through co-design between domain scientists and system software researchers to capture common data semantics that cover different science domains.

A translation layer must be developed to match the high-level data view and requirements defined in a unified interface to the underlying machine-level formats and granularity. Such a translation layer needs to derive efficient data organization and formats in the underlying storage device to match the access patterns to the data. For instance, a key-value store interface could have key and value data physically separated into different memory/storage locations. Frequently queried keys reside in fast storage and potentially use relaxed consistency constraints because they are mostly read, not updated. Instead, values could be compressed and stored in slower but larger storage devices, where the translation layer may even insert data staging processes to bridge the speed gap. Another example is to derive data distribution in storage based on the data views. For instance, temporal or spatial features processing would benefit from different data organizations in the storage to maximize bandwidth and reduce latency. In addition, the translation layer should also leverage hardware acceleration to speed up the conversion of data organizations. Such a translation layer needs to be easily extendable -- to cope with changes in the underlying memory/storage devices, new hardware capabilities and capture best practices developed in domain sciences.

Developing unified interfaces to access in-memory and in-storage data structures could open up several performance optimization opportunities that are infeasible in today's separate interfaces to memory and storage. First, in-transit data management during complex workflows and coupled multi-scale multi-physics simulations can be substantially accelerated if data storage can aggressively leverage underutilized memory resources and near-data processing capabilities. For instance, dynamically available resources, e.g., underutilized node memory and fabric-attached memory, can be considered a transient tier in the memory/storage hierarchy. However, it is infeasible to rely on users to leverage this storage tier because system resource utilization is unpredictable. Our previous work on leveraging underutilized memory on HPC systems for out-of-scale processing [2] proves the feasibility from system software's perspective. An open question is how applications could inform system software about their intent on data use to bypass unnecessary overheads in conversion, serialization, and consistency, etc.

Timeliness and Maturity:
Unified interfaces to access scientific data across memory and storage hierarchy are needed now because commercially available memory-like storage devices have arrived, including high-density memory modules like Intel's Optane DC PM, high-throughput SSD like NVMe, and the CXL standard. Memory, storage, and network switches with computing capabilities are emerging—for instance, Nvidia's ConnectX SmartNICs can offload data processing to the NIC, and El Capitan supercomputer features Rabbit-p processors attached to chassis-level SSD networks for near storage processing. Unfortunately, the diversity of new data storage and processing technologies further complicates scientific data management. Our previous works on user-space memory mapping service[1] show that providing a unified interface for applications to access data residing in different storage tiers improves the usage of new storage devices and application performance portability. A unified interface for in-memory and storage-level format with flexibility for domain scientists to express their high-level requirements on data could effectively reduce the conversion overhead on current systems and improve the utilization of new technologies on future machines.

References:
[1]Peng, I.B., Gokhale, M., Youssef, K., Iwabuchi, K. and Pearce, R., 2021. Enabling Scalable and Extensible Memory-mapped Datastores in Userspace. IEEE Transactions on Parallel and Distributed Systems.
[2]Peng, Ivy, Roger Pearce, and Maya Gokhale. "On the Memory Underutilization: Exploring Disaggregated Memory on HPC Systems." 2020 IEEE 32nd International Symposium on Computer Architecture and High Performance Computing.

LLNL-ABS-830016

Revisiting Database Technology for Scientific Data

Sandia National Laboratories, Jay Lofstead (gflofst@sandia.gov), Margaret Lawson (mlawso@sandia.gov)

Topic: The overlap between traditional storage systems and I/O (SSIO) efforts and data management.

Challenge: As parallelism increases, existing scientific IO library file formats are under strain dealing with the extreme scale. The two most popular styles, physically contiguous as used by HDF5 and PnetCDF and log-structured used by ADIOS, have hit the strain under different conditions. For the physically contiguous model, a 3D domain decomposition requires either a tremendous number of small IOs to construct the single, contiguous model in storage or a massive data rearrangement phase where processes exchange data to build larger blocks that can be written more efficiently. Either of these cases are problematic for disk-based storage arrays. The disk latencies accumulate quickly and dominate the IO time and/or the data rearrangement phase can account for 90%+ of the IO time large domains. For solid state storage, the data rearrangement time is unchanged, but the extreme number of IOPS for the small IO operations still add up to an inefficient operation.

The log-structured approach avoids both of these issues by building a buffer in memory and writing a series of log entries representing part of the simulation domain. While this avoids the data rearrangement and the small IO operations, it does not deal well with small data per process.

The metadata overhead for the contiguous model is fixed size no matter the data size. For the log-structured, the metadata is fixed per log entry. For strong scaling, the log-structured approach shifts from a low percentage overhead to a significant overhead. Further, the IO time for log-structured IO when the entire domain can be fit into memory of a single node, yet is split into multiple log entries, can be significantly worse than the physically contiguous model.

The challenge is not just in the write/read granularity, but also in the ability to select and find data within the structure. Both HDF5 and ADIOS walk the metadata tree to find the requested variable with the performance degrading as the metadata load increases. Further, while both support annotations to identify data of interest, there is not an efficient way to access that data either. In the case of HDF5, all metadata entries must be searched to find if an annotation is attached to it. For ADIOS, it can offer a list of the annotations, but it is a simple list that must be searched linearly with the same scalability issues mentioned above. Both the data size/granularity and the searching performance needs to be addressed in a way that can accelerate scientist activities.

Opportunity: Examining alternative data storage formats that may break through these barriers is needed to address the needs of extreme scale data management. For example, NoSQL databases, such as MongoDB, and key-vale stores/object stores like Redis, offer an alternative approach that has proven effective in non-traditional HPC applications. Initial attempts to use them in HPC environments [3] revealed incompatibilities with how these systems assume data management works. For example, columnar databases like Cassandra offer an unlimited number of optional columns with a few fixed columns. Unfortunately, the only columns that can have an index are the fixed columns. Any query that relies on the optional columns results in a table scan (or a spawn of Spark to perform the search in parallel).

While these results are not conclusive evidence these approaches cannot be used, they do offer challenges. Alternative examples exist, such as DAOS being something like a key-value store/object store and MadFS (Tsinghua University) or OceanStore (Huawei) that have adopted these tools to offer a new storage system design paradigm. In the case of DAOS, no independent evaluations of the software on a variety of different hardware has been offered making the advantages of the system neither attributable to software nor the Optane hardware used in the published studies. MadFS dominated the IO500 benchmark for ISC 2021 and OceanStore came in second for SC 2021 showing the potential of the alternative approach. However, neither of these systems have revealed publicly their specific design choices nor made their software available for others to try. MadFS offered confidential design notes and source code access to validate their performance numbers, but no public release was offered. OceanStore is a commercial product, but has had no publications nor any detailed design or implementation details offered to insights into how they work. These three efforts are promising, but are all focused on the storage system layer itself making the performance gains and stored data format non-portable. Instead, examining how to achieve these advantages at the user layer will offer a portable way to achieve the advantages without having to reformat the storage system and deal with things like how to archive a key-value store to tape.

Anecdotal testing of using embedded database technology demonstrated potential for a fast, low overhead

(space and time), reliable data format with an easy to program interface. Further detailed scalability demonstrations are shown in the EMPRESS [1] metadata management system. EMPRESS uses an embedded SQLite database to scalably store user annotations for any other data format by storing the logical domain coordinates rather than file offsets. These annotations are indexed using typical database indexes offering greatly accelerated data identification and direct logical access to the corresponding data without searching the underlying storage format. Extensive testing has shown this approach can offer as much as a 500x acceleration compared to using just using the underlying HDF5 format.

This work further inspired employing SQLite as the actual data storage format for a new IO library, Stitch-IO [2]. Traditional IO libraries largely require an entire simulation domain be written or at least the size of the entire domain be known at the start of the file use. For applications such as simulating 3D additive manufacturing, thermal spray modeling, and other cases where the domain is potentially very large, but the compute is focused in a small part of the domain for each timestep, using these traditional approaches made these simulations infeasible at scale. Stitch-IO eliminated the need to define the simulation domain size and simply stored blocks with coordinates and data stored in an associated blob. Using databases indexes on the coordinates and the timestep, selecting blocks for reading to construct the arbitrary request can automatically select the correct blocks and order them for easy combination into the request buffer. B-tree index performance and database query optimization statistics quickly cut down the potential matching blocks making the query far more efficient than any linear search of stored blocks. Further, by only storing the small area where compute occurred, the data storage requirements is reduced by as much as 99% with no fidelity loss. Even when running the simulation on 10s of processes rather than 1000s, the wall clock time is the same or less due to the radically reduced IO time.

Further advantages of this approach include SQLite's easy accessibility in Python and other environments. With the database engine controlling concurrency, while the simulation is running, an external monitoring process can perform periodic queries to determine the simulation progress. The full implications and potential of this approach has yet to be explored.

Timeliness or Maturity: While the advantages of using embedded databases technology as a data storage format are formidable, it is not without cost. SQLite, the most used database engine in existence, is designed for single user environments. Write operations lock the entire database serializing access. This does not scale to 100,000s of processes even with decomposing into multiple sub-files. Further, locking is currently implemented assuming POSIX locking works for the underlying storage system. GPFS properly implements POSIX locks while Lustre, the most common parallel storage system, does not. This requires user-level serialization.

Other efforts, such as DuckDB, have focused on having high functionality, but have ignored any performance concerns. However, like SQLite, DuckDB is offered as a source file that can be added to a project rather than deployed as a system-level service. With these newer database systems being developed, a strong foundation to address parallel writing access, for example, can be added without having to write an entire database engine. The reliability characteristics offered by a database engine and the performance advantages for indexed data access can complete change the face of data management for scientific codes.

References

[1] M. Lawson and J. F. Lofstead. Using a robust metadata management system to accelerate scientific discovery at extreme scales. In *3rd IEEE/ACM International Workshop on Parallel Data Storage & Data Intensive Scalable Computing Systems, PDSW-DISCS@SC 2018, Dallas, TX, USA, November 12, 2018*, pages 13–23. IEEE, 2018.

[2] J. F. Lofstead, J. Mitchell, and E. Chen. Stitch it up: Using progressive data storage to scale science. In *2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS), New Orleans, LA, USA, May 18-22, 2020*, pages 52–61. IEEE, 2020.

[3] J. F. Lofstead, A. Ryan, and M. Lawson. Adventures in nosql for metadata management. In M. Weiland, G. Juckeland, S. R. Alam, and H. Jagode, editors, *High Performance Computing - ISC High Performance 2019 International Workshops, Frankfurt, Germany, June 16-20, 2019, Revised Selected Papers*, volume 11887 of *Lecture Notes in Computer Science*, pages 227–239. Springer, 2019.

The Twilight of I/O as a User Concept

Jerome Soumagne[1], Gerd Heber[1], Andres Marquez[2], Elena Pourmal[1]

Topics:

- Interfaces for accessing data
- Understanding the overlap between traditional storage systems and I/O (SSIO) efforts and data management

Challenge

I/O middleware libraries (e.g., HDF5, ADIOS, PnetCDF, MPI I/O), commonly used on production systems for storing scientific data, have all been originally designed when parallel file systems and disk-based storage were prevalent. Dressed up in different flavors of higher-level models of abstraction (e.g., hierarchical, object-based model in HDF5), they have nevertheless perpetuated I/O as a user concept and concern. It is an unfinished job, because it creates an obvious dilemma: users have to match the application's data models to middleware primitives and worry about how these choices affect I/O performance. What is worse is that as these middlewares are ported to new storage types, the performance characteristics of middleware primitives may shift, creating the perfect nightmare for users [Xie2021]. This issue has only been exacerbated with the increasing amounts of data generated due to both an increase in problem complexity, in processing power, and new system architectures. Applications have therefore been forced to restructure and finely tune their I/O models so that they could get the most from the I/O bandwidth offered by parallel file systems [Wan2022]. This has been often at the detriment of slower reads, as post-processing algorithms generally no longer match with the data format that was output to disk, commonly forcing extra post-processing steps to be taken for data analysis to be realized efficiently and in a timely manner. As a consequence, the increasing need for more complex workflows (e.g., AI, multi-physics workflows) has drastically encouraged application users to find new solutions (e.g., in-situ, in-transit analysis) that bypass file systems in an effort to reduce the cost of analysis and I/O optimization efforts. To further reduce the cost of I/O, solutions such as asynchronous I/O are slowly taking an important role, but they require an important commitment from the application developer in order to be used properly, potentially making the use of I/O middleware even more complex. In that context, there is a challenge for I/O middleware libraries to provide software that no longer requires application's I/O to be finely tuned based on the storage system architecture but remains in tune with the application needs and data models.

Opportunity

With the emergence and preponderance of new technologies such as persistent memory and object stores, the limitations that used to be imposed by disk-based storage are now replaced with new opportunities for defining file formats and storing data—this has been seen recently with solutions such as Intel's DAOS [Soumagne2022], but also more broadly with Cloud solutions (e.g., Amazon S3, Ceph RADOS). While those solutions can be brought into existing I/O middleware solutions, application developers and users may only be able to fully grasp their benefits by rethinking once more the way they are doing I/O in order to reach the desired performance. In other words, those new capabilities, while beneficial, are not sufficient in themselves to directly fit to the needs of applications and the interfaces

[1] The HDF Group, [jsoumagne,epourmal,gheber]@hdfgroup.org
[2] Pacific Northwest National Laboratory, andres.marquez@pnnl.gov

exposed by I/O middleware. They must also evolve so that they can be fitted to the application data models as they were originally thought of. The evolution from disk and block-based storage to object stores now opens an opportunity for I/O middleware to rethink how interfaces should be presented to applications. The traditional write and read semantics may be re-thought to instead focus on the description of the I/O data model to prevent tedious and constant optimization efforts from the application developer. There is therefore an effort that must take place between application and middleware developers to design and evolve the current I/O interfaces.
Furthermore, along with those new technologies, new types of workflows and I/O needs have emerged, which are gradually being expressed through custom data services [Ross2020], designed to address a response to a particular I/O need (e.g., data staging, monitoring, multi-tiered storage abstraction, etc). Those data services, while becoming essential, also introduce another degree of complexity for application users, who may need to design custom I/O pipelines. In that sense, it also becomes the responsibility of I/O middleware to rethink the way data is accessed in order to facilitate that process and take advantage of those new types of I/O methods more efficiently. This may be done along with rethinking the type of semantics that are provided to applications so that they remain tied to data models. A more radical proposal would be to eliminate I/O as a user concept altogether. With persistent memory overcoming the chasm between random access and block I/O, there is now an opportunity for providing users with convenient primitives to state “persistence intents” and let a virtualized I/O layer be the new frontier for middleware developers (e.g., through I/O compilers, data services, etc).

Timeliness and Impact

The storage and HPC community has now reached a turning point where the traditional storage system is no longer the only means for accessing data. With the emergence of object stores and data services, there is a need to evolve I/O middleware interfaces that were designed at a time where monolithic block-based parallel file systems were the norm. The increased complexity of workflows and storage architectures require the rethinking of interfaces so that they no longer interfere with the applications’ data model and no longer require extraordinary engineering efforts of optimization.

Removing that burden from application developers has an extremely high impact and has the potential for saving precious development time by focusing more on science aspects and less on the engineering efforts. By removing complexity from the application, it has also the potential to open to new opportunities and types of workflows that were until now discouraged.

References

- [Xie2021] B. Xie *et al*., "Battle of the Defaults: Extracting Performance Characteristics of HDF5 under Production Load," *2021 IEEE/ACM 21st International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*, 2021, pp. 51-60, doi: 10.1109/CCGrid51090.2021.00015.
- [Soumagne2022] J. Soumagne *et al*., "Accelerating HDF5 I/O for Exascale Using DAOS," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 4, pp. 903-914, 1 April 2022, doi: 10.1109/TPDS.2021.3097884.
- [Wan2022] L. Wan *et al*., "Improving I/O Performance for Exascale Applications Through Online Data Layout Reorganization," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 4, pp. 878-890, 1 April 2022, doi: 10.1109/TPDS.2021.3100784.
- [Ross2020] R.B. Ross *et al.* Mochi: Composing Data Services for High-Performance Computing Environments. *J. Comput. Sci. Technol.* **35,** 121–144 (2020). https://doi.org/10.1007/s11390-020-9802-0

Leveraging In-network and In-Storage Computation for Complex Scientific Workflows

Joaquin Chung[1]*, Ganesh Sankaran[2], Nageswara S. V. Rao[3], and Raj Kettimuthu[1]
[1]Argonne National Laboratory, [2]USC Information Science Institute, [3]Oak Ridge National Laboratory
chungmiranda@anl.gov, gsankara@isi.edu, raons@ornl.gov, kettimut@mcs.anl.gov

Topic: *Optimizing data movement in complex workflows by using in-network and in-storage computation*

1 Challenge

Traditionally, complex scientific workflows produce data in advanced scientific instruments and perform analysis in one or more remote high-performance computing (HPC) facilities. Current practice, as shown in Figure 1, is often to stage data as files from experiment to HPC via an intermediate file system: typically, one accessible from dedicated data transfer nodes (DTNs) located in a Science DMZ at the perimeter of the HPC facility's campus network. The incoming data from the wide area network (WAN) are received by services running on the DTNs, which write the data to the file system; analysis codes running on HPC compute nodes then read the data from the file system. Most often, this data sit idle on the file system for a significant time as the compute jobs almost always have a queue wait time on the HPC systems. In case if the compute nodes are readily available, intermediate disk-based file system for data staging can introduce significant performance degradation, even when using a high-performance parallel file system [1]. The challenge here is to address these inefficiencies for complex scientific workflows.

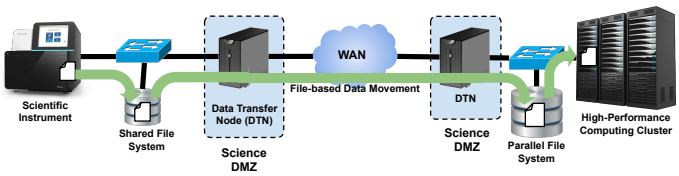


Figure 1: State-of-the-art complex scientific workflow using file staging via intermediate file system

2 Opportunity

Emerging in-network and in-storage compute capabilities offer promising opportunities. Advanced networks can also provide storage capabilities in addition to compute capabilities. For example, the recent NSF funded FABRIC test bed offers both compute and storage capabilities in the network. These new capabilities open new opportunities to overcome challenges related to available bandwidth and resource co-scheduling in remote data analysis use cases for scientific computing. We envision new infrastructures capable of "parking" data in the network while waiting for HPC resources to become available (see Figure 2). Furthermore, while data is waiting for HPC resources, we could perform computation in the network devices (or storage) by keeping data in movement across the infrastructure.

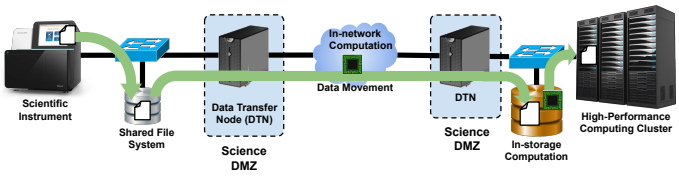


Figure 2: Complex scientific workflow using in-network and in-storage computing

Opportunities in Single-job Scenarios: While traditionally, data have been moved closer to computation, pervasive computing and storage capabilities in-network [2] and in-storage [3] present an opportunity to move computation closer to the data. Usually, computation code is much smaller in size compared to data and this presents an interesting optimization opportunity. Offloading precomputation to network and storage devices will improve the chances of meeting the overall objective of optimizing the complex workflow. For instance, when the lead time to start computation on an HPC is high, compute resources available

*Corresponding author

within the network and storage units could be orchestrated to offload the precomputation or a part of the computation.

With security becoming a growing concern, data getting processed as it moves across a network presents an interesting moving target type of defence that is hard to hack as compared to a longer duration mapping between computation and data. However, this approach data moving across the network should be optimized as more traffic in the network could produce unnecessary congestion. Finally, we should start reasoning about how complex scientific workflow can benefit from in-network and in-storage computation while maintaining a balance between computation efficiency and infrastructure usage.

Opportunities in Multiple-job Scenarios: A short term goal is to minimize duplicate data movement along a network path. A straight forward approach is to use multicast or unicast replication to optimize data movement of the same dataset to multiple jobs in a complex workflow. Another example is to consider jobs starting at different times that can stage data in-network to minimize latency and maximize reusability, or match multiple jobs' data consumption rates using many heterogeneous storage resources (e.g., RAMdisk, SSD, and HDD-based SAN).

We envision progressive computation of data along a network path to cater to different jobs requiring different functions of same data (e.g., one consuming raw data, another consuming normalized data, and another consuming log-normalized data) as a mid- to long-term goal. This could enable the realization of security and policy primitives with in-storage and in-network computation across multiple jobs, placement of desired computation at vantage points to cater to multiple consumers minimizing data stream replication with heterogeneous in-network computation resources (e.g., P4 [4] switch or VNF), and bifurcation (trifurcation and more m-furcation) of one aggregate data stream to cater to many jobs by placing filtering at vantage points: schema based ($field_1$ to job_1 and $field_2$ to job_2) and semantic filtering ($field_1 = value_1$ for job_1, $field_1 = value_2$ for job_2) or a combination of both.

3 Timeliness

The emergence of programmable switches has inspired new ideas in in-network computing. For instance, early proposals have explored implementing network functions such as load balancers fully on the data plane. In-network cache applications that leverage programmable switch hardware to implement key-value stores are among early demonstrations as well. Similar approaches have taken advantage of solid-state drives (SSD) to perform in-storage computing.

Modern programmable switches have different processing capabilities depending on their hardware and software architectures. In general, their main benefits are high throughput (10 billion packets/s) and low latency (sub-microseconds). Similarly, in-storage processors leverage large memory bandwidth (1Gbps per processor) and their proximity to storage drastically reduces data movement. However, programmable switches support only basic arithmetic/Boolean operations and they do not allow loops. Furthermore, both programmable switches and in-storage processors lack a common interface for optimal orchestration. Although these capabilities may be sufficient to implement network functions or key-value stores in the network, scientific applications require more complex operations.

By efficiently integrating in-network and in-storage computation into complex scientific workflows, we intend to improve the ratio of data used by the workflow to the amount of data transferred by leveraging heterogeneous compute and storage resources without impacting performance and security. To achieve this goal, we have identified the following research directions: placement of security primitives, function transformation and staging at vantage points to provide relevant information depending on the application access pattern looks like a good research direction, development of common interfaces for orchestrating in-network and in-storage computing resources, and co-design of complex/heterogeneous scientific environments.

References

[1] A. Gainaru, G. Aupy, A. Benoit, F. Cappello, Y. Robert, and M. Snir, "Scheduling the I/O of HPC applications under congestion," in *2015 IEEE International Parallel and Distributed Processing Symposium*, pp. 1013–1022, May 2015.

[2] A. Sapio *et al.*, "In-network computation is a dumb idea whose time has come," in *Proceedings of the 16th ACM Workshop on Hot Topics in Networks*, pp. 150—-156, 2017.

[3] I. Jo, D.-H. Bae, A. S. Yoon, J.-U. Kang, S. Cho, D. D. G. Lee, and J. Jeong, "Yoursql: A high-performance database system leveraging in-storage computing," *Proc. VLDB Endow.*, vol. 9, p. 924–935, aug 2016.

[4] P. Bosshart *et al.*, "P4: Programming protocol-independent packet processors," *SIGCOMM Comput. Commun. Rev.*, vol. 44, pp. 87–95, July 2014.

Automating Data Management Through Unified Runtime Systems

John Wu, Bin Dong, Alex Sim, Lawrence Berkeley National Laboratory, KWu@lbl.gov

Topic: Providing data management support for AI and complex workflows; Interfaces for accessing data

Challenges: Scientific discoveries, such as Higgs boson and gravitational wave, require extensive data analyses. To accomplish these analysis tasks, domain scientists often build extensive tools around popular data analysis tools and artificial intelligence (AI) tools. For example, the discovery of Higgs boson utilizes a large framework (i.e., ROOT) built by the high-energy physics community [1], while the gravitational wave discoveries make use a collection of data management and AI tools tied together with custom glue layers [2, 3]. These custom tools require considerable efforts, maybe thousands of person-years, to develop and maintain for each application domain. Are there ways to reduce such custom software development efforts? Currently, the most successful large-scale data analysis and AI frameworks are designed for commercial applications, originally motivated by large internet business needs. Their core data management functions are generally known as Big Data systems. The core design principle of these systems is to separate the concerns of data management systems from application logic, which dramatically simplified the development of large applications that require the power of many CPUs to work together. These Big Data systems created a virtuous cycle that attracted millions of data scientists to build more useful tools for more applications, including scientific applications. There are a number of high-profile reports on using the current machine learning tools for science [3-5]. However, we also see fundamental differences between internet business use cases and scientific use cases, and believe it is worthwhile to investigate options for developing more effective data management frameworks for science.

To illustrate the differences and their impact, we briefly describe two use cases involving astronomy. In a study of dark matter distribution, a deep neural network named CosmoFlow was used to predict the evolution of the dark matter in the universe. In this case, because each data record that needs to be fed into the neural network is much larger than that of a typical commercial application, the training process spent considerably more time in the I/O operations than in commercial applications. In order to reduce the I/O time, scientists have been exploring different approaches, such as to use large amount of solid state storage to hold the data. Unfortunately, such approaches prevent the data records to be shuffled as the training process generally calls for, which created doubt about the trustworthiness of the solutions found. In another example involving gravitational wave from a neutron star merge, a considerable effort is spent to search for relevant observations distributed in databases around the world to augment the main gravitational observation. The existing AI tools generally assume all necessary data records are available on the same platform, while this use case requires dynamic integration of data records from widely distributed sources. These examples bring up a number of common data management tasks that are currently handled through custom software [3]. Following the lesson from Big Data successes, we believe the best way to address these data management challenges is to develop a unified runtime system of data management services to connect scientific data with the popular AI tools.

Opportunities: Scientists have a vast amount of data to analyze and the internet businesses have developed a myriad of analysis tools, we see a great opportunity to bring them together. This has recently become an effective approach because the foundational **data model** used in

science and in popular machine learning tools are **converging**. In science much of the data is stored in multi-dimensional arrays, while many of the new generation of deep learning frameworks are similarly using arrays (tensors) as their core data structure. Earlier Big Data tools have used key-value pairs as their data model, but experiences showed that common analyses could be much more effectively expressed using arrays than using key-value pairs. Now that scientific applications and popular data tools are using a compatible data model, we see a great potential to develop a unified data management system to connect different data components, such as object stores, I/O libraries, and machine learning tools. Due to the dynamic nature of data accesses from the examples above, we foresee the unified data management system to include a number of different run-time services to automate common data management operations. Next, we briefly outline a few example services.

Advanced Data Searching: In a large data set, usually a relatively small subset of the records contains the most critical information for a particular task. In the past, the indexing techniques to accelerate the searches only involve known features in the data. It would be useful to bring together the state of art on statistical feature extraction, hierarchical indexing, and I/O performance optimization to support advanced searching operations on HPC systems.

Efficient Data Access for AI: The upcoming solid state storage systems and memory systems offer brand new opportunities to accelerate the I/O operations for massive scientific datasets. Work is needed to build effective I/O tools and services to take full advantage of these new storage devices.

Optimized Data Communication: Most of the existing ML tools require the data to be randomly shuffled periodically, which creates random data accesses or massive data communication. Compared with computation, such as additions and multiplications, reading a byte from disk and sending a byte another node take considerably longer time. It is therefore necessary to optimize HPC data communication for common types of data analysis operations.

Timeliness: In the data management context, the lesson we take from successes of Big Data systems is to separate the data management tasks from the data analysis operations (separation of concerns). In the recent years, we have seen a convergence of data models that could make this separation feasible for scientific application. It is possible to develop an efficient set of data services to mediate between scientific data and popular tools. These services would allow the scientists to concentrate on their analysis operations while leaving the data management tasks to the data run-time system.

References:

1. Brun, R. and F. Rademakers, *ROOT — An object oriented data analysis framework.* Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, 1997. **389**(1-2): p. 81-86.
2. Talbot, C., et al., *Parallelized inference for gravitational-wave astronomy.* Physical Review D, 2019. **100**(4): p. 043030.
3. Cabero, M., A. Mahabal, and J. McIver, *GWSkyNet: A Real-time Classifier for Public Gravitational-wave Candidates.* The Astrophysical Journal, 2020. **904**(1): p. L9.
4. Jordan, M.I. and T.M. Mitchell, *Machine learning: Trends, perspectives, and prospects.* Science, 2015. **349**(6245): p. 255-260.
5. Piccione, A., et al., *Physics-guided machine learning approaches to predict the ideal stability properties of fusion plasmas.* Nuclear Fusion, 2020. **60**(4): p. 046033.

Support for In-Flight Data Analyses in Scientific Workflows

John Wu, Lawrence Berkeley National Laboratory, KWu@lbl.gov
Ben Brown, Paolo Calafiura, Quincey Koziol, Dongeun Lee, Alex Sim, Devesh Tiwari

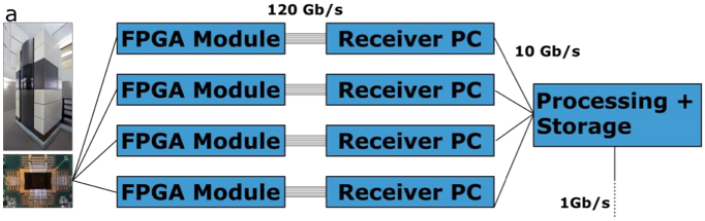
Topic: Providing data management support for AI and complex workflows

Challenges: DOE user facilities produce large volumes of data that led to grand scientific discoveries [1-3]. However, the raw data from these facilities are very large, often noisy, and typically require extensive data analyses to produce scientific results [4, 5]. We observe that each application domain has their own data analysis tools. For example, High-Energy and Nuclear Physics (HENP) experiments have developed extensive online data reduction tools [4, 5]. Similarly, other experimental and observational facilities each has their own software tools [6-8]. Since each tool took considerable efforts to develop, often requiring many thousands of person-years, a natural question is: Would a unified framework be able to serve many scientific domains?

Our basic thesis is that a common data management system could support many different data analysis algorithms. Next, we use compression as an example of data analyses because it is a common operation and actively used in many science workflows. Furthermore, we concentrate on a scenario where data analysis operations could be performed while it is *en route* from data acquisition site to the storage site.

To be concrete, next we briefly describe a use case from operations of an electron microscope. The National Center for Electron Microscopy (NCEM) is a DOE user facility located at Berkeley. One of the latest addition to NCEM is a four-dimensional scanning tunneling electron microscope (4D-STEM) that scans each sample in a 2D grid and collects a 2D electron diffraction pattern at each grid point [9]. This camera provides a new imaging capability previously thought to be only theoretically possible, and opens up an entirely new set of techniques for measuring strain, magnetism, phase, grain orientation, at the nano- to atomic-scale [10].

The 4D camera generates data at about 500 Gbit/s (4X120 Gbit/s) and requires online data reduction as illustrated in the figure on the right [9, 11]. Compression will allow users to produce more data and larger scans that produce "more science." Further improving the data reduction process would reduce the storage requirement and network bandwidth requirement, and could lead to online diagnoses of the image scanning process, which improves the overall operation of the 4D camera. Currently, the reduction process uses a combination of FPGA and custom PCs. Replacing this custom setup with off-the-shelf edge computing devices and using data centers for permanent storage and analyses could reduce cost of operating the microscope. In this process, we would need an in-flight data reduction framework to connect data producers, such as this 4D-STEM with DOE computing facilities.



Opportunities: Storage and data communication technologies are going through a period of rapid changes recently, which creates a great opportunity to introduce in-flight data management capability to support the data reduction workflow mentioned above. These changes include data communication networks are becoming softwarized (through software-defined networking), data storage are being active, and edge computing is bringing more devices outside of computer centers.

Together, these technologies could be used to effectively support complex data analysis workflows with flexible in-flight data analyses.

In addition to the changing hardware technologies, a key roadblock in software technology that was preventing different applications from sharing a common set of data management tools is that they each uses a different file format to store their data. Fortunately, many of these experimental and observational researches are starting to adopt new file standards based on HDF5, which suggest that HDF5 could be a common platform for a unified set of data management tools.

Timeliness: HDF5 has been a key technology in managing data, performing I/O, and providing a portable file format for science data in various fields. It is one of the most used I/O libraries at DOE supercomputing facilities. A number of recent features introduced into HDF5 library are very useful for online data reduction tools [12]. For example, HDF5 supports filters to perform compression and decompression transparently and asynchronous execution user-defined operation that could include data reduction and analysis. Even though additional features might still be needed, these recently added features form a strong foundation for a unified data run-time.

References:

1. Abbott, B.P., et al., *Observation of Gravitational Waves from a Binary Black Hole Merger.* Physical Review Letters, 2016. **116**(6): p. 061102.
2. G. Aad et al., *Combined Measurement of the Higgs Boson Mass in pp Collisions at √s=7 and 8 TeV with the ATLAS and CMS Experiments.* Phys. Rev. Lett. **114**, 191803
3. Adams, J., et al., *Experimental and theoretical challenges in the search for the quark–gluon plasma: The STAR Collaboration's critical assessment of the evidence from RHIC collisions.* Nuclear Physics A, 2005. **757**(1): p. 102-183.
4. Albrecht, J., et al., *A Roadmap for HEP Software and Computing R&D for the 2020s.* Computing and Software for Big Science, 2019. **3**(1): p. 7.
5. Espinal, X., et al., *The Quest to solve the HL-LHC data access puzzle.* EPJ Web Conf., 2020. **245**: p. 04027.
6. Datta, A., et al., *A data reduction and compression description for high throughput time-resolved electron microscopy.* Nature Communications, 2021. **12**(1): p. 664.
7. Girod, L., et al., *A self-calibrating distributed acoustic sensing platform*, in *Proceedings of the 4th international conference on Embedded networked sensor systems*. 2006, Association for Computing Machinery: Boulder, Colorado, USA. p. 335–336.
8. Hammersley, A., *FIT2D: a multi-purpose data reduction, analysis and visualization program.* Journal of Applied Crystallography, 2016. **49**(2): p. 646-652.
9. Ophus, C., *Four-Dimensional Scanning Transmission Electron Microscopy (4D-STEM): From Scanning Nanodiffraction to Ptychography and Beyond.* Microscopy and Microanalysis, 2019. **25**(3): p. 563-582.
10. Kirkland, E.J., *Advanced computing in electron microscopy*. 2020: Springer Nature.
11. Ercius, P., et al., *The 4D Camera – An 87 kHz Frame-rate Detector for Counted 4D-STEM Experiments.* Microscopy and Microanalysis, 2020. **26**(S2): p. 1896-1897.
12. Warren, R., et al. *Analysis in the Data Path of an Object-Centric Data Management System*. in *2019 IEEE 26th International Conference on High Performance Computing, Data, and Analytics (HiPC)*. 2019.

A self-validated data model to enable integrative, reproducible analysis

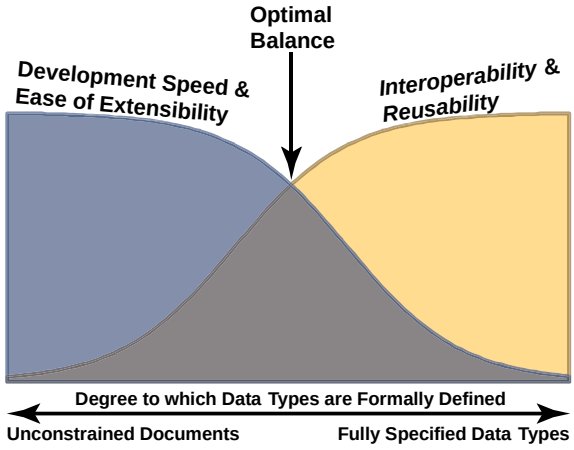
John-Marc Chandonia (JMChandonia@lbl.gov) and Adam P. Arkin (APArkin@lbl.gov), Environmental Genomics and Systems Biology Division, Lawrence Berkeley National Laboratory

Topic: FAIR data modeling

Challenge: Many organizations face challenges in managing and analyzing data, especially when relevant datasets arise from multiple sources and methods. Analyzing heterogeneous datasets and additional derived data requires rigorous tracking of their interrelationships and provenance. This task has long been a Grand Challenge of data science, and has more recently been formalized in the FAIR principles[1]: that all data objects be *Findable*, *Accessible*, *Interoperable* and *Reusable*, both for machines and for people. Adherence to these principles is necessary for proper stewardship of information, for testing regulatory compliance, for measuring the efficiency of processes, and for facilitating reuse of data analytical frameworks.

The problems of making an organization's data *Findable* and *Accessible* to its members are largely solved by modern databases, including relational databases such as SQL[2] and non-relational "NoSQL" databases such as document stores[3]. The challenges of assigning a unique, permanent ID to each dataset generated within an organization, and then ensuring that the dataset is deposited into a database where it may be retrieved by appropriate people, are largely managerial problems rather than technical ones. On the other hand, making all types of data both *Interoperable* (enabling powerful integrated analyses that span many datasets generated by different teams within an organization) and *Reusable* (enabling later use of datasets by somebody other than the person or team that originally generated the data) is extremely difficult. *Reusability* is challenging because non-specialized data storage formats often do not allow or require specification of key details, even basic ones such as units of measurement. As a result, undocumented assumptions and conventions can make it very difficult to reproduce or reuse data. Ensuring *Interoperability* between datasets is challenging for many of the same reasons: when different groups within an organization produce data, impedance matching must be done in order to perform an integrative analysis. Some sources of impedance are differing units, incompatible scaling or normalization of different datasets, and different identifiers used by different groups to refer to the same objects.

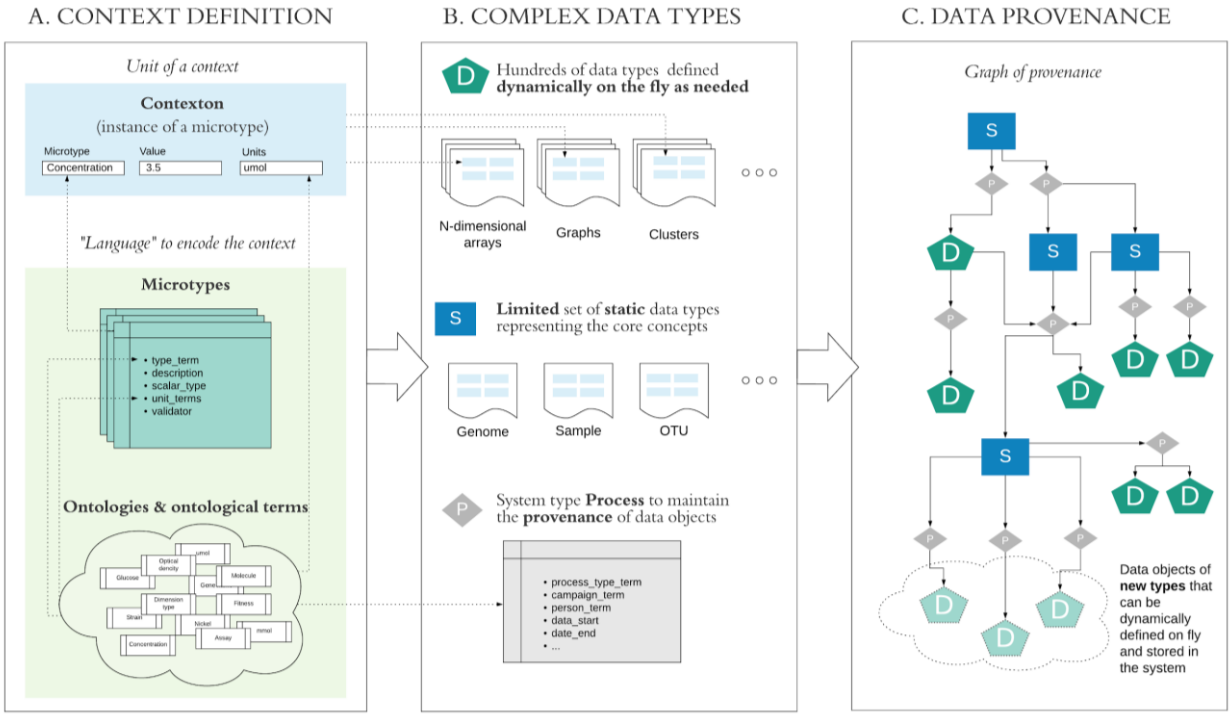
FAIR data has been achieved for some specialized data types, such as protein structures, evolutionary relationships, and genome sequences (as well as the data types in KBase[4]). However, modeling new data types in a FAIR way involves hand-crafting complex data structures (e.g., SQL tables) to capture all relevant details of these data; these models are expensive to build and maintain. Most bench scientists prefer to store their data in Excel spreadsheets or other general-purpose formats such as document stores, even though these are difficult to make FAIR. These tradeoffs are illustrated in the figure to the right.



In practice, most data types can be represented using a small number of data models, such as arrays, graphs, trees, and hash tables. We surveyed hundreds of data types used by our colleagues in the ENIGMA project, a large consortium of researchers that study how communities of microbes interact with their environment (https://enigma.lbl.gov/). We discovered that the vast majority of data, from raw assays to processed results, can be represented as multi-dimensional arrays of scalars. We believe that this result is generalizable across many fields of research and business. For example, climate modelers widely use the Xarray library for storing data in multi-dimensional arrays, in which key-value pairs are used to label each dimension[5]. Similar libraries exist in most computer languages, and file formats such as HDF5 and NetCDF-4 are well supported, mature technologies. However, a common file format alone is not sufficient to ensure *Interoperability* and *Reusability*: in addition to a standard file format, all data, dimensions, and units in these multidimensional arrays must also be formally and rigorously documented.

Opportunity: We have developed the Contextual Ontology-based Repository Analysis Library (CORAL), a novel framework for data modeling and analysis, which aims to achieve an optimal balance between the ease of adding new data types and adherence to FAIR principles.

An overview of the CORAL data model is shown in the figure below. A) to rigorously document context for all data, we introduce the concept of a ***contexton***, or unit of context. Contextons are built using ***microtypes***, which we define as atomic data types representing a simple concept relevant to a domain of interest. Both rely on ***ontologies***, which define a controlled vocabulary for describing a domain of interest. Together, the microtypes and ontologies defined for a particular instance of our platform represent a language that allows users to formally describe all data in that instance in a way that is both *Interoperable* and *Reusable*. B) Dynamic data types, which make up the vast majority of data, are defined by the users of the system as they are needed. Dynamic data types are built by combining commonly used mathematical data structures with contextons. This "building blocks" approach enables new data types to be defined as needed, with low costs, but also ensures that they are documented in the formal and rigorous manner that is necessary for *Interoperability* and *Reusability* of the data. A limited number of static core types, which are fully specified traditional data structures, are also built using contextons in order to ensure *Interoperability* with the dynamic data. These static core types include the system type *Process*, which is a special core type needed to document the provenance of each data object. C) All static and dynamic data are referenced in an object graph, where nodes are static or dynamic datasets, and edges are processes. This graph formally annotates the provenance of all data.



Timeliness: CORAL relies on new multi-model database technology (ArangoDB) to enable simultaneous searches of both the provenance graph and the linked data objects.

1. Wilkinson, M. D. *et al.* The FAIR Guiding Principles for scientific data management and stewardship. *Sci. Data* **3**, 160018 (2016).
2. Codd, E. F. Relational database: a practical foundation for productivity. *Commun. ACM* **25**, 109–117 (1982).
3. Pokorny, J. NoSQL databases: a step to database scalability in web environment. in *Proceedings of the 13th International Conference on Information Integration and Web-based Applications and Services* 278–283 (Association for Computing Machinery, 2011). doi:10.1145/2095536.2095583.
4. Arkin, A. P. *et al.* KBase: The United States Department of Energy Systems Biology Knowledgebase. *Nat. Biotechnol.* **36**, 566–569 (2018).
5. Hoyer, S. & Hamman, J. xarray: N-D labeled Arrays and Datasets in Python. *J. Open Res. Softw.* **5**, 10 (2017).

Data Fabric and Data as a ”First Class Citizen”

Jon Fortney[1], Marshall McDonnell[2], Dylan Johnson[3], and Stuart Chalk[4]

[1]fortneyjm@ornl.gov, Computer Science and Mathematics, Oak Ridge National Laboratory
[2]mcdonnellmt@ornl.gov, Computer Science and Mathematics, Oak Ridge National Laboratory
[3]n01448636@unf.edu, Department of Chemistry, University of North Florida
[4]schalk@unf.edu, Department of Chemistry, University of North Florida

TOPIC: Data-management support for AI and complex workflows

I. Challenge

Science is currently in the age of a digital, data-driven science revolution. With this, data has an increasingly significant role and value to facilitate data-directed decision making. As researchers, trying to perform interdisciplinary science, we increasingly need our data to be more connected, regardless of its source. Yet, a challenge that currently exists is the ”data silos” for the ever-growing and heterogeneous nature of scientific data across domains.

A data silo means the data is not as accessible as it should be or maybe not at all for teams besides the ones that generate it. If a great deal of time is required just decoding data to make it translatable to another team, one or more data silos are likely present in the organization. Data silos stem from issues that are structural (i.e. many layers of separation between teams), cultural (i.e. keeping data from each other, rather than working together), and technological (i.e. applications most likely not designed to be integrated together).

The Findable, Accessible, Interoperable, and Reusable (FAIR) data principles aim to help address data silos. Specifically, they describe as an end goal that, for our scientific data, we need *”...more rigorous management and stewardship of these valuable digital resources, to the benefit of the entire academic community.”* [1] The FAIR data movement requires additional metadata be recorded. This provides a challenge since there is a burden of accurate metadata for FAIR data. This requires rethinking how we collect scientific data from ”birth” (i.e. when the data is produced at the experimental instrument or computational resource)

II. Opportunity

A. What is a Data Fabric?

A data fabric, overall, is a set of data management principles, guiding practices, communities and standards that can *”...optimize access to an organization’s distributed data and intelligently curate and orchestrate it for self-service delivery.”* [2] Given that an organization’s distributed data will evolve with time, both in content, scale, and format, it is important to have a flexible and scalable approach. These concepts of evolution and scalability are central to a data fabric. The data fabric you need is highly correlated to questions like: How can data be more valuable to you? What obstacles and roadblocks do you currently have with data? What are the software and services that provide data or allow you to perform operations on data?

1) Principles, best practices, communities, and standards: For an organization to foster this evolving nature of the data fabric, it requires a set of data management principles, guiding practices, communities and standards and very importantly a philosophical view that data and the services that enable it to have more value are really important. The software community and the landscape around it changed drastically during the big data revolution in an effort to handle and make the increasing amount of disparate data useful to more people. Not all the changes were positive but a common set of principles, approaches and best practices evolved and became more useful. A data fabric should always strive to be cooperative with communities and working groups because of the inherent value in connected data and services. Part of that cooperation can and does result in formally published standards and approaches.

2) Design principles: The design principles will be core to the scientific data fabric.

a) Data availability: Data should be intuitive. You should know how to get it. You should know how long that takes (aka Service Level Agreements/SLAs) and what the steps are to get it. You should have services that provide it to consumers in the format they need using a technology that makes sense for the problem. It should be easily discovered. It should be searchable the way you need to search: Textual, spatial, temporal, graph, ontological...

b) Data value: Data has value intrinsically. It is hard, arguably impossible, to understand the full value of any specific piece of data a priori. That is, before it is inspected and analyzed: Is it like any of the data you already have? If you talk about it the same way you talk about your existing data, can you learn more about your data and ask it harder questions? Treat all of your data like it might have a secret, that when known, is very valuable.

c) Connected Data: Data is inherently more valuable when it is connected. When connected by something like an ontology, it earns semantic value that may be applicable to the way you design and build the data fabric. Knowledge graphs [3] are a good example of this idea and they are illustrative of the kinds of problems connected data and data fabric solve.

d) Data is FAIR "from birth": As the knowledge graph is created for the connected data, this should not have to always be a manual process after the data is originally created. Instead, the data should be created according to FAIR data principles "from birth". The more relevant semantic meaning the data has, the richer it will make the data fabric. Also, this will streamline the data ingestion process as well.

e) Harmony: Data, Services and Software: We all use similar data. Even if it not explicitly connected, yet. There are large and very important movements going on currently that emphasize the value in talking about data similarly through ontology and various other semantically expressive approaches both technically and conceptually. We should acknowledge the obvious value in these efforts and enable that way of thinking in the services and software we build.

f) Learn from your data: If captured and connected properly you can learn a lot about your data that will provide value to your overall effort and mission. Embrace that and constantly be looking for ways to derive value from it. Specifically, explore using AI to try to learn patterns that might be useful at a broader level, use statistical analysis to determine the most efficient ways to solve problems, and utilize predictive workflows based on current knowledge and many more.

B. What does a Data Fabric look like for a Scientific Organization?

The National Science Data Fabric [4] is a new project funded by the National Science Foundation that shows potential to create the first international scientific data fabric. We speculate this will be a pioneering project that will lay the foundation for additional large-scale data fabric developments for scientific data. This is a unique opportunity for the Advanced Scientific Computing Research (ASCR) program to also invest in empowering data-driven science by promoting data to a "first class citizen" across domains via similar development into integration of a data fabric.

III. Timeliness

A. Short-term strategy and gains

The data fabric can be adopted easily and unobtrusively. In fact, the approach values flexibility, ease of adoption and embraces the requirement to provide value early and consistently while iteratively growing into what it needs to be. It should start by identifying, including and participating in the communities and groups already working on these concepts, even outside of ASCR. As a group we should be understanding, refining and adopting methodologies and practices to unify our data and services to make them more useful. This purposeful approach offers the chance to innovate the way we do science and write software potentially solving harder problems than we have ever solved.

a) Best case for on-boarding data: Everyone talks about data the same way and the data unification effort is very minimal. That is to say, we use a connected ontology and the semantics we express are in a similar domain. We can leverage existing raw data and we only use, much lighter, metadata such that our hardware, network, storage needs etc. are minimal and they are already available and paid for. The amount of effort to participate in and leverage the power of the data fabric must be kept minimal. Ultimately, semantic meaning can be derived and directly leveraged to enhance the data or guide the types of services and software required.

b) Worst case for on-boarding data: There is no understanding or value given to connected data and services. It is challenging and takes more time to talk about data accurately and universally for organizational or other reasons. Existing services and data are hard to access and can not provide programmatic access in a meaningful way. Connected data technologies can take longer to develop at scale. We have to automatically try to derive semantic meaning rather than have participants do that explicitly, which is almost always more precise.

B. Long-term strategy and gains

One important long-term strategy of the data fabric is it must be allowed to evolve with research and scientific applications. A rigid data fabric that is inflexible for new data ingestion will quickly become just another isolated, data silo for the organization. Only by continuing to invest in and develop the connection of the organization's data, governed by data fabric design principles, will the organization continue to reap the benefits of the new data-driven science revolution.

References

[1] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne *et al.*, "The fair guiding principles for scientific data management and stewardship," *Scientific data*, vol. 3, 2016. [Online]. Available: https://doi.org/10.1038/sdata.2016.18

[2] IBM, "Data fabric," 2021. [Online]. Available: https://www.ibm.com/analytics/data-fabric

[3] V. K. Chaudhri, "Stanford university lecture for cs520: What is a knowledge graph?" 2020. [Online]. Available: "https://web.stanford.edu/class/cs520/2020/notes/What_is_a_Knowledge_Graph.html"

[4] F. W. A. S. J. A. G. S. A. G. I. R. J. G. C. K. G. T. N. Z. P. F. O. G. D. B.-T. D. D. D. C. B. S. K. B. M. P. S. S. I.-h. C. J. L. Valerio Pascucci, Michela Taufer, "National science data fabric website," 2021. [Online]. Available: http://nationalsciencedatafabric.org/

Scientific Reproducibility and Management of Data

Joshua S. Brown (brownjs@ornl.gov), Rafael Ferreira da Silva (silvarf@ornl.gov), Alex May (mayab@ornl.gov), Olga A. Kuchar (kucharoa@ornl.gov)
Oak Ridge National Laboratory, Oak Ridge, TN

Topics:

- Capturing provenance information about scientific data
- Providing data management support for AI and complex workflows

Challenge:

In recent years, there has been rapid production of data artifacts – both large scale (traditional Big Data) and large ensembles of small-scale data sets (typical machine learning applications). A growing technical burden has been placed on researchers that attempt to organize and manipulate data. Often scripts must be used to do simple tasks. Simply copying data from a remote computer to a local machine can take days depending on the network bandwidth and size of the data source.

Furthermore, tracking the many parameters that are involved in running simulations or computational models is often nontrivial. This poses a problem in keeping data generation reproducible and testable. Using word documents, README files, or lab notebooks to track changes made while configuring a computing environment to set up and run a model requires a level of detail that is not easy to maintain, even if extreme care is taken. As science typically follows an iterative process this reproducibility challenge makes it difficult to build and expand on existing scientific discoveries. Appropriately capturing the provenance of a dataset, the metadata around the configuration of models, and analysis tools used in its evolution in a complicated workflow is essential for adding transparency to the scientific methodology and a data set's reproducibility.

Reproducibility and manipulation of a data set are also complicated by outdated methods of dissemination. Journal articles remain the de facto means of disseminating scientific results. In 2015, a study looking at the reproducibility of results attained in several economics studies noted that in nearly half the papers it was impossible to reproduce the results [1] often due to missing files. The complexity of computing environments in which a data set is generated or analyzed can also make it difficult to manually record all the relevant parameters correctly. Two scientists trying to repeat a scientific workflow are likely to see differences in their results, due to computing environments or missed parameters. It can take multiple iterations of adjusting parameters to reproduce the results exactly.

Opportunity:

Publication of a dataset to a repository along with a journal article is not enough to ensure reproducibility. The complexity of modern scientific problems means they cannot be accurately captured in traditional pen and paper approaches. Computational sciences that generate large amounts of data from scratch using first principles are uniquely poised to take advantage of pipelining tools that can immortalize the generation of a data set with a corresponding small digital footprint. Ideally, it would be possible to then download such a pipeline and reproduce the results at a press of a button. If science is truly meant to be a collaborative endeavor it should be possible for others to adjust and build upon such a digital pipeline without having to repeat processes in the pipeline that are required for reproducibility but offer little scientific value. The challenges of using digital pipelines to reproduce experimental results are larger but there remains opportunity for improvement given that almost all scientific instrumentation has a digital component with configuration options and software that could be captured, though there exist hurdles in overcoming restrictions related to proprietary vendor instrumentation.

Tools exist for many aspects of managing data, but they only solve parts of the problem. The combination of these tools to generate truly reproducible digital pipelines still faces problems in the form of:

- proprietary software used to generate the data;

- the reliability and long-term stability of data sets or tools hosted on the cloud or accessed over the internet;
- the effective capture of metadata and configuration options around experiments and simulations; and
- the publication of data pipelines that can be run along with journal articles.

Timeliness:

Reproducibility of data has benefitted from active development in the field of Machine Learning models on workflow tools such as SnakeMake [2], Kubeflow [3], and many others. These tools help to automate complex data workflows. Container technologies such as Singularity [4] have also gone a long way in helping to make scientific workflows reproducible and portable. Dependency management tools such as Spack [5] have attained success in the reproducibility of building complex software stacks to run simulations.

Upcoming Web3.0 technologies offer a means of indexing, discovering, and monitoring provenance of scholarly articles, data resources, software, data standards, and conglomerate configurations. These may be indexed and officiated using distributed autonomous organizations (DAOs) such as using non-fungible tokens (NFTs) as an analog to digital object identifiers (DOIs).

Technologies evolve quickly, resulting in a treadmill effect whereby older versions of software become deprecated quickly and competitors arise quickly. As a result, pipelines dependent on these technologies run the risk of failed reproducibility over time. What is needed is an approach to compile or provide a containerization wrapper that is invariant to changes in time to maximize the likelihood of conformance to standards and backward compatibility. Given appropriate hardware and software preconditions, and with the ability to compile, bind data, and have a machine-readable means of interpreting metadata enough information can be made available to an experimental sandbox to reproduce scientific experiments.

References:

[1] A. C. Chang and P. Li, "Is Economics Research Replicable? Sixty Published Papers from Thirteen Journals Say 'Usually Not,'" *SSRN Electron. J.*, 2015, doi: 10.2139/ssrn.2669564.

[2] F. Mölder *et al.*, "Sustainable data analysis with Snakemake [version 1 ; peer review : 1 approved , 1 approved with reservations]," *F1000Research*, no. May, pp. 1–25, 2021.

[3] E. Bisong, "Kubeflow and Kubeflow Pipelines," in *Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners*, Berkeley, CA: Apress, 2019, pp. 671–685.

[4] G. M. Kurtzer, V. Sochat, and M. W. Bauer, "Singularity: Scientific containers for mobility of compute," *PLoS One*, vol. 12, no. 5, pp. 1–20, 2017, doi: 10.1371/journal.pone.0177459.

[5] T. Gamblin *et al.*, "The Spack package manager: Bringing order to HPC software chaos," *Int. Conf. High Perform. Comput. Networking, Storage Anal. SC*, vol. 15-20-Nove, 2015, doi: 10.1145/2807591.2807623.

Data Object Distribution for Experimental Science Pipelines

Justin M. Wozniak,[1] Ray Osborn,[2] and Jacob Ruff[3]
[1] Data Science and Learning Division (ANL), [2] Materials Science Division (ANL),
[3] Cornell High Energy Synchrotron Source

Topic: Fine-grained, wide area data management and access for experimental science.

In order to perform both data reduction and more advanced analysis, such as machine learning (ML), on data collected at major facilities such as the Advanced Photon Source, it is largely the responsibility of the visiting scientists to navigate the patchwork of computational resources available at facilities themselves or to transfer large volumes of data to their own institutions. If an experiment involves multiple collaborators who each contribute to different modes of data analysis, the experimental team may have to transfer TBs of data to multiple locations. For example, a team in Argonne's Materials Science Division regularly collects several TB/day at Sector 6 of the APS. This entire volume is currently streamed to an APS data cluster for initial data reduction, then transferred to MSD for spectral analysis and, in parallel, copied to Cornell for ML via Gaussian mixture modeling (GMM) for unsupervised Bragg peak clustering, then to ALCF Theta to fit parameters. The raw data are archived by the APS for 1-2 years, but the intermediate results from each stage of analysis are distributed over multiple machines in multiple locations, as shown in Figure 1. Furthermore, the rate-limiting step for each stage of analysis is the data transfer (1 day of transfer for 15 minutes of analysis!). The Management and Storage of Scientific Data community could have a big impact on improving the data movement costs of this workload by enabling and optimizing remote data operations for a range of data creation and access patterns.

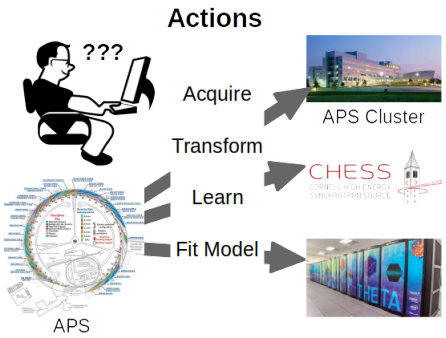


Figure 1: Multiple stages in data analysis workflow distributed across multiple computing sites, each with differing processing and I/O capabilities. Whole data sets are copied in bulk, incurring complexity and inefficiencies.

This white paper sketches an alternative paradigm that could profoundly impact how facilities handle large-scale experimental data in the future. In our approach, users are presented with a unified view accessible to all collaborators, which is populated on-demand with experimental data and/or theoretical simulations from multiple sources, and synchronized with distributed remote locations (cloud, exascale computing facilities) as needed. From the user perspective, this will have three advantages: (1) accelerated analysis and learning pipelines due to reduced data transfer overheads; (2) improved integration of data analysis and advanced modeling (digital twins), and (3) increased productivity due to less management overhead and development overhead. To this end, we need to design and develop a portable, adaptable infrastructure that exposes high-level data manipulation primitives to filter, query and assimilate simulation/experiment/collaborator data into unified views at fine granularities, while optimizing interactions with the remote data sources through a combination of local caching and data planning strategies.

The choices a scientist makes in performing this post-pipeline analysis are usually specific to the particular scientific question being addressed by the experiment, and therefore impossible to encapsulate in a predefined pipeline. We envision that pipeline construction by exposing Python network-accessible object interfaces. The community is already moving toward Python for sequential processing and basic

data manipulation, while performance-critical sections are expected to be written in C, C++, or Fortran, and exported to Python. Such libraries are likely to be usable in other settings, such as high-performance computers. The ongoing development of high resolution, high frame rate detectors and the x-ray and neutron scattering science capabilities they offer has created a data management and I/O challenge. Large experimental datasets must be rapidly stored and managed for near-real time analysis. The construction of analysis pipelines partially automates the analysis process, transferring data to storage and among multiple analysis software packages, but approaches are often too rigid for dynamic studies by human or learning agents.

It is as important therefore that we have a plan for ensuring that the results of any pipeline are accessible in a convenient and reasonably standardized form as it is to develop tools for constructing the automated pipelines in the first place. It is no longer sufficient to deposit the data in an archive, if the user is then required to download it in order to perform any followup operations. Current methods of data management at large-scale facilities have not adapted to the needs of facility users as data volumes have grown and the speed of data collection accelerated. Facilities typically provide medium-term stores to archive the data, but these are only accessible as file repositories through SFTP or Globus.

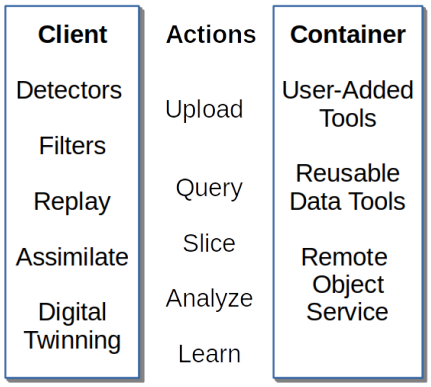


Figure 2: Multiple stages in data analysis workflow supported by remote data tools served from the containerized service library, including Pythonic objects for learning from small slices of large data.

We have to develop an I/O abstraction and architecture that allows fine-grained access to the data without requiring significant data transfers. Thus we propose several key I/O abstraction challenges: 1) Detector **ingest, filtering and upload (IFUP)**, via Pythonic filter plug-ins wrapped around advanced buffering and staging with multiple back-ends; 2) **Fine-grained local caching** that maintains the illusion of a fully available local view, but efficiently populates it on-demand and maintains its coherence in the background; 3) **Orchestration of remote data sources,** which automatically selects the best remote source to interact with (based on proximity, availability of data, permissions) and keeps the remote data sources synchronized; 4) **High level indexing and query support** to extract, group and present the data to analytics and learning tasks using familiar plug-and-play abstractions, from multiple data services.

Our approach is to develop a toolkit of learning and analysis-ready Python libraries that can be integrated into workflows by users from a broad range of disciplines, depicted in Figure 2. We envision a pluggable framework in which user Python fragments can be wrapped around existing IFUP technologies (e.g., Globus) but with a range of callback points controllable via Python tools. Fine-grain local caching will be addressed at lower levels, investigating how to efficiently buffer and issue fine-grain put/get operations in bulk to hide the latency of accessing remote data sources without compromising coherence. To this end, we envision approaches based on snapshot isolation that can take advantage of related efforts. Enhanced data analysis and learning will be enabled through extensions to a remote object toolkit which provides a familiar numerical interface, and optimizable aggregated data pipelines for deep learning frameworks. For example, in the GMM case, a container library entry for Bragg peak identification would be run at data ingest time, and the learning agent would walk the resulting peak index over remote object interfaces to quickly and efficiently produce usable results which are also stored for use by others.

XD/ML Pipelines: Challenges in Automated Experimental Science Data Processing

Justin M. Wozniak, Ryan Chard, Kyle Chard, Bogdan Nicolae, and Ian Foster
Argonne National Laboratory
woz, rchard, chard, bnicolae, foster @anl.gov

Topic: Data-management support for AI and complex workflows.

I. Challenge

Experimental data processing pipelines are growing not just in volume and velocity but also in complexity, as machine learning (ML) methods become an ingrained part of the workflow at multiple levels. Data access and transfer rates are often a limiting factor in overall time-to-completion. These workflows, here dubbed Experimental Data for Machine Learing (XD/ML) pipelines are also heterogeneous, consisting of phases that have different performance limits.

Typical XD/ML pipelines contain multiple phases relevant to the study of data management architectures. These include *data capture:* encompassing retrieval from the instrument, fast edge inference, and staging to temporary storage; *indexing:* extracting and synthesizing metadata, loading into catalogs and databases; *reconstruction:* moving selected instrument data to high performance computing (HPC) for full first-principles analysis; and *model training:* using reconstructed data to (re)train edge inference models. Each of these phases could involve a full read and write of the dataset to persistent storage, but many optimizations are typically employed. Additionally, in the presence of ML-based, application-aware, policy-based workflow decisions, these steps could proceed in different orders, be omitted, or re-executed in loops.

The data access workloads in complex XD/ML pipelines are qualitatively more difficult to diagnose, validate, optimize, and reason about when compared to traditional data acquisition patterns. This is not only because of the inherent difficulty of explaining ML models, but also because the ecosystem of data management tools and services is not tightly integrated. Ultimately, this prevents researchers from explaining why a particular result was obtained, sharing training data with others, or reproducing experiments [1].

II. Opportunity

Experimental science projects continue to advance from a manual "collect data, analyze later" paradigm to an automated, AI-enhanced, self-documenting "collect, analyze, feedback, document" paradigm. Project teams typically assemble one or more "flows" to perform a series of tasks, usually associated with particular available hardware. Over time, multiple flows are assembled into an application suite, generalized for multiple uses, and made portable to multiple data acquisition and processing systems. Scientific developers in these contexts, however, lack an ecosystem in which such workflows can be rapidly assembled, ported, scaled, and optimized; here we describe how the problem space can be bridged to a Management and Storage of Scientific Data (MSSD) context.

These flows have different computational requirements and priorities: for example, in serial crystallography experiments it is important that quality control and stills processing [2] be performed within seconds, and can employ local accelerators, while structure determination requires HPC but can be delayed if needed. Ptychography experiments that process intensive data streams on an HPC system, however, can introduce severe I/O bottlenecks due to, for example, the random sample loading used in model training to reduce bias. AI/ML-specific sample loading and caching optimizations are essential, with special emphasis on awareness of flow and data streaming. Additionally, flows may be controlled by nearly opaque ML models, making dynamic decisions about progress in the overall experiment, (e.g., when new data inferred to be of type D are produced, retrain model M, record outputs in catalog C, and proceed to the next sample material).

A. Exposing data access challenges through policy engines

Policy engines must be deployed at one or more levels to ease the interoperation of flows and regulate users and facilities. Such policy engines need not be centralized. A handful of key policy points are near-universally needed for flow management. These include: *deadline:* a flow must be started or completed by a particular deadline; *priority:* a flow should be prioritized relative to other flows from the same user, group, or facility; *resource limit:* a flow component should not consume more than a predetermined limit in terms of resource allocation used; *locality:* computation should minimize data transfer, favor use of an accelerator, or execute close to an instrument; *application-specific conditions:* for example, an ML model should be trained until it reaches a threshold metric. At the present time, teams are integrating ML methods into pipelines to minimize human intervention with the workflow, e.g., for automatic detection of regions of interest, mechanically cropping diffaction patterns, and enhancing reconstruction with AI/ML techniques.

Generalized flow policy management offers a potential interface between XD/ML researchers and the MSSD community. By exposing the structure of flows and exploiting MSSD optimizations, high-quality, portable, re-usable flows could be developed.

Expecting the emergence of a fully generalized, centralized policy engine is unreasonable. Rather, steps toward distributed policy management can be taken following the approach of stable distributed control systems [3]. This approach would start with the development of common performance control interfaces across services and tools. Then, local, constrained policy decisions and optimizations could be made. Meta-controllers could be deployed as applications are scaled up to negotiate broader problems, and optionally integrated with centralized policy services.

The goal of distributed service negotiation is not unreachable. Interoperable software and services are necessarily being developed in the HPC community. For example, the E4S [4] and Spack [5] projects have built up a very large suite of build routines enabling HPC teams to integrate and even link many tools together. Similarly, the ExaWorks project [6] is building an interoperable ecosystem of workflow tools that can build up nested workflows and other compositions.

B. Connecting flows to streams with data movement optimizations for continual learning

In this context, experimental science flows governed by policies that control data-in-motion introduce opportunities to optimize data management at multiple levels. In terms of abstractions, it is not sufficient to reason about input data as a static sequence of bytes available from the beginning (either as a file on parallel file system or a set of key-value pairs in an object store). Instead, data arrive continuously as a stream and computations need to be started as soon as possible, then adapted to process new data on-the-fly, as they arrive. This raises an entirely new set of challenges in the ML space. For example, a DNN model training cannot simply update the model by processing new mini-batches, as this leads to the problem of *catastrophic forgetting* (bias towards newest samples). On the other hand, accumulating all streamed data and retraining from scratch every time new mini-batches are available is not feasible either. Therefore, there is a need for advanced data streaming abstractions that look and act like normal streams, but can be augmented with policies to transparently mix historic and new data in order address application requirements.

In the example considered above, one can imagine data streams that automatically cache representative training samples based on a selection policy. Then, when a new mini-batch arrives, it is automatically augmented to include historic representative training samples, which effectively results in a transparent rehearsal for the DNN training without the need to modify the training process itself. Naturally, the design points of such abstractions have important implications on the whole storage stack, including: *locality:* where to cache and/or persist historic data; *categorization:* whether to handle live data and historic data separately or together; *hierarchy:* how to leverage multiple storage levels; *new hardware:* how to take advantage of persistent memory to reduce the performance gap between accessing historic data and live data.

C. Capturing, interpreting, and replaying automated data access decisions

Given that data access is supported by policy-level tools, it becomes possible to automate the inspection of flow progress and to make created datasets FAIR. Provenance records generated by workflows must be augmented with model version history so as to record how a possibly complex ensemble of variably-accurate models have been trained and updated over time, because these models impact the MSSD-relevant data accesses made by the system. The system would thus create a structure so the user can ask how a workload pattern was obtained, in the form of a provenance history.

III. Timeliness

The identified challenges and opportunities are highly relevant in the next 2-3 years. At Argonne, data flows between APS and ALCF are already being used by multiple applications. Without addressing these challenges, performance and scalability will be limited, because the benefits of advanced processing rates available on emerging exascale systems will be lost to inefficiencies in data movement. The Braid project at Argonne is addressing some of these challenges in deploying workflows based on Globus Flows [7] and the Braid Provenance Database [8].

References

[1] J. Borycz and B. Carroll, "Implementing FAIR data for people and machines: Impacts and implications - Results of a research data community workshop," 2020.

[2] M. Wilamowski, D. A. Sherrell, G. Minasov, Y. Kim, L. Shuvalova, A. Lavens, R. Chard, N. Maltseva, R. Jedrzejczak, M. Rosas-Lemus, N. Saint, I. T. Foster, K. Michalska, K. J. F. Satchell, and A. Joachimiak, "2'-o methylation of RNA cap in SARS-CoV-2 captured by serial crystallography," *Proceedings of the National Academy of Sciences*, vol. 118, no. 21, 2021. [Online]. Available: https://www.pnas.org/content/118/21/e2100170118

[3] E. Camponogara, D. Jia, B. Krogh, and S. Talukdar, "Distributed model predictive control," *IEEE Control Systems Magazine*, vol. 22, no. 1, pp. 44–52, 2002.

[4] M. Heroux, J. Willenbring, S. Shende, C. Coti, W. Spear, J. Peyralans, J. Skutnik, and E. Keever, "E4S: Extreme-scale Scientific Software Stack," in *Collegeville Workshop on Scientific Software*, 2020.

[5] T. Gamblin, M. P. LeGendre, M. R. Collette, G. L. Lee, A. Moody, B. R. de Supinski, and W. S. Futral, "The Spack package manager: Bringing order to HPC software chaos," in *Proc. SC*, 2015.

[6] A. Al-Saadi, D. H. Ahn, Y. Babuji, K. Chard, J. Corbett, M. Hategan, S. Herbein, S. Jha, D. Laney, A. Merzky, T. Munson, M. Salim, M. Titov, M. Turilli, T. D. Uram, and J. M. Wozniak, "ExaWorks: Workflows for exascale," in *Proc. WORKS @ SC*, 2021.

[7] R. Chard, K. Chard, S. Tuecke, and I. Foster, "Software defined cyberinfrastructure for data management," in *Proc e-Science*, 2017.

[8] J. M. Wozniak, Z. Liu, R. Vescovi, R. Chard, B. Nicolae, and I. Foster, "Braid-DB: Toward AI-driven science with machine learning provenance," in *Proc. Smoky Mountains Conference*, 2021.

Title: Toward a machine-actionable future for the DOE science mission

Authors: Katie Knight <knightke@ornl.gov>, Kuldeep Kurte <kurtekr@ornl.gov>, Rafael Ferreira da Silva <silvarf@ornl.gov>, and Valentine Anantharaj <anantharajvg@ornl.gov>
Affiliation: Oak Ridge National Laboratory, Oak Ridge, TN

Topics:
- Interoperability in FAIR digital objects, and among FAIR infrastructure
- Stewardship of scientific data

Vision: We envision a future in which the US Department of Energy (DOE) Office of Science (SC) R&D community will soon be able to easily identify, acquire, adapt, merge, and reuse scientific data to develop and implement bespoke hybrid applications to support the DOE mission. The collection of data assets in DOE data repositories will be findable, accessible, interoperable and reusable (FAIR) during the era of machine-actionable data, as discussed in Wilkinson et al. [1]. The collection of benchmark scientific data supporting the DOE science mission and the associated digital assets and artifacts will advance Advanced Scientific Computing Research (ASCR) goals toward democratizing (autonomous and egalitarian) access to science data. The collection of scientific data will also be complemented by semantically enriched metadata, which will help advance the goals of other external frameworks in "relating data, models, and tasks" [2].

Challenges: All five program offices within the DOE Office of Science leverage ASCR facilities to meet their computational needs and accomplish their science goals. The ongoing COVID-19 pandemic has revealed the necessity of "virtual collaboration, enhanced automation, autonomous controls, and robotics [which] could open up new ways of performing complex experiments, collaborating, and accelerating scientific breakthroughs" [3]. In the context of scientific data management and stewardship, this challenge extends beyond collaboration among science teams to autonomous machines collaborating (or even contending with one another) to facilitate breakthrough science via "a seamless integration of research infrastructure," which also implies seamless data flows across organizations and facilities while concurrently negotiating across legal frameworks [4,5]. Here, we are primarily focusing on the challenges of interoperability in actionable FAIR digital objects (FDO) toward facilitating ease of discovery, access and manipulation in the context of R&D toward the goal of knowledge discovery and innovation. As noted by Wilkinson and coauthors, the grand challenge here is in "assisting machines in their discovery and exploration of data," which then "becomes a first priority for good data stewardship."

Opportunities: The 2020 SC User Facilities Roundtable recognized that "data, computing, and networking infrastructure are critical for scientific productivity; they are the substrates the research community uses to explore, create, and share information" [3], clearly identifying that the first class opportunity that the SC user facilities pioneer the development, implementation and stewardship of FAIRecosystem(s) comprised of FDOs and FAIR workflows. There is a strategic opportunity for ASCR to provide the leadership across all U.S. agencies in realizing the vision for integrated ecosystems of experimental facilities, observing systems, computational resources and data assets – all interlinked by means of interoperable FAIRecosystems and services.

Much work remains to be done in defining domain-specific FDOs [6], characterized by their persistence, uniqueness and in being actionable [4]. They have so far been introduced into the European Open Science Cloud (EOSC) as a means to turn FAIR principles into practice and assist with establishing cross-disciplinary infrastructures, but research is still needed to understand appropriate methods to bind all relevant, domain-specific information to a stable digital entity, as well as how these digital entities may be linked across domains. Interoperability must be expressed in the metadata, where data is described using a "formal, accessible, shared, and broadly applicable language for knowledge representation" as per FAIR's definition of the I1 indicator [1]. It is also necessary that metadata: adopt "vocabularies that follow FAIR principles" (I2); and "include qualified references to other metadata (I3)." As more progress remains possible with the I1, I2, and I3 indicators, standardized formats and vocabularies that assist with inter- and cross-domain knowledge representation will be essential for linking data assets.

Timeliness and Priorities: The DOE scientific community is striving toward adhering to the FAIR data principles. Several DOE projects have already achieved success in making the data assets findable and

accessible. There are also many examples of reusability of data assets, often enabled via standardization and synthesis of metadata. Progress is being made toward making the data more reusable in the sense of FAIRness, not only within the same domain science community but also across collaborating science communities – for example in producing, synthesizing and consuming climate projections by collaborators across ASCR, BER and BES. Various science communities are still working toward defining semantic models that will facilitate linking data resources, which is critical for the data assets to be successfully integrated as FDOs in a framework[1] for true interoperability, especially across domains.

During the past decade, scientific data at ASCR facilities (both experimental and computational) are growing at unprecedented rates that have outpaced the DOE scientific community's ability to exploit the full potential of the data. We are approaching the era of acquiring experimental data at unprecedented rates; and high-fidelity simulation experiments are now generating multi-petabytes of simulation data [7]. A timely investment by ASCR in FAIRecosystems will enable scientists to develop autonomous systems, that will enable them to: (1) dynamically exploit resources where and when available, often during a limited window of opportunity, while preserving provenance for reproducibility; (2) derive innovative methods and new hybrid models that respect the laws of nature while aided by advances in machine learning and artificial intelligence that are explainable; and (3) extract knowledge and information from diverse and heterogeneous sources.

References

[1] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, et al. "The FAIR Guiding Principles for Scientific Data Management and Stewardship." *Scientific Data* 3 (2016): 160018.

[2] K. Fagnan, Y. Nashed, G. Perdue, D. Ratner, A. Shankar, and S. Yoo. "Data and Models: A Framework for Advancing AI in Science." United States, 2019. Web. DOI: 10.2172/1579323. http://www.osti.gov/biblio/1579323.

[3] U.S. DOE. 2021. Office of Science User Facilities: Lessons from the COVID Era and Visions for the Future. Report from the December 2020 Roundtable. DOI: 10.2172/1785683.

[4] Lehväslaiho, Heikki, Parland-von Essen, Jessica, Behnke, Claudia, Laine, Heidi, Riungu-Kalliosaari, Leah, Le Franc, Yann, & Staiger, Christine. (2019). D2.1 Report on FAIR requirements for persistence and interoperability 2019 (v1.0). FAIRsFAIR. https://doi.org/10.5281/zenodo.5535719

[5] Shankar, Mallikarjun, and Lancon, Eric. *Background and Roadmap for a Distributed Computing and Data Ecosystem*. United States: N. p., 2019. Web. doi:10.2172/1528707.

[6] G. Strawn (2019). *Open Science, Business Analytics, and FAIR Digital Objects*. *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, pp. 658-663, doi: 10.1109/COMPSAC.2019.10283.

[7] U.S. DOE. 2020. "DOE National Laboratories' Computational Facilities - Research Workshop Report". United States. https://doi.org/10.2172/1601798.

[1] https://fairdigitalobjectframework.org/

Towards an International Portal of Ontologies and Metadata Standards for Science

Katie Knight, Swen Boehm, Olga A. Kuchar, Alex May, Rohit Srivastava
knightke@ornl.gov; boehms@ornl.gov; kucharoa@ornl.gov; mayab@ornl.gov; srivastavar@ornl.gov
Oak Ridge National Laboratory, Oak Ridge, TN

Topics Addressed:

- Devising metadata management infrastructure to support FAIR principles
- Opportunity to build a metadata standards repository and integrated tools to facilitate cross-domain and facility optimization of scientific data management

Challenge: By the end of 2020, there was an estimated 44x10^21 bytes (44ZB)[1] of accumulated data. To give an approximation of relative size, there are 352 x 10^18 gallons of water in the Pacific Ocean. Using this as a reference, 44ZB is 125 Pacific Oceans partitioned into individual gallons. The scientific community is drowning in data and the new problem we face is how to make sense of it.

Part of this sense-making lies in metadata, which enables navigation within and across data stores: without metadata, data loses its usefulness. That is, good science requires good metadata. However, most metadata is poorly modeled (or non-existent) and requires manual, labor-intensive processes to "map" metadata before value creation can begin. Furthermore, if the community looks to FAIR as a roadmap for better metadata management, FAIR stresses that "Metadata needs to meet domain-specific standards" [1], and yet, there are limited metadata standards available within every domain (for instance, energy research lacks such a standard [2]) and, where such standards do exist, there is no centralized location to find and make use of them. Constructing metadata that links technologies, methodologies, and infrastructures, let alone individual data units, across domains will require not just extensive labor, but a comprehensive knowledge of cross-domain metadata standards.

In addition to standards, metadata is often time-sensitive and will require adequate versioning. Scientific research that includes scaling, analysis, and comparing heterogeneous data with different levels of granularity across groups and domains is complex. This will require adequate metadata that describes not just present events, but those that have already happened (i.e., legacy data) as well as those that have not yet happened. As data changes over time, so will the standards and ontologies to make sense of that data. The metadata management infrastructure of the future will require more than just a clever technological implementation; it will require a deep understanding of domain-specific knowledge, an easy-to-use means to translate that knowledge from one domain to the next, and a clear cognitive map to provide appropriate context for researchers as knowledge changes over time.

As of now, data tagging is frequently the proposed solution (e.g., [3]). As such, data tagging methods are often reinvented, which then generates technical debt related to identifying data tag relevance and linkages. The major challenge, then, is simple to characterize, but appropriate planning and execution will be necessarily complex: the scientific community needs to facilitate metadata (re)use and comprehension across scientific domains without placing the bulk of this cognitively arduous task (labor, comprehension, versioning) on the overwhelmed scientist.

Opportunity: We are aware of tools like BioPortal[2], an open, automatically updated repository of versioned biomedical ontologies stored in various formats accessible via Web browsers and services, and the Cedar Workbench[3], which allows scientific researchers and data producers to generate high-quality,

[1] https://news.uci.edu/2021/03/01/dna-data-storage-redesigned/
[2] https://bioportal.bioontology.org/
[3] https://metadatacenter.org/

semantically enriched metadata with terms from controlled vocabularies and ontologies (and which is connected with the BioPortal). These two tools are domain-specific – namely, to help the biomedical research community – but also provides a beautiful example of what is possible, given time and resources: (i) a repository of community approved ontologies, vocabularies, and mappings; (ii) related widgets in that repository that make seamless integration with existing tools possible for developers; and (iii) a separate front-end tool connected to this ontology/vocabulary repository that aids researchers and data producers with metadata creation and mappings either on-the-fly or by API integration with existing tools.

We propose the research and development of both such a repository as well as a complimentary metadata facilitation tool for the wider science community, with an eye to cross domain and facility use. DOE science labs are perfectly positioned for such research, as they represent a huge breadth of science domains (and as such can survey and test ontology availability, utility, and (re)use) and can provide the necessary scope of research expertise: information scientists and related experts in knowledge management, software engineers, and domain scientists with a vested interest in facilitated metadata management and connected data infrastructure. With the emergence of Web3.0, ontologies, data dictionaries, and standards are ideal candidates to be published and managed with new tools, like smart contracts that define rule sets capable of locking down interface rules for metadata curation, provenance using blockchain, and developer APIs using smart contract tools like Brownie and Truffle.

Timeliness: The European Open Science Cloud (EOSC)[4] and other international Open Science initiatives like BOOST 4.0 [4] are underway now that address the need for developing large-scale data-driven architectures that have adequate metadata management to aid in algorithm flexibility, analytics, analysis, and sharing. This is an excellent opportunity to leverage the international interest in open, shared data, and provide tools for applying rich, domain-specific metadata to the wider science community. Managing metadata on peta- and exascale file systems is challenging, and an active research topic. Several options to solve scalability challenges are addressed in [5]. Moreover, cross-facility initiatives like ORNL's INTERSECT project that aim to build and deploy a scalable system-of-systems environment that interconnect high performance computing, edge computing, data analysis, and experimental instruments to enable autonomous "self-driving" scientific experiments will need effective metadata tools to facilitate this interconnectivity; enabling the (re)use of domain ontologies and vocabularies will be essential and should happen in tandem with architecting these systems.

References:

[1] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, et al. "The FAIR Guiding Principles for Scientific Data Management and Stewardship." Scientific Data 3 (2016): 160018.

[2] Wierling, A. et al. (2021). FAIR Metadata Standards for Low Carbon Energy Research—A Review of Practices and How to Advance. *Energies*. 14. 6692. 10.3390/en14206692.

[3] Y. Wang, "Innovation Mode of Intelligent Tracking and Positioning Service for Logistics Enterprises under the background of Data Tagging," 2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC), pp. 381-384, doi: 10.1109/ICESC51422.2021.9532746.

[4] Holom, R. et al. (2020). Metadata management in a big data infrastructure. Procedia Manufacturing. 42. 375-382. 10.1016/j.promfg.2020.02.060.

[4] https://eosc.eu

[5] Cha, MH., Lee, SM., Kim, HY. et al. Effective metadata management in exascale file system. *J Supercomputer* 75, 7665–7689 (2019). doi: 10.1007/s11227-019-02974-8

Understanding the AI in FAIR: Data Management Support for AI and Complex Workflows

Katie Knight (knightke@ornl.gov), Rafael Ferreira da Silva (silvarf@ornl.gov),
Sean R. Wilkinson (wilkinsonsr@ornl.gov)
Oak Ridge National Laboratory, Oak Ridge, TN

Topic Addressed: Providing data management support for AI and complex workflows

Challenges:

Artificial intelligence (AI) and machine learning (ML) techniques are becoming popular within the scientific community [1, 2]. Workflows increasingly integrate ML models to guide analysis, couple simulation and data analysis codes, and exploit specialized computing hardware (e.g., GPUs, neuromorphic chips). These workflows inherently couple various types of tasks such as short ML inference, multi-node simulations, long-running ML model training, etc. They are also often iterative and dynamic, with learning systems deciding in real time how to modify the workflow (e.g., by adding new simulations or changing the workflow altogether).

Workflows empowered with ML techniques largely differ from traditional workflows running on HPC machines. While workflows (i.e., one large-scale application simulating a scientific object or process) traditionally take little input data and produce large outputs, ML approaches target model training, which usually requires the input of a large quantity of data (either via files or from a collection of databases) and produces a small number of trained models. These models can be standalone applications, or even embedded within larger traditional simulations. There exists an inherent tension between traditional HPC, which evolved around executing large capability codes, and AI-HPC, which requires the coordinated execution of many smaller capability-scale applications (e.g., large ensembles of data generation co-mingled with inference and coinciding with periodic retraining of models).

With their reliance on data, effective AI-workflows should provide fine-grained data management and versioning features, as well as adequate data provenance capabilities [3]. This data management will have to be flexible: some applications and workflows might need to move data via a filesystem, while others could be better served from a traditional database, data store, or a streaming dataflow model. During inference, it may be best to couple the (lightweight) model as close to the data it is processing as possible. In any case, effective data management is a key feature of successful AI workflows.

Furthermore, any data management support for AI will require some type of privacy management: algorithms trained on data with PHI are clearly sensitive, but there may be other, less obviously sensitive data used in AI research that could be inadvertently exposed. Part of the support for such systems needs to assume that many scientists using AI may not be AI “experts”, and data management support should make it as clear as possible what these AI algorithms may/may not be used for. AI and ML packages have become incredibly easy to use, which makes them that much harder for domain scientists to explain.

Current AI workflow systems (e.g., MLFlow, AirFlow, Kubeflow, Pachyderm, etc.) are also difficult to deploy in complex research environments. These systems assume use of specific languages or environments that are not easily deployed where, say, the user lacks root access, needs to delegate tiered access permissions to specific portions of a workflow, or is beholden to institution-specific security policies.

AI algorithms are notoriously opaque [4]. Tracking the provenance of intermediate and final artifacts through all stages of the workflow is crucially important for engineers and scientists to understand and explain the output of a particular AI algorithm. Without such fine-grained metadata, it can be difficult or impossible to verify that an algorithm has executed as intended, much less that its results can be trusted.

Existing approaches to logging these artifacts allow flexible use, where users choose what to log, but this approach may result in omitted artifacts or partially logged artifacts.

Opportunity:
Here is an opportunity to address data management concerns related to provenance, privacy, data access concerns and permissions, artifacts, and overall explainability. We propose the following approach: to extend a currently available workflow management system as a prototype which can satisfy the constraints to be deliberately "paradoxical" by being opaque where security is concerned (the resulting model cannot be tampered with in order to recover the original anonymized data) but transparent where the automation and execution details are concerned (by tracking provenance and metadata about the execution for debugging purposes, for explainability, and for scrutiny by other scientists, which is critical to the scientific process). Doing open science on protected data is a challenging problem, but the FAIR principles [5] can still be observed at multiple levels of the workflow. Better tools would enable scientists to satisfy all these constraints automatically so that they can just focus on doing science; rather than reinvent the workflow wheel yet again, we propose to examine a current, commonly used tool (e.g., MLFlow) and extend its capabilities to effectively understand and map out a solution for provenance, privacy, permissions, artifact tracking, and explainability.

Timeliness:
Data management and workflows will be fundamental components in various cross-facility initiatives, such as INTERSECT at ORNL. As projects such as this develop and incorporate data and workflows from multiple domains, addressing issues related to provenance, privacy, access/permissions, artifacts, and overall explainability will be crucial to ensuring that these components are incorporated into the engineering process, and not addressed as afterthoughts.

References:

[1] Thessen A (2016) Adoption of Machine Learning Techniques in Ecology and Earth Science. One Ecosystem 1: e8621. https://doi.org/10.3897/oneeco.1.e8621

[2] Chen, RC., Dewi, C., Huang, SW. *et al.* Selecting critical features for data classification based on machine learning methods. *J Big Data* **7,** 52 (2020). https://doi.org/10.1186/s40537-020-00327-4

[3] Ferreira da Silva R, Casanova H, Chard K, Altintas I, Badia RM, Balis B, Coleman T, Coppens F, Di Natale F, Enders B, et al. *A Community Roadmap for Scientific Workflows Research and Development*. 2021 IEEE Workshop on Workflows in Support of Large-Scale Science (WORKS), 81–90, 2021.

[4] Angelov PP, Soares EA, Jiang R, Arnold N, Atkinson PM. Explainable artificial intelligence: an analytical review. *WIREs Data Mining Knowl Discov. (*2021);11:e1424. https://doi.org/10.1002/widm.1424

[5] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, et al. "The FAIR Guiding Principles for Scientific Data Management and Stewardship." *Scientific Data* 3 (2016): 160018.

Heterogeneous Memory System Framework for HPC

Kazi Asifuzzaman, Narasinga Rao Miniskar, Aaron R. Young, Frank Liu, Jeffrey S. Vetter
Oak Ridge National Laboratory, Oak Ridge, TN, USA
{asifuzzamank,miniskarnr,youngar,liufy,vetter}@ornl.gov

1 Challenge

The increasing demand for computational performance along with the slowdown of Moore's law results in the need for extremely heterogeneous systems with new accelerator designs, memory technologies, and memory hierarchies to continue improving application performance and reduce energy consumption. In the current environment, applications are written with the static mapping of their memory objects based on the kernel placement on the specific computing resource (CPU, GPU, FPGA) decided at design time. It is a challenge to provide an environment to the developer with application performance portability targeting extremely heterogeneous systems which have not only heterogeneous accelerators but also have heterogeneous memories. Recent state-of-the-art techniques have proposed either the run-time mapping of tasks to heterogeneous computing units with the static mapping of memory objects [1], [2], or have proposed the run-time selection of memory without considering task mapping constraints [3]. A Heterogeneous memory system framework is needed to address the performance and portability of applications for extreme heterogeneous systems, which studies the traditional memory mapping techniques, task mapping techniques and proposes a run-time adaptable heterogeneous memory mapping approach to work in conjunction with the heterogeneous task mapping approach. It is challenging to address this problem in multiple areas such as programming model, compiler support, and run-time framework.

There are some studies that partially address this issue. Narayan et al. [3] propose a novel page allocation approach to utilize heterogeneous memory systems at the memory object level, which conducts profiling and classification of memory objects offline. Wu et al. [4] propose a lightweight runtime solution called Unimem to minimize unnecessary data movement. Unimem works in phases of profiling, modeling, and placement of memory objects, and the concept is evaluated with non-cycle accurate simulation. Olson et al. [5] extend the application run-time layer with automated monitoring and implement an online data tiring solution that facilitates data allocation and placement across heterogeneous memory only based on the latency and if there is a benefit to migrating data. Although these studies provide great ideas and insights to approach the issue, a detailed and dedicated exploration is needed in order to develop a complete solution for accommodating future heterogeneous memory systems that can effortlessly accommodate a range of memory technologies.

2 Opportunity

Applications can have very different memory requirements. Research is required to profile applications to understand their memory requirements when determining how the application can best leverage the available memory. For example, an application can be latency-sensitive, or bandwidth-bound; while DDRx DRAM (still) provides the fastest interface but High Bandwidth Memory (HBM) provides superior bandwidth performance. There could be a need for low power (LPDDR) or Non-Volatile (STT-MRAM, ReRAM, PCM) memory for a specific application as well. Therefore, there would be a great opportunity to analyze a broad range of applications from a memory requirement perspective and this research can lead to interesting insights.

Concurrently, there would also be an opportunity to conduct elaborate analysis of candidate memory technologies, many of which are novel/emerging and lack a detailed understanding of

underlying technologies and concepts. A detailed exploration of these memory technologies would certainly enrich the community with useful knowledge.

The performance and portability challenges of heterogeneous memories need to be addressed with an extension to existing programming models [2] with unique input/output (IO) objects specification of computational kernels along with the task specification. The IO object's specification should have sufficient details for dynamic mapping. Researchers have to balance the details and gains. It also provides opportunities to support the specification with compiler (LLVM) extensions. Researchers have an opportunity to investigate dynamic mapping techniques for heterogeneous memories which include heuristics (genetic, dynamic programming, etc.) and machine learning techniques, and consider trade-off axes such as performance, energy consumption, complexity, memory usage, and bandwidth requirement.

DOE applications are expected to fully exploit the advantages of the proposed heterogeneous memory system framework with both performance gains and portability to variations of the emerging extremely heterogeneous systems. A efficient heterogeneous memory system would effortlessly facilitate allocation and mapping to memory sub-modules during run-time, while maintaining coherence, and preferably without placing a large burden on the programmer.

3 Timeliness

The end of simple technology scaling for easy performance gains along with the rise in diversity of computation accelerators and memory systems are resulting in extremely heterogeneous computing systems. New emerging memory technologies like ReRAM, SST-RAM, PCM, HBM, and HMC have matured and are now being included within accelerator designs. For example, FPGA designs are now including HBM and SSD attached memories. The open-source movement further transformed the emergence of new processing cores (RISC-V), computing accelerators (artificial intelligence, neuromorphic, HPC) along with the integration of new memory technologies. In response, we have already seen the emergence of true heterogeneous commercial platforms incorporating CPUs, GPUs, accelerators, and FPGAs each potentially including diverse sets of memory. Although static mapping of tasks to accelerators and static mapping of memory to memory devices has been researched before, we are now facing a new challenge of maintaining performance portability as codes are ported between heterogeneous systems with different accelerator and memory structures. Advanced dynamic mapping of both the accelerator and memory systems will be needed to maintain portability and performance in these increasingly heterogeneous and diverse systems without creating a large burden on the programmers to write system-specific code.

References

[1] Ruyman Reyes and Victor Lomüller. "SYCL: Single-source C++ accelerator programming". In: *Parallel Computing: On the Road to Exascale*. IOS Press, 2016, pp. 673–682.

[2] Jungwon Kim, Seyong Lee, Beau Johnston, et al. "IRIS: A portable runtime system exploiting multiple heterogeneous programming systems". In: *2021 IEEE High Performance Extreme Computing Conference (HPEC)*. IEEE. 2021, pp. 1–8.

[3] Aditya Narayan, Tiansheng Zhang, Shaizeen Aga, et al. "MOCA: Memory Object Classification and Allocation in Heterogeneous Memory Systems". In: *2018 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. May 2018, pp. 326–335. DOI: 10.1109/IPDPS.2018.00042.

[4] Kai Wu, Yingchao Huang, and Dong Li. "Unimem: Runtime Data Managementon Non-Volatile Memory-Based Heterogeneous Main Memory". In: *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*. SC '17. Denver, Colorado: Association for Computing Machinery, 2017. ISBN: 9781450351140. DOI: 10.1145/3126908.3126923. URL: https://doi.org/10.1145/3126908.3126923.

[5] M. Ben Olson, Brandon Kammerdiener, Kshitij A. Doshi, et al. *Online Application Guidance for Heterogeneous Memory Systems*. 2021. arXiv: 2110.02150 [cs.PF].

Profusion of Userspace I/O

Johann Lombardi - Intel Corporation - johann.lombardi@intel.com
Kevin Harms - Argonne National Laboratory - harms@alcf.anl.gov

Topic

A critical capability for improving latency and bandwidth of I/O from clients to storage systems is the use of kernel bypass for application I/O. Performing I/O without trapping into kernel space has many benefits for applications such as reducing context switch overhead, kernel resources from page cache, CPU cycles if specialized hardware is utilized and supporting advanced security models (e.g. RBAC) beyond traditional POSIX ACLs. The emergence of specialized compute accelerators has resulted in more complex memory subsystems that require specific handling of memory to optimize data movement. Userspace I/O enables a data path that provides efficient I/O for working with these new memory hierarchies and compute accelerators because they are not bound by kernel page handling and memory addressing requirements. Userspace I/O allows storage to move beyond POSIX APIs and more importantly POSIX semantics. A number of software packages that provide userspace I/O have been introduced with different APIs and behaviors.

Challenge

The proliferation of userspace storage APIs make it difficult for application developers to write portable code that can run on different systems. Those userspace APIs may serve different purposes like supporting a vendor-specific accelerator (e.g., NVIDIA GDS [1]) or allowing direct access to local (e.g., PMDK, SPDK BlobFS) or distributed (e.g., DAOS, UnifyFS, IME, CephFS, S3, GCS, Azure blob) storage without involving FUSE for performance reasons. Another example is the advent of io_uring that offers a new asynchronous and efficient kernel interface. The support matrix is not sustainable for application developers who don't want to become storage experts which limits the potential for utilizing advanced hardware features.

Opportunity

With this profusion of storage APIs and little common ground in the community, there exists an opportunity for DOE to step in and fill this need. There are two major parts needed, first is a common userspace kernel bypass interface that implements the POSIX API (i.e. libsysio [2]) which uses a common backend API that can be implemented for any storage system. Research is required to understand the impacts from new memory technologies (persistent memory, HBM) in the data path, impacts from new system architectures where memory is provided on several different devices (NIC, GPUs, Compute accelerators), and understanding what are the minimum/weakest semantics that provide a useable system. The findings of this work can then

be used to define a common backend userspace storage API. Ideally this enables different storage systems to provide optimized implementations of the backend API which can be used by developers directly or within the common POSIX implementation to support legacy/existing code. The backend API will be critical in properly exposing and handling memory so that remote memory operations can be used to move data directly onto accelerator units that potentially contain multi-level memory hierarchies.

A key part of this effort must be building a community around the solution. An open specification with a published governance model to allow participation across academia, government and industry would allow this effort to succeed where others have failed. This model allows for potential growth in both backend APIs as well as potential support for extensions to POSIX [3]. A reference implementation would allow for providing a functional (but perhaps not performant) implementation on any platform.

Timeliness

As the proliferation of compute accelerators continues, each with different capabilities and architectures, each of these needs to integrate with the main compute subsystem. These accelerators need to take input for computation or generate data as output for analysis. Currently, the space is evolving with vendors proposing different APIs. Acting now will allow DOE to drive the broader storage community toward a common vendor neutral API before the market becomes segmented with various incompatible APIs. The MPI standard occurred at just the right time to drive adoption of high performance interconnects using a standard communication interface which allows vendors to compete. On the other hand, no similar standard happened around GPUs and now codes require a heavy investment to move away from CUDA in order to run on hardware other than Nvidia. Also, with the explosion of many cloud computing offerings and increasing merging of HPC with cloud technologies, this provides motivation for software developers to make changes to applications where in the past they were extremely resistant to changes in the I/O path due to perceived value of the effort.

References

[1] https://docs.nvidia.com/gpudirect-storage/overview-guide/index.html
[2] https://sourceforge.net/projects/libsysio/
[3] http://www.pdsw.org/pdsw06/resources/hec-posix-extensions-sc2006-workshop.pdf

Composable Data Management Architecture

Modern HPC and AI workflows increasingly depend on availability of large amounts of scientific data across variety of computer systems. Some of this data comes directly from laboratory or field instruments, other data result from interim computational steps, and certain data may come from earlier collection or calculation that may have taken place years and even decades ago. As scientific research frequently involves data sharing on a global scale, security and access control requirements come into play. Achieving timely, secure, and economically viable access to data from computational systems becomes one of the key enablers of scientific workflows.

This concept paper describes a composable architecture for managing, storing, and moving scientific data. Rather than dictating an entirely new concept, it attempts to generalize modern data management techniques and classify existing components into “planes” by their function. This allows to break down otherwise complex problem into manageable pieces and point out areas of improvement that require focused research.

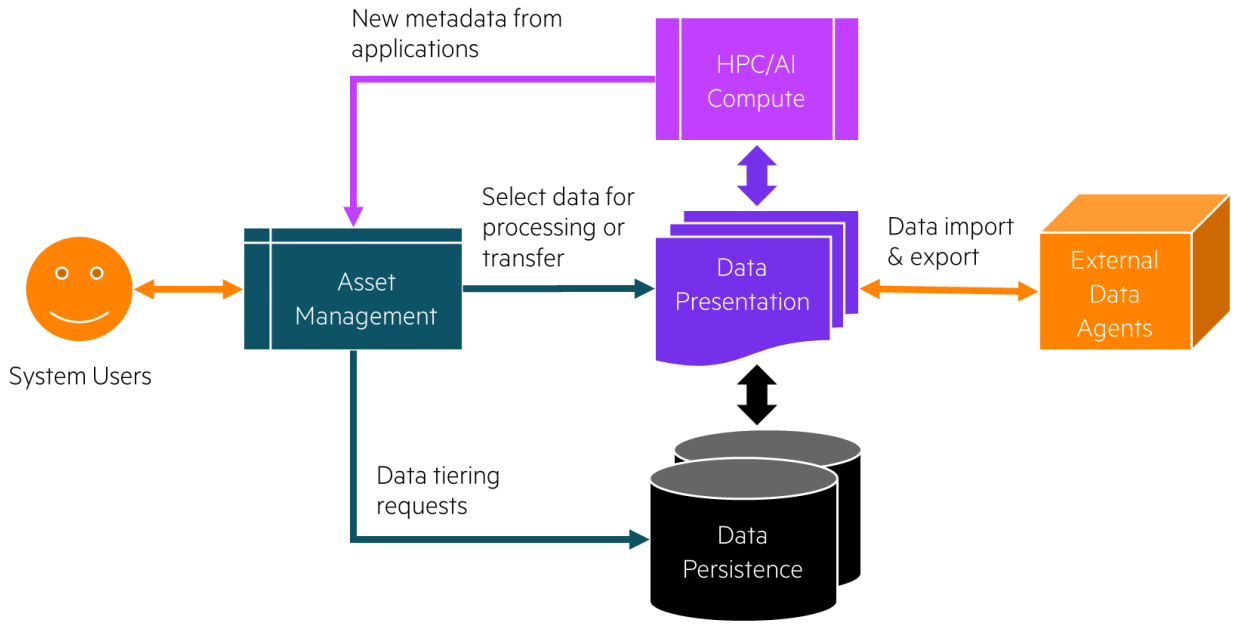
The proposed architecture consists of building blocks or, “components”, that exist within the following three “planes”:

- Asset Management plane. This is a domain-specific plane that relies on metadata to identify and point to data files, objects, and collections. Asset management implementation could be as trivial as a directory structure that follows a certain naming convention, or as complex as a searchable domain specific database with domain aware UX front-end that points to collections of files and S3 objects with opaque names and identifiers. The key parameter of this plane is ease of use.
- Data Presentation plane. This is traditional HPC/AI storage, such as parallel filesystem, NFS server, or an object storage system. Its main purpose is to deliver data and capture results produced by the applications running in HPC/AI compute systems. Components on this plane are not domain aware, and their key parameter is performance. Going forward, components on this plane may also include streaming data sources, high-performance ephemeral namespaces, key-value stores, etc.
- Data Persistence plane. Traditionally, this is a “data archive”. This component is responsible for safekeeping the data for a prolonged period. The data stored by this component is immutable and may not be accessible directly. For access, the data needs to be moved into Data Presentation plane. The data objects stored in Persistence plane are protected by erasure codes or by storing multiple copies on different media in multiple locations. This plane is not domain aware and is driven via API. The key parameters of this plane’s components are the cost of storage and the time it takes to produce or store data to or from Presentation plane.

The relationship among these planes is shown on the diagram below. The users interact predominantly with the components of the Asset Management plane. This helps them identify the data that is available in the Data Presentation plane or that is stored in the Data Persistence plane. The data may be readily available or may need to be restored via API calls. As the applications produce new data in the Presentation plane, they update the Asset Management accordingly. The data may also be imported from or exported to external agents, such as scientific instruments. This activity is also facilitated and tracked by components of the Asset Management plane.

Considering that there may be multiple components in each plane, this architecture can accommodate many currently practiced workflows. The architecture assumes that components of only one plane (Asset Management) are domain aware. The components of two other planes are domain agnostic and therefore can be generalized and delivered by an independent system vendor. The Asset Management components must take advantage of the APIs to the components of two other planes, particularly to the Data Persistence planes that does not have commonly used APIs such as POSIX (unlike Data Presentation layer).

By interconnecting one or more components of each plane to their counterpart(s) in other planes via APIs, practitioners can compose an HPC/AI data ecosystem of arbitrary complexity that is fit for a particular scientific workflow.



Title: Science Campaigns: A Paradigm for Scientific Discovery for Exascale and Beyond

Authors: Kshitij Mehta[1], Ana Gainaru[1], Norbert Podhorszki[1], Scott Klasky[1]
[1] Oak Ridge National Laboratory - (mehtakv, gainarua, pnorbert, klasky) @ornl.gov

Topic: Devising metadata management infrastructure to support FAIR principles

Challenges

A science campaign is a collection of simulations/experiments and analyses performed towards a larger goal of scientific discovery. Over a period of months or even years, science teams may run their codes and analysis using different parameters on different systems to achieve their science goal. Since the teams and resources are distributed, this typically leads to vast amounts of data spread over geographically distributed locations. This data needs to be analyzed for discovering knowledge. The challenge is both in ease of management of this distributed data and in the efficient mechanisms for the collaborators to find this data during a campaign. There is a need to analyze the data collectively although tools are generally built to manage files and not data collections. No abstractions or tools exist in the scientific community that treat campaigns as a core paradigm of computing. For example, scientists make simulation runs, and manually perform tasks such as copying data to an end point, organizing the files, writing analysis scripts that depend on this ad hoc data and file structure, organizing data into a bespoke directory hierarchy, and so on.

Fundamental challenges associated with data management, performant data analysis, and reproducibility are observed widely across various science domains[1,2,3]. While versioning systems are used to track application source codes, tracking input and output data and workflows is rare. ***When a science campaign completes, data is distributed and disorganized; there is no central repository that consists of all components (data, codes, workflows) of the campaign along with rich metadata to establish provenance***. Consequently, there is no easy way to analyze all data in a full campaign in a holistic way. This problem of metadata management and provenance tracking for a campaign is further exacerbated by the emergence of A.I. for science research. Data that constitutes A.I. workflows consist of training data, A.I. models, and dynamically evolving models using on-the-fly training. Managing all this data is ad hoc, unorganized, and ultimately inefficient. FAIR principles (Findable, Accessible, Interoperable, Reusable) have thus gained traction for both data [4] and workflows [5], with increased emphasis on reproducible A.I. However, rich metadata and core abstractions to represent science campaigns that consist of managing data and workflows are missing.

Opportunity

The above challenges can be addressed by considering the science *Campaign* as a core data-driven paradigm for computing in the exascale era and beyond. As opposed to applications simply being an instance of computational kernels that ingest data, perform computations, and generate output, a simulation must be associated with a 'Campaign'. ***A campaign must be a structured data store of all components in the campaign - different data objects, applications, workflows, and analyses codes, visualization tasks, performance provenance, plots, and different types of metadata to describe components***. It must have the mechanism to express links or actions between components that establish a knowledge graph from core components to science results. Detailed metadata about components along with rich labels and annotations for data can make a campaign Findable (F). Sophisticated federated infrastructures for distributed campaigns can make them Accessible (A). Labels, metadata, and digital object identifiers for various data objects in the campaign can make it interoperable (I). Rich, annotated data and workflows used for obtaining science results can make it Reusable (R).

The Campaign abstraction must be implemented by a Campaign Management System - a collection of APIs, tools, and a federated data store for storing all data associated with a campaign. High-level APIs with sophisticated options

for managing different data must be a core component of such a system. Applications must publish or store data using scientific data management frameworks instead of directly using conventional file I/O APIs. Internally, the campaign management system may use files, object stores, databases, or a combination of several technologies to store the raw data depending upon fundamental properties of data such as type, size, scale, persistence, and reusability. The flexibility of this approach is that it is decoupled from the underlying technology for storing data, which makes it scalable. The system must consist of a collection of services for efficient data management, analysis, and visualization. Using user-provided and automated provenance extraction, the system must capture rich metadata about data objects and workflows. The system must index the data internally for efficient query performance. Queries submitted from local or remote locations must be transparently translated into operations that locate the data and read it efficiently using sophisticated data management frameworks.

In addition to high-level interfaces for data management, there needs to be core support for additional options for managing data and composing workflows, as listed below.

- Data lifecycle: automated data movement - long term storage, local storage, transient data etc.,
- Data movement: interfaces for streaming or file-based data movement,
- Data compression/reduction: Policy-driven compression/reduction of data

In short, a high-level abstraction for managing data and information that decouples science applications from low-level system details can open a plethora of opportunities for optimizing data management transparently. ***A rich Campaign API centered around a federated Campaign data store and efficient query mechanisms for data and information retrieval can form the core components needed to build the infrastructure for supporting the FAIR principles for next generation science research.***

Timeliness

The emergence of federated computing, A.I. for science, and the ever-growing data scales pose additional challenges for verifiability and reproducibility of science results. More data are generated from simulations and experiments, and are distributed across different sites and further spread across different objects. As I/O bandwidth and storage capacities have not kept up with the increase in computing power, data has become a vital commodity, and efficiently analyzing it for extracting knowledge requires a structured and methodical approach. The concept of a *Campaign* as a central paradigm for associating data and workflows with scientific research can help achieve this objective.

References

1. Wan, Lipeng, et al. Data management challenges of exascale scientific simulations: A case study with the Gyrokinetic Toroidal Code and ADIOS. Oak Ridge National Lab.(ORNL), 2019.
2. Choi, Jong, et al. "Data Federation Challenges in Remote Near-Real-Time Fusion Experiment Data Processing." Smoky Mountains Computational Sciences and Engineering Conference. Springer, 2020.
3. Pineau, Joelle, et al. "Improving reproducibility in machine learning research: a report from the NeurIPS 2019 reproducibility program." Journal of Machine Learning Research 22 (2021)
4. Wilkinson, Mark D., et al. "The FAIR Guiding Principles for scientific data management and stewardship. Sci Data 3: 160018." (2016).
5. Goble, Carole, et al. "FAIR computational workflows." Data Intelligence 2.1-2 (2020): 108-121.

Topic: Object-based Interface for Massive Storage Disaggregation

Johann Lombardi – Intel Corporation – johann.lombardi@intel.com
Lance Evans – HPE – lance.evans@hpe.com

Position

The time has finally arrived to research, develop, and standardize a high-level persistent device interface that enables solid state storage infrastructure to scale from laptop to exaflop, to expose capabilities and speed of emerging low-latency storage, and to serve and operate on decades of data accumulating in the deepest bulk archives. A modern object interface is needed, that can be presented over local bus interfaces (e.g., PCIe via NVMe and/or CXL), that is also designed from the outset for highly scaled networking.

Challenges

A trend toward userspace storage with kernel bypass has opened opportunities to innovate. Available storage interfaces are inadequate, block is too primitive, NAS is hampered by file semantics, T10 OSD was conceptually sound but premature. A variety of object interfaces have emerged as alternatives to POSIX in the intervening years. Meanwhile the advent of NVMe has contributed momentum to development of new device interfaces that have been quickly integrated into the specifications ([1]). Examples include a new set of commands for key-value storage operations (i.e., NVMe-KV) and the support of zoned namespaces.

Innovators continue attempting - with good reason - to delegate useful functions into storage devices. As storage controllers become more powerful this becomes more practical. An obvious first target is to distribute important storage functions into them. The Flash Translation Layer, as a good example, already adapts a front-side (block) interface to media-specific layout, and handles important hidden functions in support of that interface (e.g. caching, power loss protection, wear levelling, garbage collection…). A higher-level more useful abstraction than block, is not a huge stretch.

But what features should such an interface have? The NVMe K/V spec is one example, but is too simplistic for resilient scalable storage. LSMT and other storage interfaces have emerged and are being used as mid-layer interfaces to DBs, object stores, etc, but haven't been scaled. NFS was arguably the most successful networked storage device interface, but is hampered by antiquated semantics. While tremendously useful, traditional block- or byte-addressable device interfaces haven't the functionality required to meet the needs of modern disaggregated and hyperconverged software-defined storage use cases. And block devices have significant security vulnerabilities; one bug, a rogue node, or an intruder can wipe out entire device regions, or entire storage systems. And software abstractions are required over many of these, to be reasonably consumed by applications or extended across the network.

Attempts to embellish block with computational storage functions have met with limited viability; function shipping into raw block devices is a primitive approach (NVMe computational storage model), and enabling device-local processing by embedding a traditional filesystem has been attempted, but is simply a stopgap – another indicator a more sophisticated access method is required.

A secure way to embed logic within a local persistent device, along with structured data, is needed. This is a perfect use case for object-oriented storage devices.

Discussion

Some core principles can guide contemplation of a modern object device interface.

Disaggregation of storage devices from compute nodes is beneficial in efficient use of resources. Trapping devices within a node forces fixed / overprovisioned allocation ratios of compute to storage, despite varied workloads. Disaggregation can programmatically allocate device bandwidth, IOPS, as well as capacity more efficiently.

Fabric bandwidth, latency, and scale affect the topology and effective reach of disaggregated storage. Chassis- or rack-scale fabrics offer lower latency and potentially higher parallel bandwidth. Datacenter-scale fabrics offer more flexibility of resource scheduling and data or device sharing opportunities. One size doesn't fit all, though a common storage access method is beneficial across all topologies.

A demand for (optional) data resilience drives distributed data placement across fault domains. Data must be placed close enough to computational units to meet performance requirements. Redundant data must be distributed widely enough to ride through faults while avoiding data access loss or data loss. And capabilities must be enabled for devices to be recovered and coordinated in large-scale deployments.

Redistribution of server- and client-side networked functions are required to scale out disaggregated storage. Traditional data path models that delegate resiliency to server-side storage controllers, and journaling to client-side filesystems or the like, must be inverted. Distributed transactions coordinated between clients and servers is required, at scale, without locks/bottlenecks.

A new, more sophisticated device interface beyond block or simple key-value, is needed to drive the solid-state era of our industry. Inevitable interdependent client-to-server and server-to-server functions require a more advanced device API. Such must be lean enough to support a broad set of layered use cases including file, K/V, container, database, and others. It must be useful when scaled down to a single device, operate at datacenter scale across a network, and have potential for coarse- or fine-grained security, all without limiting expected HPC performance.

Timeliness

The storage industry is now at a unique inflection point, with mature solid state storage media, powerful device controllers, an explosion of storage interfaces, and an opportunity to build the foundation for post-exascale data systems with advanced functions reaching aggregate bandwidths of hundreds of TB/s, billions of OPS, & single-digit-microsecond latencies. The potential of an object-based device interface is more compelling than ever.

Opportunity

Research is required to further evolve, stabilize, standardize, and enable an ecosystem with such an interface, and devices that embody it. An investigation would survey historical implementations and state-of-the-industry object interfaces (e.g. DAOS target, various OSDs / OSTs, LSMT implementations, filesystem and DB backends, etc). Missteps in prior similar standardization efforts must be considered and avoided. A practical demonstrator should be built to validate any proposed design. The result should be an open interface and potentially a protocol to extend the latest data interface standards (e.g., NVMe, CXL) for widespread adoption.

References

[1] https://nvmexpress.org/developers/nvme-command-set-specifications/

Supporting Next-Generation Data and Workflows

Lavanya Ramakrishnan, Oluwamayowa Amusat, Shreyas Cholia
Devarshi Ghoshal, Anna Giannakou, Daniel Gunter, Drew Paine, Gilberto Pastorello
Email: [LRamakrishnan,ooamusat,scholia,dghoshal,agiannoku,dkgunter,pained,gzpastorello] @lbl.gov

Topic: Data-management support for AI and complex workflows

Challenge: The growth in volume, variety, and complexity of data generated at experimental, observational, and computational facilities, is fundamentally changing the landscape of workflows seen in the DOE complex. Models and digital twins running on the edge that provide a closed-feedback loop to experiments and observations (e.g., decisions for sensor placement, resolution of data collection), data ingestion at data repositories, AI models running on the edge and HPC, and data analyses that combine experimental and simulation data are becoming increasingly commonplace. Today, there are limited capabilities available to support the lifecycle of data from acquisition to analysis, sometimes over a distributed set of resources. Data management in the future will include myriad goals from preparing products for consumption by models and AI, ingesting to make it available through search in data repositories, and archiving for long-term use and retrieval.

Next-generation scientific discoveries increasingly rely on a comprehensive and integrated strategy that brings together experiments, models, and data. Our current approach that separates data management from workflow management is no longer feasible since the data and processing life cycles are often intertwined, requiring support for real-time data management during data acquisition, data wrangling, processing, and analysis. For example, we need algorithms, methods, and tools to support AI and complex workflows that apply QA/QC (Quality Assurance and Quality Control) and provide access to diverse and large datasets, and manage the memory-storage hierarchy on current and future HPC systems.

Opportunity: The emergence of large-scale compute needs for experimental and observational data combined with new paradigms and technologies, including edge computing and AI, provide us an unusual opportunity to rethink the data management layer for scientific applications to satisfy the needs of a wide range of applications from sensor data (e.g., the Watershed SFA) to images collected from instruments (e.g., light sources, microscopes, telescopes). This white paper captures a few key areas of data management that require immediate and timely attention - a) managing distributed data, b) data wrangling, c) metadata and provenance management.

Managing Distributed Data. Scientific data today is already distributed across organizations, projects, and facilities – often managed in an ad-hoc manner by individual users. The growing volumes of data and the emergence of network, edge, and cloud storage will further exacerbate the situation. Data storage and access tools will need to allow users to seamlessly and transparently access storage regardless of location. The data storage layer will need to provide seamless access to relational, time series, and non-relational databases, filesystems, block and object stores, etc. In addition to being accessible for analyses and processing, it is critical to provide methods and tools for larger community access through repository and archives. Data repository tools need to support scientific stewardship and preservation, with the goal of expanding access and improving usability of critical data generated by DOE. Data management methods for storing, versioning, organizing data by citation and tracking data usage, searching and retrieving data via web and programmatic tools, and supporting FAIR, data standards, formats and protocols, is becoming increasingly

critical. There is an opportunity to automate the lifecycle of data in these curated collections starting from data capture, ingestion, annotation, management, consistency checking, and organization of data. Automation while allowing humans-in-the-loop will be key to large-scale creation of FAIR scientific data and require us to address key research challenges in data management.

Data Wrangling. Experimental, observational, and simulation data are often messy and rife with inconsistencies. Today, data wrangling and QA/QC methods are largely manual and require a mix of domain-specific and general techniques. There is a need to develop generalized methods to manage QA/QC on diverse, high-dimensional variable data sets. Automation and scalability of these methods are necessary to support data sets beyond current size and variability. Data wrangling processes require human-in-the-loop interaction. It is important to augment scalable methods with transparent and collaborative interfaces, analogous to software issue trackers, that allow data providers, archive operators, and data consumers to interact with the data wrangling and QA/QC process. AI provides us a way to automate tedious data wrangling tasks by possibly making suggestions to the end-user on data cleaning methods. Future data management methods will need to provide a performant set of pre-HPC data extract-transform-load (ETL) pipelines tailored to allow data to land quickly from scientific sensors, and rapid validation / preparation for HPC analytics. Automating the real-time and near-time processing, curation and QA/QC of data is an essential precursor for improving efficiency while using it for AI at scale.

Metadata and Provenance management. Metadata and provenance management has been considered critical for data management for many years [1]. The emergence of AI workflows makes it even more critical to address these challenges to ensure the data used by the AI models are traceable and accountable and ensure that we can trust the AI models. However, AI also provides an unique opportunity to automate metadata and provenance management that have been largely tedious and human-intensive. Generating metadata using machine learning, deep learning and natural language processing from a number of artifacts including proposals, publications, images has shown promise. Similarly, AI/ML methods can be used to learn and improve the methods for tracking, capturing and correlating provenance information within and beyond a single system. These AI-driven methods can be used to identify new, unknown data sources, instrument them on-the-fly, and collect any missing provenance to make the system trustworthy.

Timeliness. Scientific data is increasingly at risk of being unusable without appropriate data management methods, techniques, and tools. Data wrangling and curation techniques, metadata, and provenance are especially critical as we enter the era of Artificial Intelligence where data generated will be used not only by humans but also complex algorithms.

References

1. *Report of the DOE Workshop on Management, Analysis, and Visualization of Experimental and Observational data – The Convergence of Data and Computing*. United States: N. p., 2016. Web. doi:10.2172/1525145.
2. S. Vazhkudai, *et al*., "Constellation: A science graph network for scalable data and knowledge discovery in extreme-scale scientific collaborations," in *2016 IEEE International Conference on Big Data (Big Data)*, doi: 10.1109/BigData.2016.7840959
3. G. P. Rodrigo, M. Henderson, G. H. Weber, C. Ophus, K. Antypas and L. Ramakrishnan, "ScienceSearch: Enabling Search through Automatic Metadata Generation," *2018 IEEE 14th International Conference on e-Science (e-Science)*, 2018, pp. 93-104, doi: 10.1109/eScience.2018.00025.

Title: Enabling Intelligent Scientific Workflow and Data Management through Provenance Learning

Authors: Lipeng Wan[1], Scott Klasky[1], Ana Gainaru[1], Qing Liu[2], Norbert Podhorszki[1], Greg Eisenhauer[3], Todd Munson[4]

1 Oak Ridge National Laboratory, {wanl, klasky, gainarua, pnorbert}@ornl.gov

2 New Jersey Institute of Technology, {qliu}@njit.edu

3 Georgia Institute of Technology, {eisen}@cc.gatech.edu

4 Argonne National Laboratory, {tmunson}@mcs.anl.gov

Topic: Capturing provenance information/Utilizing AI to improve I/O patterns

As we step into the era of exascale computing, two trends are bringing new challenges to our community. The first trend is that many scientific applications are becoming data-intensive, where the volume and velocities of data produced by the simulations and experiments are growing exponentially. Searching for useful information in these exponentially growing datasets is like finding a needle in a haystack. The second trend is that the data produced by scientific applications is accessed by different scientists for many purposes. Therefore, workflow and data management strategies need to be flexible to adapt to the diverse user requirements as the data is repurposed during its lifecycle.

Challenges:

In order to accelerate scientific discovery under these trends, one *challenge* is to manage the data placement and movement intelligently across the storage hierarchy so that scientists can access their most needed data with low overheads. Particularly, if data can be prefetched and preprocessed for fast access based on predictive guidance before it is requested by the user/application, it can greatly speed up the time spent in post processing the data, allowing a greater percentage of the data to be analyzed. In practice this task is extremely challenging because it requires an accurate predictive model and efficient scheduling system to be built which leads to a few challenges: 1) sufficient provenance of the data and user access patterns needs to be captured and maintained with very low overheads for model training; 2) the model needs to capture the user intentions which can be updated dynamically since the user access patterns might change over time; 3) the data prefetching and preprocessing need to be scheduled intelligently to avoid interference with other user/system activities. In our vision, the efficiency of scientific workflows will be significantly improved if these challenges can be addressed.

Opportunities:

There are abundant research opportunities in addressing these challenges. First, capturing the provenance of scientific workflows has been studied by the community for decades, and many approaches have been proposed [1, 2, 3]. Most of the existing approaches focus on collecting provenance of a certain type or at a certain layer [4]. For example, some tools can record user space information, such as user identities, operations executed, etc., while others focus on collecting system level information, such as file access history, etc. In our vision of building an accurate predictive model to guide the data prefetching operations, we need to capture provenances at different layers and combine them to train the model. For example, the provenances captured at the user space can tell us which user generated or accessed which datasets, what operations the user applied to different datasets, etc. The metadata associated with the self-describing dataset generated by middleware like HDF5 [5], ADIOS [6], etc. allows us to know the detailed semantics of data objects inside each dataset. Tools like Darshan can capture system level provenance that include file access history along with the performance of I/O operations. The

provenances collected from different layers need to be organized in a compact format to mitigate the storage overhead and be findable, accessible, interoperable, and reusable.

Second, the development of AI/ML techniques has advanced significantly over the past decade, which makes building and training an accurate predictive model based on the collected provenance to guide data prefetching and preprocessing possible. Particularly, this model needs to capture the explicit relationships or even reveal the hidden relationships between three major factors involved in typical scientific workflows. There are many cases this model will enable, such as : 1) when a user is running a certain scientific code, the model can predict which operation is likely to be applied to which dataset in the near future; 2) when a certain variable was just accessed, the model can inform us a different variable might be needed by the user soon; 3) when a user accessed a dataset similar to another collaborator who analyzed the data in a similar fashion, the model can infer that the same dataset will likely to be accessed by a collaborator. In practice, there might be hundreds of users, thousands of operations, and millions of datasets involved in the process of scientific discoveries. In order to be as flexible as possible, there is an opportunity to use a graph-structured model to represent these factors and the relationships between them.

Finally, there is an opportunity to build an efficient scheduling system to mitigate the interference between data prefetching/preprocessing and other user/system activities. For example, if the I/O and communication traffic are sharing the same network infrastructure, severe interference might occur, leading to a performance degradation. Therefore, the scheduler needs to find a good tradeoff between the benefit and risks of enabling data prefetching at a certain time. Since the historical I/O performance can also be recorded in the provenance, a performance model can be built to allow the scheduler to calculate the cost-effectiveness of data prefetching and preprocessing.

Timeliness:

The recent developments in AI/ML technologies, especially the innovations in graph neural networks (GNNs) [7], make our vision of intelligent scientific workflow and data management promising. Compared with the regular deep learning techniques, an important advantage GNNs have is they are able to capture the graph structure of data, which emphasize not only the data itself but also the relationship between data. This is particularly useful for capturing the relationships between the user, operation and data involved in scientific workflows.

References:

[1] F. Chirigati, D. Shasha, and J. Freire. ReproZip: using provenance to support computational reproducibility. Workshop on Theory and Practice of Provenance (TAPP), pp. 1–4 (2013)

[2] D. Deutch, Y. Moskovitch, and V. Tannen. A provenance framework for data-dependent process analysis. Proc. VLDB Endow. 7(6), 457–468 (2014)

[3] R.K.L. Ko, and M.A. Will. Progger: an efficient, tamper-evident kernel-space logger for cloud data provenance tracking. IEEE Conference on Cloud Computing (CLOUD), pp. 881–889 (2014)

[4] M. Herschel, R. Diestelkämper, and H. B. Lahmar. A survey on provenance: What for? What form? What from? The VLDB Journal (2017) 26:881-906

[5] https://www.hdfgroup.org/solutions/hdf5/

[6] https://adios2.readthedocs.io/en/latest/

[7] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu. A comprehensive survey on graph neural networks. IEEE Transactions on Neural Networks and Learning Systems, vol. 32, no. 1, pp. 4-24, Jan. 2021

Towards Unified FAIR Metadata Services for Scientific Data

Mai Zheng[†], Runzhou Han[†], Haojun Tang
Iowa State University[†], Lawrence Berkeley National Laboratory

1 Topics

Metadata management infrastructure to support FAIR principles; Storage-system architecture design.

2 Challenges

Data-driven scientific discovery has been well acknowledged as a new fourth paradigm of scientific innovation [4]. As the complexity of data infrastructures rapidly increases, however, the paradigm shift imposes multifaceted challenges for managing scientific data with necessary rigorousness. The traditional HPC storage models centered around parallel file systems (e.g., Lustre) may no longer be enough because many computations today may be conducted on non-HPC infrastructures at scale (e.g., Cloud platforms and Edge sites). Such complexity and diversity makes sharing and reproducing scientific data difficult. Inspired by the FAIR principles [5] (i.e., Findable, Accessible, Interoperable, and Reusable), we identify three major challenges as follows:

Lack of FAIR Metrics. The FAIR principles were proposed as a guide for maintaining digital artifacts with critical properties that are increasingly important at scale (e.g., interoperability [5]). While the concept has been recognized across communities in recent years, the-state-of-the-art FAIR prototypes (e.g., F-UJI FAIRness Evaluator [2], FAIR Maturity Indicator [3]) only cover a limited number of basic metrics (e.g., containing URLs) which is too simple for describing the FAIRness of scientific data. For example, scientists often seek a variety of metadata information from scientific data, including but not limited to the origins of data products, the configurations used for deriving results, the usage patterns of datasets in different granularity (e.g., files, groups, attributes), and so on. Such complex metadata are important for ensuring the scientific rigorousness, but are unfortunately missing in existing FAIR metrics.

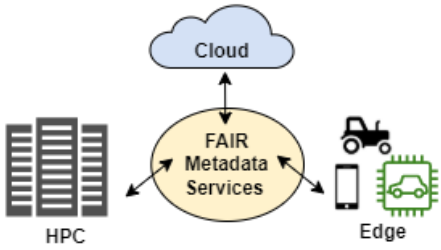


Figure 1: FAIR Metadata Services Bridge HPC and non-HPC Environments by Ensuring FAIR-Compliance of Scientifc Data.

Lack of In-Depth Validation. The existing FAIR evaluation typically parses the description of a dataset and checks if the description contains expected information or follow a pre-defined format (e.g., a valid URL or DOI format). While such text-based format checking is helpful, it is relatively shallow because it does not evaluate the findability or accessibility of the data in depth (i.e., whether it is truly retrievable from the URLs or the DOI.ORG). Moreover, it is impossible for existing solutions to determine whether a dataset is actually interoperable or reusable because there is no any execution involved in the evaluation process.

Lack of System Support for Generating FAIR-Compliant Metadata. Most fundamentally, existing solutions only consider measuring the FAIRness *after* a dataset has been generated, which is often too late to achieve the complete FAIRness. Many of the critical metadata that are necessary to ensure the FAIR compliance (e.g., provenance information for reproducibility) must be collected *during* the process of creating the data products, which requires dedicated system support that is mostly ignored by existing solutions. Without such build-in metadata support (e.g., provenance tracking and querying), it is impossible to achieve full FAIRness for scientific data on modern infrastructures.

3 Opportunities

To address the challenges identified above, a unified framework providing critical FAIR metadata services covering the measurement, validation, and generation of FAIR-compliant data across infrastructures is much needed (Figure 1). Such a framework will likely require cohesive efforts across communities (e.g., HPC, cloud & edge computing, storage systems). We identify a few key opportunities and enablers as follows:

FAIR Awareness & Requirement. As the scale and complexity keeps increasing, more communities are realizing the needs of FAIR. A wide range of stakeholders (e.g., funding agencies, publishers) increasingly require FAIR-compliance on data [5]. Reinforcing such positive trend will likely forster the wide adoption.

Provenance System Support. Data provenance, or data lineage, describes the lifecycle of data. Great efforts have been made to collect provenance at different levels (e.g., databases, operating systems, workflows). While practical provenance support for scientific data at scale is still in its infancy, we expect the advancement will lay a cornerstone for interoperability and reusability.

Containerization & Serverless Computing. Software containerization and the serverless paradigm is pushing the decoupled compute and data storage to the next level, both of which are contributing to the elasticity inherently needed in FAIR.

Open-Source Data Infrastructures. Large-scale data infrastructures (e.g., ARA [6] in Figure 2) will enable deep integration of metadata services in open-source software stack for generating FAIR-compliant data at scale.

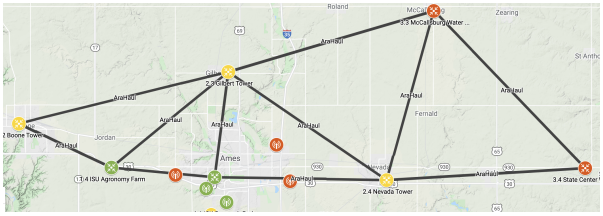


Figure 2: The ARA Infrastructure Spanning Over 37 Miles Enables Experimental FAIR Metadata Services at Scale [1].

4 Why Now

A wide range of technologies along the four dimensions identified above (e.g., GoFAIR initiative, W3C provenance model, Apache OpenWhisk, OpenStack, ARA-like Infrastructures) are growing and flourishing now. Based on such key enablers, the vision of unified FAIR metadata services, if success, will push the scale of scientific discovery as well as the rigorousness of scientific data to a new level.

References

[1] ARA Infrastructure. https://arawireless.org/.

[2] F-UJI FAIR Data Assessment. https://www.fairsfair.eu/f-uji-automated-fair-data-assessment-tool.

[3] FAIR Evaluation Services. https://fairsharing.github.io/FAIR-Evaluator-FrontEnd.

[4] A. J. Hey, S. Tansley, and e. Tolle, Kristin Michele. *The Fourth Paradigm: Data-Intensive Scientific Discovery*, volume 1. Microsoft Research Redmond, WA, 2009.

[5] M. D. Wilkinson, M. Dumontier, S.-A. Sansone, L. O. B. da Silva Santos, M. Prieto, D. Batista, P. McQuilton, T. Kuhn, P. Rocca-Serra, M. Crosas, et al. Evaluating fair maturity through a scalable, automated, community-governed framework. *Nature Publishing Group: Scientific Data*, 6(1):1–12, 2019.

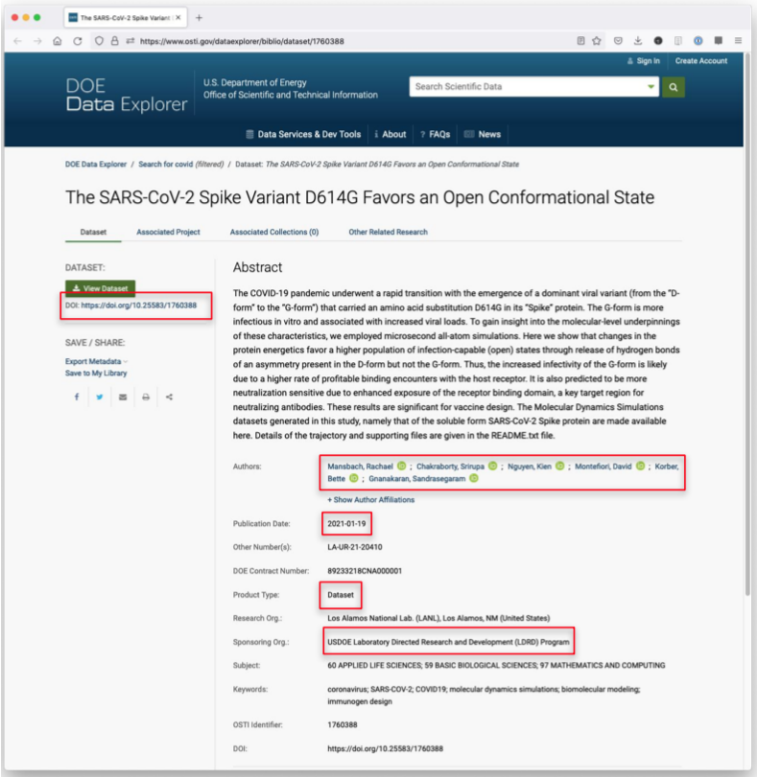
[6] H. Zhang, Y. Guan, A. Kamal, D. Qiao, M. Zheng, A. Arora, O. Boyraz, B. Cox, T. Daniels, M. Darr, et al. ARA: A Wireless Living Lab Vision for Smart and Connected Rural Communities. *Proceedings of ACM WiNTECH, New Orleans, LA, USA*, 2021.

Title: Making Scientific Data Available to Machines

Author: Martin Klein, Los Alamos National Laboratory, mklein@lanl.gov

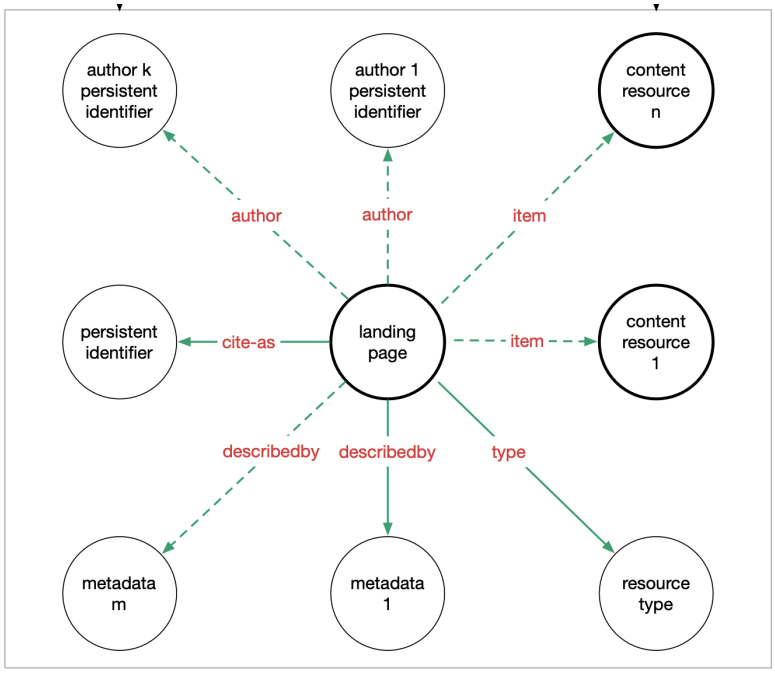
Topic: Metadata management infrastructure to support FAIR principles

Most of today's published scientific data resides in domain-specific or general purpose (institutional) repository systems. These platforms often come with attractive characteristics such as convenient submission procedures, linking between related resources (datasets and source code, for example), and intuitive user interfaces. However, they are still siloed in a way that most of their data and value-added information is merely accessible to humans. The figure below shows a screenshot of a dataset available via OSTI's DOE Data Explorer. This is only one of many examples that highlights valuable information i.e., descriptive metadata and links to content, conveyed on the landing page. We can see the dataset's persistent identifier (in this case a DOI), its authors including their ORCIDs, publication date, type of object (dataset), and even the sponsoring entity of the work. All these data points are trivial for a human to collect and use. For a machine i.e., a web crawler, however, this information is not easily accessible. When encountering the URL of the landing page, a machine has no easy or uniform way to extract the data of interest describing the (in this case) dataset. Too often, the only way is to scrape the page, re-engineer the HTML code, and apply heuristics to somewhat intelligently make decisions on, for example, how to retrieve the authors' ORCIDs. Not only is this process error-prone, it is also time-consuming and usually breaks the moment the user interface is changed. In addition, scraping is typically discouraged by content providers and frequently results in blocked IP addresses of the scraping client. A content aggregator, for example, is therefore left with using the API, specific to each portal and platform. Typically, APIs provide information that seems useful for the hosting entity but often has no anchor in the community actually using the data. So, given the lack of synchronization between APIs, their functionality, and serialization, true system interoperability is very hard to achieve. To support FAIR principles as they are intended, in a machine-actionable and uniform way, we need to find better ways of making scientific content available on the web. More specifically, we need a robot to be able to determine which link among the many on a landing page leads to the content and which leads to metadata.



The principles of the web i.e., links and link relation types can play a role in addressing this problem. In the above example, if a machine resolved the URL of the landing page and, together with its content, was offered self-describing and meaningful links that convey the resource's persistent identifier, what type of content to expect (dataset vs source code, for example), the persistent identifiers of the authors, the location of the metadata records, etc, such links would significantly contribute to making FAIR's Findable, Accessible, and Reusable a reality. The concept based on such machine-actionable links also contributes to FAIR's Interoperable through its uniform approach and because it is entirely based on widely implemented web protocols specified in IETF RFCs. As such, the interoperability that results from adopting it is not restricted to

the scholarly landscape or individual scientific platforms but encompasses the web at large. The below figure schematically shows some of the links that are possible for a landing page to convey. The red labels are currently defined link relation types that may be applicable in this context.



We have seen a lot of acquisitions, mergers, and consolidations in the scientific data market, in large part driven by the “Economies of Scale” principle. As such, we are increasingly losing control over our scientific data, its management, and even use. It is therefore of utmost importance to investigate, test, and deploy working standards-based solutions in support of open science principles that ultimately FAIR is all about. Some of the technology is already available, for example, a number of link relation types to make links meaningful are already well described and standardized but very likely more is needed to encompass the broader scientific landscape.

In addition, the replication crisis in various scientific disciplines is real. From Psychology [1], to Neuroscience [2], to artificial intelligence [3], and, more recently, cancer research [4,5], we observe the same issues with aspects of FAIR. It is on the scientific community to address this crisis, we can not rely on commercial publishers or vendors to solve it for us.

References

[1] Baker, M. Over half of psychology studies fail reproducibility test. *Nature* (2015). https://doi.org/10.1038/nature.2015.18248

[2] Perrin, S. Preclinical research: Make mouse studies work. *Nature* **507,** 423–425 (2014). https://doi.org/10.1038/507423a

[3] Matthew Hutso. Artificial intelligence faces reproducibility crisis. Science, 16 Feb 2018, Vol 359, Issue 6377, pp. 725-726. https://doi.org/10.1126/science.359.6377.725

[4] Begley, C., Ellis, L. Raise standards for preclinical cancer research. *Nature* **483,** 531–533 (2012). https://doi.org/10.1038/483531a

[5] Timothy M Errington, Alexandria Denis, Nicole Perfito, Elizabeth Iorns, Brian A Nosek. Reproducibility in Cancer Biology: Challenges for assessing replicability in preclinical cancer biology. eLife 2021;10:e67995 https://doi.org/10.7554/eLife.67995

Title:
Scientific Data Lifecycle Management Across Storage Services
Authors:
Matt Macduff, Pacific Northwest National Laboratory, matt.macduff@pnnl.gov
Chitra Sivaraman, Pacific Northwest National Laboratory, chitra.sivaraman@pnnl.gov
William I. Gustafson Jr., Pacific Northwest National Laboratory, William.Gustafson@pnnl.gov

Challenge:
With the increased commoditization of storage services there are a lot of choices facing data owners. These services including varying capabilities in locality, availability, reliability, access control, network bandwidth, security, performance, longevity and cost. For large projects or institutions, these considerations are part of designing a dedicated system. For many other efforts, it is neglected. These tradeoffs can easily reduce the impact of the data by placing barriers to use or even existence. If the choice is between expensive and available or cheap and not open and easily available, the result is not impactful. In the case of scientific data, data are derived from a variety of sources and the cost to producing the data can be significant. An expensive field campaign may be irreplaceable and not easily reproducible. A large simulation may be extremely expensive to rerun. These challenges drive the need to keep the data permanently even though the lifecycle of the data may change over time. The data often transitions from active use to a historical reference. It may be replaced by a newer dataset or suddenly be in demand because of a publication. There are a broad variety of intermediate states across the spectrum of data service requirements that can vary with the lifecycle of the data.
Additionally, the stakeholders of the data evolve over time with each having their own interest and preferences in the data service capability. The stakeholders consist of producers, consumers from diverse domains and system managers. The producers typically do not have a long-term engagement with the data lifecycle. They can provide good input into the expected patterns of use during the first phase of the data. With deliberate effort, data producers and owners can predict the long-term lifecycle, but they may be inconsistent in their ability to do this, and unforeseen uses cases can sometimes be the most impactful. The consumers clearly have a vested interest in the data, but it is often 'in the moment' or erratic. They work around and work with whatever is provided. Long-term consumers certainly have insight into the preferred data service capabilities that would help their work. The data managers have the role of managing cost against ambiguous requirements for data services and changing service options.

Opportunities:
Tapping into the shared needs and insights of each group to create a data service framework for long term management is difficult with a variety of storage services combined with a diverse user base.
Developing transparency and shared vocabulary with these groups would enable planning of effective data services. This could include creating of common data service patterns with an understanding of the tradeoffs well documented. As an example, a dataset is initially created and shared with the few key collaborators reviewing it. Data producers would like instant access to the data next to where they are doing their computations and evaluation. When the data is made available to a broader community, the locality changes as well as expectations around access control and reliability. Some datasets may include dynamic features of searching and transformation that include computing capabilities and performance during this phase. As the data continues to mature, a subset may evolve into a robust reference dataset with long-term expectations of instant public availability while other portions are migrated to colder storage. The data still needs to be available to the broader community, but the infrequent access makes longer delivery times acceptable. The data manager may migrate the data to a cheaper storage solution that still meets the reliability and longevity needs of the data while sacrificing some user convenience. To mitigate the often hidden and significant cost of labor to move, manage, access and update the data, tools are needed that clearly and effectively handle the heavy lifting with minimal involvement. Defining policies and processes for managing data is a continuous process and is important to manage the cost of data services while providing research material for those with little or no funding and promote innovation and potential new data uses.

Timeliness:
The increased diversity in storage services and capabilities that are now available and often expected and with associated costs, creates a challenge for everyone involved in data management to reconcile.

General Scalable Cooperative Provenance Capture

Matthew L. Curry, (Sandia National Laboratories), Purushotham V. Bangalore (UA),
Anthony Skjellum (UTC), Csilla Farkas (U. of South Carolina), Farah Kandah (UTC)

1 Introduction

Data provenance is defined as one or more artifacts of metadata that can be used to track changes to data over time and to ensure its integrity. Secure provenance is achieved in a system when the integrity of provenance data can be ensured and the provenance data itself is always available for querying. There are efforts to build provenance systems in HPC already; but, most of them are tied to individual workflows. There are some notable exceptions to such workflow-specific systems, such as SPADE [2], PASSv2 [4], the Lightweight Provenance Service [1], and other language-specific mechanisms (R, Python, etc.) [3]. Because we seek to generate quality provenance that is actionable for its intended *a posteriori* purposes, this whitepaper discusses an approach to general scalable cooperative provenance capture, together with a strategy for integrated provenance management enabled by current and emerging system infrastructure. The integrated framework envisioned here is one that brings together research achievements from the database, security, and HPC communities to support secure collection, maintenance, and sharing of scientific workflow provenance data.

2 Challenges

We identify these key challenges: (1) siloing of provenance; (2) issues with diverse storage system interfaces on a given system, (3) cross-system provenance correlation, and (4) provenance trustworthiness.

HPC storage systems have not supported provenance well. While individual user communities have embraced provenance as a means of ensuring reproducibility, reducing error, and understanding results, many application areas are not well equipped to leverage provenance in their workflows. This creates silos of provenance information that are often user-managed along with the output data itself, leading to similar data management issues. Such data is only useful for those who know how to find it, or are careful enough to ensure it is preserved properly.

While general provenance tracking for HPC has been demonstrated, the preponderance of storage system interfaces available presents a key challenge. It is now common to find traditional parallel file systems alongside system-provided object stores, key-value stores, ephemeral stores (like burst buffers or BeeOND [6]), or glue components that facilitate cross-component data exchange (like Faodel [5]). Kernel-level provenance systems that leverage Linux kernel events will be unable to see interactions with many of the previously listed storage systems; and, if data is able to be collected for these other interfaces, correlating the provenance will be difficult.

Similar issues exist for provenance correlation between different provenance systems. As simulations become more complex, disparate components are tied into workflows that may not have compatible representations or query interfaces. Data spanning use from one provenance system to another may not have a useful link, requiring manual annotations or inspection of project artifacts to ascertain provenance. Such methods are more error prone and less useful for automatic processes.

Finally, the provenance captured must be trustworthy in terms of its integrity, completeness, and availability. Secure provenance is required to ensure accountability and support non-repudiation. Potentially, certain provenance data must also be subject to access controls.

3 Opportunities

The above challenges can be addressed by focusing on flexible provenance systems that can operate at different levels of specificity to the workload, thereby allowing rational trade-offs of overhead vs. specificity (e.g., system-wide, automatic, and efficient provenance capture, paired with domain-specific systems that are able to provide

more information that cannot be inferred). Such cooperative behavior has the potential to increase assurance in provenance through implicit, secure mechanisms and unified management while allowing application-level or workflow-level information to increase the value of the provenance.

Designing such systems will be challenging. Provenance systems are not necessarily compatible, so an underlying common system requires attention to both ingestion and queries. However, users will be able to perform powerful operations that can span provenance crossing different workflow engines and provenance frameworks. Further, since the provenance is more integrated through unifying management and query mechanisms, users of provenance can extend from developers through to system administrators and architects who want to understand how systems are used.

By extending common provenance gathering from specialized workflow tools to general system use, new ways of leveraging provenance can be applied. For example, it becomes possible to understand what data sets are most valuable within a system, based on the number of derivative products and their impacts. Such queries might be performed at the behest of a program manager, and be used to understand how investments can be made for maximum benefit.

The big impact of this approach is a design and strategy that takes the goals of future provenance analyses (queries) as input requirements for the integrated management of the provenance, models for correlating provenance across systems and within a system, as well as rationale for acceptable levels of overhead incurred during collection for a given workflow or use-case.

4 Timeliness/Maturity

Recent developments in operating system and storage technology have made high performance integrated provenance more feasible:

- Elastic system infrastructure such as Kubernetes allows for the ability to scale unexpected heavy workloads into the compute section of an HPC system, removing the difficulty of forecasting provenance load.
- High-bandwidth, low-latency storage technologies (like NVMe, accessible over RDMA, and persistent memory) can benefit the analytic workloads required by provenance stores.
- High-performance analytics support for parallel file systems (like SkyhookDM via Ceph and Apache Arrow via its efficient columnar store implementations for distributed storage) provide similar capabilities for hosting and processing large provenance data on parallel file systems.

While hardware developments are important, the push for reproducibility in HPC is driving the need for pervasive provenance at many HPC sites. Users are now more willing to use provenance facilities, especially if they don't come with a large developer burden or efficiency reduction.

References

[1] Dong Dai, Yong Chen, Philip Carns, John Jenkins, and Robert Ross. Lightweight provenance service for high-performance computing. In *2017 26th International Conference on Parallel Architectures and Compilation Techniques (PACT)*, pages 117–129, 2017.

[2] Ashish Gehani, Raza Ahmad, Hassan Irshad, Jianqiao Zhu, and Jignesh Patel. Digging into big provenance (with SPADE): A user interface for querying provenance. *ACM Queue*, 19(3):77–106, 2021.

[3] Jingmei Hu, Jiwon Joung, Maia Jacobs, Krzysztof Z. Gajos, and Margo I. Seltzer. Improving data scientist efficiency with provenance. In *Proceedings of the ACM/IEEE 42nd International Conference on Software Engineering*, ICSE '20, page 1086–1097, New York, NY, USA, 2020. Association for Computing Machinery.

[4] Kiran-Kumar Muniswamy-Reddy, Uri Braun, David A. Holland, Peter Macko, Diana Maclean, Daniel Margo, Margo Seltzer, and Robin Smogor. Layering in provenance systems. In *Proceedings of the 2009 Conference on USENIX Annual Technical Conference*, USENIX'09, page 10, USA, 2009. USENIX Association.

[5] Craig Ulmer, Shyamali Mukherjee, Gary Templet, Scott Levy, Jay Lofstead, Patrick Widener, Todd Kordenbrock, and Margaret Lawson. Faodel: Data management for next-generation application workflows. In *Proceedings of the 9th Workshop on Scientific Cloud Computing*, ScienceCloud'18, NY, USA, 2018.

[6] BeeGFS Wiki. BeeOND: BeeGFS On Demand. https://www.beegfs.io/wiki/BeeOND.

End-to-end Scientific Data Provenance: Challenges and Opportunities

Swen Boehm, Michael J. Brim, and Olga A. Kuchar, ORNL, {boehms, brimmj, kucharoa}@ornl.gov

Topic: Provenance capture, curation, and sharing for scientific data

Challenge:

Data provenance is information that describes “the entities, activities, and people involved in producing a piece of data” [3]. Scientific data takes many forms including but not limited to scientific modeling and simulation input or output datasets, data analysis and visualization products, databases containing domain-specific data populations, and scientific publications and presentations. Prior DOE workshops and funding calls have clearly stated the need for capture and curation of scientific data provenance to improve data usability, scientific integrity, and scientific reproducibility [1,2]. Current research directions that explore the potential to enhance DOE science using artificial intelligence and machine learning, where many approaches rely heavily on a large quantity of quality data, are only magnifying the lack of associated technologies and solutions that meet this need.

Capturing end-to-end provenance information across a diverse range of scientific domains and environments poses many challenges. Provenance information has many potential uses (e.g., scientific reproducibility, data veracity, and data lineage), and each use may have different requirements for the information that needs to be captured and its level of detail. For example, reproducibility may benefit from having detailed provenance information for each of the applications and methods used in the production of data, while data lineage may only require high-level descriptions of the data transformations used by those applications and methods. Ideally, we could capture provenance information for all data in enough detail to cover the superset of all use case requirements, but practically that may be intractable due to storage or process overhead constraints that could potentially create artificial bottlenecks in scientific workflows. Furthermore, many workflow solutions exist that collect at least some provenance data for their respective workflows. However, collection is often limited to the activities occurring within the workflow system, which makes it difficult to collect end-to-end provenance information for data that is used or produced outside of a particular workflow solution.

Storage of provenance information poses a challenge as well. Recording the provenance for all data products may significantly increase the amount of metadata that needs to be stored. Short-lived or intermediate data products could further exacerbate this issue. Provenance information also needs to be accessible to everyone who has access to the data. One approach would be to add the information to the data product, which may require changes to widely-used data schemas that introduce undesirable incompatibilities with existing data consumers. Alternatively, and more promising, would be the use of linked data records that are stored separately from the data products. Another level of complexity is added if a data product is derived from a database or created with (complex) queries. Depending on the type of query and of the data this might require storing of intermediate data in addition to the query data. And finally, and maybe most challenging is ensuring the integrity and verifiability of the provenance records. To trust the data and its provenance, it needs to be verifiable that the provenance record has not been altered and indeed belongs to the data product.

Opportunity:

While there are standard methods for recording provenance information (e.g., PROV-O [4]), there are no established standards on how to capture and curate provenance data to enhance data usability and sharing. The breadth of science domains within the DOE mission portfolio provides a unique

opportunity to address this problem through co-design and research towards a provenance capture, curation, and sharing standard that enables provenance tracking across different solutions and environments, based on guidance from domain scientists, information scientists and computer scientists. Such a standard is the first step toward design and deployment of technologies that enable end-to-end scientific data provenance. Existing workflow systems are obvious targets for deploying the enhanced technologies in an automated and minimally intrusive manner, but many scientific workflows still include manual data tasks, such as records in physical lab notebooks or entries in spreadsheets. These manual processes cannot be ignored and require provenance tracking solutions that do not add significant overhead for the scientist. Additionally, the new provenance technologies should integrate with and leverage existing scalable data and metadata management solutions.

Timeliness or maturity:

There exists a suite of upcoming Web3.0 tools well-suited for data provenance. Blockchain is arguably data provenance at its core. Sequences of transactions are linked together, distributed for redundancy and security, and become an immutable and verifiable resource. Similarly, smart contracts such as on the Ethereum blockchain offer a means of publishing usable code by its users. Such an approach provides a verifiable means in which to publish standard, taxonomies, and distribute programming rule sets around standards that may be pulled and compiled into client-side tools. Thus, a clear, immutable lineage may be established and encourage building off of the work of others, such as extending or incorporating scientific smart contracts. Concerns about the energy requirements of blockchains can be addressed by utilizing Proof of Stake based blockchains or DAG (Directed Acyclic Graph) based distributed ledgers [5]. Another potentially useful technique involves the use of verifiable data structures as described in [6].

Other timely aspects include software management solutions that naturally provide useful provenance information. Open source software development practices including public version-controlled code repositories (e.g., GitHub) provide useful identifiers (i.e., commit hashes) to easily and uniquely identify specific software code versions. Containerized applications and technologies like Spack enable unique identification of entire software stacks used in the creation chain of a data product. Runtime introspection to capture the execution environment, such as the modules loaded within an HPC batch job, can be combined with software provenance records to enhance reproducibility of experiments. Another opportunity the provenance records for the software stack provide is tracing all the data products created with an application or library that contained a bug and introduced errors.

References:

1. “Scientific Discovery at the Exascale: Report from the DOE ASCR 2011 Workshop on Exascale Data Management, Analysis and Visualization”, Online: http://science.energy.gov/~/media/ascr/pdf/program-documents/docs/Exascale-ASCR- Analysis.pdf
2. “Synergistic Challenges in Data-Intensive Science and Exascale Computing, DOE ASCAC Data Subcommittee Report”, Online: http://science.energy.gov/~/media/ascr/ascac/pdf/reports/2013/ASCAC_Data_Intensive_ Computing_report_final.pdf
3. “An Overview of the PROV Family of Documents”, Online: http://www.w3.org/TR/prov-overview/
4. “PROV-O: The PROV Ontology”, Online: http://www.w3.org/TR/prov-o/
5. P. Ferraro, C. King and R. Shorten, "Distributed Ledger Technology for Smart Cities, the Sharing Economy, and Social Compliance," in *IEEE Access*, vol. 6, pp. 62728-62746, 2018.
6. “Verifiable Data Structures”, Online: https://github.com/google/trillian/blob/master/docs/papers/VerifiableDataStructures.pdf

HPCFAIR:
An Infrastructure for FAIR AI and Scientific Datasets for HPC Applications

Murali Emani[1], Argonne National Laboratory
Chunhua Liao, Lawrence Livermore National Laboratory
Xipeng Shen, North Carolina State University

Introduction: While Machine Learning (ML) and Artificial Intelligence (AI) has disrupted every computing industry, the challenges in quickly accessing, reproducing the results, or reusing the research components have become overwhelming for researchers. The massive data produced by research communities such as experimental datasets, AI models constitute a rich repository of *artifacts*. Implementing sound data management principles is the need of the hour to leverage the rich repositories. Future of Research Communications and e-Scholarship (FORCE11) [1] defined the four foundation pillars, namely **FAIR**, that stands for data artifacts to be *Findable*, *Accessible*, *Interoperable* and *Reproducible*. It dictates the publication of scientific datasets and AI/ML models and associated research components making them adhere to FAIR principles. The advantages are manifold in that it helps end-users such as domain scientists or application developers to adopt and easily integrate data artifacts into their applications for reuse. This significantly cuts down the application time development and support reproducing their experiments. The emergence of the frameworks' development to address these challenges demonstrates a conspicuous necessity for applying FAIR [2] data guiding principles driving better scientific data management and stewardship.

The HPCFAIR Framework: Adhering to the previously explained FAIR principles, we developed a framework, HPCFAIR [3], to assist the high performance computing and science communities comprehend the relationship between models, datasets, and workflows with a high-level ontology. The overarching goal of this framework is to implement FAIR principles for ML-driven HPC. Here, we have concisely summed the FAIR data principles for data objects that include AI models, datasets and associated workflow components.

- *Findable* (F): F1. Data objects (defined by R1 below) are described with rich metadata.
  F2. Metadata clearly and explicitly include the identifier of the data objects it describes.
  F3. Enable mechanism to find data objects AI models by rich associated metadata.

- *Accessible* (A) A1. Data objects stored are retrievable by their unique identifier with persistent metadata.
  A2. Communication protocol to retrieve data objects is open, free, and universally implementable.
  A3. Access to data objects requires authentication and authorization, where necessary.

- *Interoperable* (I) I1. Data objects use a formal, accessible, and shared language for information description.
  I2. Data objects are interoperable from one format to another.
  I3. Data objects include qualified references to other data objects.

- *Reproducible* (R) R1. Metadata (of the data object) is extensively described with high fidelity.
  R2. Data objects are served with a public and accessible data usage license.
  R3. metadata adheres to domain-relevant community requirements.

HPCFAIR has several components including a front end, metadata, containerized images storing models, and a Python library for managing models and datasets. All metadata is provided by a supportive ontology

[1]Corresponding author: memani@anl.gov Funded in part by the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357. Prepared by LLNL under Contract DE-AC52-07NA27344 (LLNL-ABS-827807) and supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Program under Award Number DE-SC0021293.

[4]. We are also working on advanced component to automatically synthesize workflows. The components of HPCFAIR communicate with each others using a set of APIs. With the APIs provisioned, users can query for datasets and models trained for a specific task, with metadata. These can be deployed in their applications and run them without the need to worry about the software support. If the data objects are not available, the user can save them in a central repository for future reuse. Designing as a three-tier architecture will enable us to implement each component as an independent module with minimal dependencies and is easily extensible to the other language APIs. The indexed metadata allows users to search for the required data objects based on tags or keywords. We store the metadata in the JSON-LD format to ensure that it can be accessed via open and standard communication protocols like API calls. We currently support access to public data objects and present steps to access any behind the login data objects. Similarly, while loading any data object, we check for its existence in the cache. In such scenarios, we provide users with either a reuse option or newly force-load the data object. We aim to incorporate authentication checks to ensure that access is granted to only authorized users.

HPCFAIR empowers researchers to explore the research methodologies, metrics databases, varying datasets, and novel learning techniques. Notwithstanding the proposed framework's capability to support the generic ML use cases, we primarily focus on tailoring it to suit the large-scale HPC workload.

Research Opportunities: The experience of developing HPCFAIR has helped us identify new research opportunities to facilitate provenance and metadata management infrastructure FAIR AI.

First of all, metadata is essential to improve data provenance and trustworthiness. While ontologies with controlled vocabularies and properties provide the required metadata, developing ontology-based metadata is still a tedious, repetitive manual process for different domains. Even with available ontologies, annotating real world application data with ontology concepts is another bottleneck. There is an urgent need to develop techniques to automatically generate and update ontology-based metadata for any scientific domains and subsequently annotating large amount of data objects.

Next, while there are efforts to have unified formats for AI models (e.g. ONNX), more research and development efforts are needed to standardize APIs and data formats involving fast-evolving machine learning techniques. Yet another limitation is that existing APIs to access data objects do not have sufficient support to collect and manage rich metadata for scientific data objects. There is a need to design APIs and the associated data formats with enhanced features. It requires an emphasis on costs and benefits analysis of scientific data management.

Based on the FAIR data principles, researchers have developed different ways to quantitatively and even automatically evaluate the level of FAIRness of a given dataset. We believe that the same approach can be adopted for scientific datasets. The community should expand the FAIR data principles to incorporate more guidelines for to enable this feature. We should also invest in both qualitative and quantitative metrics and automated evaluation processes and tools to improve data provenance, and other favorable properties.

References

[1] The FAIR Data Principles. `https://www.force11.org/group/fairgroup/fairprinciples`.

[2] Mark D Wilkinson, Susanna-Assunta Sansone, Erik Schultes, Peter Doorn, Luiz Olavo Bonino da Silva Santos, and Michel Dumontier. A design framework and exemplar metrics for fairness. *Scientific data*, 5(1):1–4, 2018.

[3] Gaurav Verma, Murali Emani, Chunhua Liao, Pei-Hung Lin, Tristan Lucas Vanderbruggen, Xipeng Shen, and Barbara Chapman. HPCFAIR: Enabling FAIR AI for HPC Applications. In *2021 IEEE Workshop on Machine Learning in High Performance Computing Environments (MLHPC)*, 2021.

[4] Chunhua Liao, Pei-Hung Lin, Gaurav Verma, Tristan Lucas Vanderbruggen, Murali Emani, Zifan Nan, and Xipeng Shen. HPC Ontology: Towards a Unified Ontology for Managing Training Datasets and AI Models for High-Performance Computing. In *2021 IEEE Workshop on Machine Learning in High Performance Computing Environments (MLHPC)*, 2021.

Data-centric Abstractions and Adaptation to Enable Distributed Scientific Exploration

Nathan Tallent (tallent@pnnl.gov), Pacific Northwest National Lab
Josh Suetterlein, Oceane Bel, Ryan Friese, Burcu Mutlu

December 15, 2021

Topics. Interfaces for accessing data; Capturing provenance information; Utilizing AI to improve I/O patterns

Challenge

Scientific exploration is hindered by storage rates and volumes. Scientific exploration is increasingly distributed and data-intensive. The answers to many questions involve composing applications with different characteristics, e.g., numerical solvers, data analytics, and machine learning. To focus on exploration rather than application development, domain scientists emphasize productivity and flexibility. The result is *workflows*, or loose compositions of different applications or tasks. Each application potentially uses a different programming model. Data sources are distributed. As a result, distributed I/O is often the de facto method for task communication or composition. Although easy to program, this composition faces bottlenecks from storage rates and data volumes [4, 5].

Shifting workloads means online performance adaption is important. Online performance adaption has significant potential. Changing workloads cause dynamics in data and storage flows [1, 3]. Further, large-scale workflows often have latent inter-task (intra-workflow) locality that, if exploited, would reduce data rates and volumes. However, effecting performance adaption faces challenges in monitoring, modeling, and diagnosis. Of particular note, AI/ML can learn and detect complex patterns within workload execution, but faces the challenges in data volume, data variance, training cost, feature selection, and model selection [2].

Emerging architectures expose severe limitations in storage interfaces. Emerging storage architectures blur memory and storage — and even compute. First, memory systems are increasingly likely to consist of hybrid technologies, potentially with separate address spaces specialized for 'low' latency and high capacity. For example, a big-data system today may contain both 'fast' DRAM and ≈8x more byte-addressable persistent memory. Second, byte-addressable persistent memory could consist of even larger 'nearby' pools. With the new Compute Express Link (CXL) standard, it will be possible to directly attach nodes to very large, coherent, pools of persistent memory. Third, near-data accelerators will enable near-data computing. High-performance storage systems already include accelerators, bringing the potential for application-directed near-data exploitation The CXL standard also supports accelerators within memory pools.

Today's storage interfaces contain implicit assumptions about storage architecture that fail to account for this flexibility and will therefore poorly utilize emerging storage architectures. Examples of deprecated assumptions are block access, limited parallelism, simple consistency models, and 'dumb' devices (read/write only).

Opportunity

Data-oriented workflow abstractions. To accomplish goals within domain science, it is important to avoid bottlenecks from storage rates and volumes — as well as subsequent data movement through networks and memory. To avoid such bottlenecks, workflows should incorporate designs that elevate data flow and volume to first-class concerns. For example, the Map-Reduce pattern enabled near-data tasks. However, this one pattern was insufficient: it assumed a flat data space and SIMD-style parallelism that could not account for changing tasks and parallelism based on data. Some examples of needed research areas include:

- What are plausible programming abstractions for avoiding data access bottlenecks, ranging from best practices, state-of-the-art, to visionary, that generalize domain needs with respect to domain programming patterns?
- What are common algorithmic patterns for data manipulation and transformation? How can these patterns be decomposed and recomposed to capture locality, avoid unnecessary data movement, and exploit performance differences between devices? How can such recomposed tasks execute near data?
- What are productive but sufficiently flexible methods for expressing composition options between abstraction components so as to achieve performance?

Co-design of provenance and AI/ML modeling for adaptation. Workloads have important dynamics, ranging from shifts in task characteristics to latent inter-task locality. With AI/ML, it is now possible to learn how to detect complex patterns that evade most heuristics and human inspection, and use them to improve future workloads. These include patterns within code (static patterns) as well as execution (dynamic patterns). However, the practicality of AI/ML (training time, resources, data volume) is heavily dependent on carefully designed provenance and observation. Consequently, there are opportunities to explore:

- What provenance techniques and learning methods can capture patterns within distributed workflows without an enormous volume of training data? How can it avoid frequent model training and re-training?
- Can inclusion of static characteristics help minimize noise in training data without removing useful fluctuations? How do we determine a threshold to differentiate noise from data?
- What model architectures and feature selection techniques can be used to manage features with a diversity of semantics? Can we determine a set of model architectures that will best suit models that change over time?
- What methods can capture the benefits of federated learning (improved data gathering) while avoiding accuracy penalties from reducing global model update frequency?
- How can we manage model obsolescence as workloads change? Can we determine a suitable schedule to replace or update models in response to workload changes?

Interfaces for data vs. devices. Today's programs use entirely different interfaces and execution assumptions for persistent vs. volatile data, block vs. byte accesses, and compute-centric near-data execution. For more effective science, can programs be written against logical data sets that consist of logical data objects? Programs could then use "late binding" for (a) accessing data objects from persistent data stores/memory or volatile fast (e.g., `open` and `read` (block) vs. memory `load`); (b) near-data computing by selecting and moving tasks to smart devices. Late binding rules could be either automatically inferred by analysis or supplied by programmer rules/hints, where the former likely is only effective in restricted cases. To ameliorate the common bottleneck of loading large data sets, data-execution rules could maximize data movement parallelism between persistent and volatile stores and potentially overlap reads with compute.

- How can data-centric interfaces be lazily mapped to devices to resolve volatile/persistent, block/byte, near-data computing?
- How can data-centric interfaces use abstract data locality and parallelism to exploit tiers and parallelism in storage architectures?
- How can tasks be moved to smart devices so as to minimize data movement?

Timeliness

These questions are timely because scientific exploration is increasingly distributed and data-intensive because it combines sensor data, numerical solvers, data analytics, and machine learning. This combination requires coordination between experimental facilities (with specialized instruments), computing facilities with supercomputers, and possibly even cloud resources. Further, emerging systems blur memory and storage; and they will increasingly have the potential for near-data computing by co-locating compute capability near storage and memory. It will therefore become routine to exploit near-data computing.

References

[1] O. Bel, K. Chang, N. R. Tallent, D. Duellmann, E. L. Miller, F. Nawab, and D. D. E. Long. Geomancy: Automated performance enhancement through data layout optimization. In *36th Intl. Conf. on Massive Storage Systems and Technology*, Oct. 2020.

[2] O. Bel, S. Mukhopadhyay, N. R. Tallent, F. Nawab, and D. Long. WinnowML: Feature selection for maximizing prediction accuracy of time-based system modeling. In *The Fifth IEEE Intl. Workshop on Benchmarking, Performance Tuning and Optimization for Big Data Applications*, Dec 2021.

[3] O. Bel, J. Pata, J.-R. Vlimant, N. Tallent, J. Balcas, and M. Spiropulu. Diolkos: Improving ethernet throughput through dynamic port selection. In *18th ACM International Conference on Computing Frontiers*, New York, NY, USA, May 2021. ACM.

[4] R. D. Friese, B. O. Mutlu, N. R. Tallent, J. Suetterlein, and J. Strube. Effectively using remote I/O for work composition in distributed workflows. In *Proc. of the 2020 IEEE Intl. Conf. on Big Data*. IEEE Computer Society, December 2020.

[5] J. Suetterlein, R. D. Friese, N. R. Tallent, and M. Schram. TAZeR: Hiding the cost of remote I/O in distributed scientific workflows. In *Proc. of the 2019 IEEE Intl. Conf. on Big Data*, pages 383–394. IEEE Computer Society, Dec. 2019.

Data Management and Storage over Programmable Networks

Nik Sultana, *Illinois Tech*
James B. Kowalkowski, *Fermilab*
Michael H. L. S. Wang, *Fermilab*
Marc F. Paterno, *Fermilab*

Topic: We lack techniques to exploit the unification of Communication and Computation.

Challenge

Large experiment and measurement complexes—such as particle accelerators, antenna arrays, and underground detectors—present a cluster of data management and storage challenges that delay the "time to physics":

- Large volumes of data are generated and must be transported, stored, and processed. Typically, petabytes of data are transported over multi-gigabit links to—and within—compute clusters [4].
- Similar queries or processing by different project participants on stored data results in duplication of transport, computation, and storage effort across different sites [3].
- There is a latency and bandwidth gap between devices used for acquisition, communication, processing and storage of data—such as how network bandwidth has been outstripping what a CPU can process—which makes it challenging to integrate these devices into one large machine.

This complexity is mitigated in how HPC clusters are designed and operated. For example:

- When processing large volumes of data it could be better to move code to the data, rather than move data towards the code that processes it. This so-called *data centric* approach minimizes data movement and prefers in-place processing of data, but how to write general programs that can operate in this manner is not well understood.
- Various Software-Defined Networking (SDN) controllers exist to coordinate and schedule communication resources at a fine granularity. But currently this coordination relies on a logically centralized view of the network. Round-trip times (RTTs) grow with the size of a network and, coupled with centralized network management, this impedes rapid reaction to congestion or failures in large networks.
- CPU-based data processing is sometimes replaced—and often complemented—by processing on other devices such as GPUs, FPGAs and custom ASICs. But we lack general and high-fidelity techniques to reason about heterogeneous and distributed processing resources.

Opportunity

Linking these challenges and mitigations together is the network connecting the data generation, processing and storage technologies, and the infrastructure used to manage that network.

We are starting to see the proliferation of *programmable* commodity network interconnects that are unifying Computation and Communication, and this unification creates new opportunities for data management and storage through *in-network computing* [5]. Among other uses, this computing paradigm is being explored for fault mitigation and distributed and heterogeneous in-network programming [6]. There is also an opportunity to extend the features sets of HPC interconnects to incorporate additional programmability to better serve frequently-occurring workloads.

Programmable network interconnects diffuse a variety of processing devices across the fabric—on both switches and network cards—to which logic can be offloaded. Logic running on these devices is interwoven with communication to help manage the network (e.g., load balancing or fault tolerance) or opportunistically carrying out key computation+communication tasks within large-scale data processing, such as data concentration [2] and caching [3]. Other potential benefits from using in-network computing for data management and storage include: **(1)** in-network coordination to lower the RTT by making local decisions without involving a centralized SDN controller; **(2)** data-centric computing by moving code to programmable network switches or NICs; **(3)** provenance

tracking across processing stages by using custom network headers; **(4)** slicing of network resources by generalizing VXLANs; **(5)** in-network scheduling for interconnect resources: for example, to preempt high-priority tasks over delay-tolerant background tasks that are using some of a project's infrastructure; and **(6)** automating and thus simplifying checkpointing of long-running codes.

A community wide effort is needed to realize this opportunity however since we currently lack adequate models to **(i)** author and distribute programs across programmable interconnects (perhaps by borrowing tried-and-tested ideas from distributed OSs) and **(ii)** reason about the integration of distributed and heterogeneous resources along these interconnects. This effort will likely involve multi-disciplinary projects to develop non-Von Neumann-style programming [1] paradigms that can better utilize programmable interconnects.

Timeliness or maturity

The opportunity to develop techniques for data management and storage by leveraging programmable network hardware is helped by the following:

- Industry has been moving in this direction with commoditized hardware.
- The toolchains to use this hardware have matured in tandem with the hardware's take-up.
- Different communities have formed to take up and adapt this hardware for different settings—including telecoms, cell networks, and datacenter networks—and improve key features—such as security and programmability.
- There is a growing skill pool formed by graduates from universities that feature programmable network hardware in their network curricula, and from employees of companies that currently use or experiment with this hardware.
- There is an opportunity for cross-community fertilisation of techniques between network fabrics and other programmable fabrics such as those found on FPGAs and Systems-on-Chip (SoC).

References

[1] John Backus. Can Programming Be Liberated from the von Neumann Style? A Functional Style and Its Algebra of Programs. *Commun. ACM*, 21(8):613–641, aug 1978.

[2] Tomasz Bawej, Ulf Behrens, James Branson, Olivier Chaze, Sergio Cittolin, Georgiana Lavinia Darlea, Christian Deldicque, Marc Dobson, Aymeric Dupont, Samim Erhan, Andrew Forrest, Dominique Gigi, Frank Glege, Guillelmo Gomez-Ceballos, Robert Gomez-Reino, Jeroen Hegeman, Andre Holzner, Lorenzo Masetti, Frans Meijers, Emilio Meschi, Remigius K. Mommsen, Srecko Morovic, Carlos Nunez-Barranco-Fernandez, Vivian O'Dell, Luciano Orsini, Christoph Paus, Andrea Petrucci, Marco Pieri, Attila Racz, Hannes Sakul, Christoph Schwick, Benjamin Stieger, Konstanty Sumorok, Jan Veverka, and Petr Zejdl. The new CMS DAQ system for run-2 of the LHC. *IEEE Transactions on Nuclear Science*, 62(3), 5 2015.

[3] Shaun Nichols. In-Network Caching Shown to Enhance Science Community's Access to Experimental Data, Sep 2021. ESNet.

[4] Pete Clarke. DUNE Computing, Jan 2020. LHCOPN/LHCONE Workshop, CERN.

[5] Dan R. K. Ports and Jacob Nelson. When Should The Network Be The Computer? In *Proceedings of the Workshop on Hot Topics in Operating Systems*, HotOS '19, page 209–215, New York, NY, USA, 2019. Association for Computing Machinery.

[6] Nik Sultana, John Sonchack, Hans Giesen, Isaac Pedisich, Zhaoyang Han, Nishanth Shyamkumar, Shivani Burad, André DeHon, and Boon Thau Loo. Flightplan: Dataplane Disaggregation and Placement for P4 Programs. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pages 571–592. USENIX Association, April 2021.

Title: Challenges and opportunities in utilizing AI to optimize I/O and storage

Authors: N. Podhorszki, L. Wan, A. Gainaru, B. Xie, S. Oral, K. Mehta, S. Klasky (ORNL)

Topic: Utilizing AI to learn I/O patterns of emerging workloads for efficient data management

Challenges:

One of the main challenges which applications face as they scale (in/out) is minimizing the time spent in reading and writing data. This challenge continues to grow as the use of AI has further integrated into scientific workflows. This is due to the fact that much more data is being used to train a model to be used in the scientific process, which ultimately means that more data has to be written. One solution that the community has used to meet this challenge is to asynchronously perform I/O in concert with the application. Asynchronous I/O has become a promising approach to improve the user-perceived I/O performance since the data can be temporarily buffered in the DRAM or stored in the non-volatile memory before being written to the much slower storage tiers. However, since the inter-node communication and I/O traffic share the same network, using asynchronous I/O blindly without an intelligent scheduling strategy may cause significant interference with the application itself, potentially slowing down the application compared to synchronous I/O.

From the perspective of shared HPC storage systems (sharing disks, servers or network), the total I/O bandwidth is always limited and shared among concurrently running applications, thus the I/O interference between different running applications are also common and effects asynchronous I/O as much as synchronous I/O. Due to the complexity of applications' I/O patterns, efficiently and intelligently coordinating the I/O operations from a collection of running applications to reduce the overall overhead caused by I/O interference and contention is extremely challenging. Since admins are not allowed to give a user full access to all I/O traffic on a large system, a user level application cannot make decisions on its own. Building a system-wide model that advises applications by giving predictions when it is best to perform I/O operations is a feasible middle ground to utilize system-wide knowledge.

Thus, the *main challenge* is to perform both a local optimization of I/O for large-scale applications while giving the best global optimization. We envision that AI models will be built to best guide the applications and the overall performance of the filesystem. Part of the challenge is in the ability to use the memory/storage hierarchy in order to cache data on the faster tiers before draining this to the slower/larger storage tiers. The local model will need to understand how to avoid interference from internal communication of a code, so that the I/O to slower tiers is coordinated with the communication within the application. Furthermore, the global model needs to be created so that it can predict the overall I/O on the system to best optimize the writing to storage tiers. Therefore, this problem presents itself as both a global and local optimization problem.

Ultimately our goal is to both optimize the overall system I/O performance and use advanced scheduling techniques to optimize the application time. This involves local and global optimization and requires advanced scheduling techniques to coordinate the guidance from these competing models.

Opportunities:

AI methods are being used today for a large variety of tasks, including optimizing the resource scheduling and management for HPC batch jobs [1] and scientific workflows [2] and to detect write bottlenecks on supercomputer I/O systems [3]. Beyond the active use of AI/ML in HPC systems, across science domains and HPC platforms, scientific applications often perform I/O based on predefined and preconfigured I/O patterns and manage a massive amount of data using I/O middleware libraries, such as ADIOS[4] and HDF5 (https://www.hdfgroup.org/solutions/hdf5/). Given the predictable I/O patterns of scientific codes and built on the tunability of the HPC I/O middleware stack, it is possible for scientists and system administrators working at HPC facilities to leverage AI models of these patterns in order to optimize I/O performance through efficient configurations or to minimize system level congestion. The AI models have shown effectiveness when utilized to analyze application's local I/O performance [3]. Comparatively, our vision utilizes AI/ML models to characterize the local I/O behavior of individual applications as well as the global models utilized for the overall I/O traffic and patterns visible at the system level. Thus, the local and global AI models and the derived features can be utilized at different levels to predict each individual application behavior and the interaction between applications and the target I/O system in order to: (i) schedule the data transfer from the main application to the different storage tiers deciding which storage tier the data should be written and when it should be written, (ii) schedule the draining of data from the higher storage tiers to the lower storage tiers, for ultimately archiving the data, (ii) combine it with the system level patterns in order to choose the best moments to transfer the data between multiple storage tiers so that to minimize congestion at the I/O level.

Being able to understand the I/O patterns of applications with the system level traffic trends would allow the design of I/O middleware and system software, which can optimize their configurations (e.g., the best configurations for MPI aggregation and for data layout in Lustre).

Timeliness or maturity: The storage system connected to Frontier is 5 times faster than the one attached to Titan, however, Frontier is 50 times more powerful than Titan. Furthermore, with the emerging AI applications in DOE HPC workloads, our I/O patterns are rapidly changing at individual application and file and storage system levels and this trend is expected to continue with the advent of edge computing. Additionally, the availability of many cores and the flexibility to use threads in current supercomputers allows for creating asynchronous I/O solutions that were not possible in previous generations. DOE has been investing in I/O middleware libraries for more than a decade and there are mature products such as ADIOS, HDF5, and MPI-IO that can provide a clean abstraction between the applications and the proposed intelligent I/O organizer, simplifying the adaptation by HPC centers.

References:

[1] Zhang, Di, et al. "RLScheduler: an automated HPC batch job scheduler using reinforcement learning." *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2020.

[2] Xie, Bing, et al. "WIRE: Resource-efficient scaling with online prediction for DAG-based workflows." *2021 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 2021.

[3] Isakov, Mihailo, et al. "HPC I/O throughput bottleneck analysis with explainable local models." *SC20: International Conference for High Performance Computing, Networking, Storage and Analysis*. IEEE, 2020.

[4] Godoy, William F., et al. "ADIOS 2: The Adaptable Input Output System. A framework for high-performance data management." *SoftwareX* 12 (2020): 100561.

Title: Surfacing and Exploiting Metadata Relationships for Scalable Scientific Data Environments

Authors: Greg Eisenhauer (Georgia Tech, eisen@gatech.edu), Jeremy Logan (ORNL, lot@ornl.gov), Craig Ulmer (Sandia, cdulmer@sandia.gov), Patrick Widener (Sandia, pwidene@sandia.gov, corresponding author), Matthew Wolf (ORNL, wolfmd@ornl.gov)

Topic: Services for Rich Science Metadata

Challenge: Having recognized the benefits of annotating scientific data with metadata of many different types (structure, policy, provenance), the computational science community now faces the challenge of how best to go about it. Basic metadata is provided by POSIX-flavored filesystems, and file directory naming is still widely abused as a method of encoding important information about collections of bits. Container formats such as HDF5 provide APIs for attaching rich structure and access metadata. Local, bespoke solutions which provide lookaside metadata for applications and workflows have proliferated, with varying degrees of uptake, maintainability, opacity, and durability. The transitive closure, as it were, of this metadata (for a set of related artifacts), is of great utility for implementing a range of useful data management strategies. However, the variety of metadata representation and access mechanisms, each with its own methods for interrogation and update, is a significant barrier.

More importantly, these collections of metadata are *fragile*: they depend on irregular, implicit connections maintained through labor-intensive, manual processes and which frequently are only available as shared cultural knowledge within collaborative groups. Because of this fragility, it's exceedingly difficult given today's tools to associate or query arbitrary relationships between data artifacts. A POSIX filesystem can tell you about how files should be grouped in a hierarchy, but struggles to express multiple hierarchies among a set of items (let alone non-hierarchical relationships). Container formats represent scientific data very efficiently, but are not well-suited to expressing or discovering sharing and usage policies which apply to that data. Lookaside solutions can provide extensive and arbitrary metadata, but it is then difficult to apply FAIR (Findable, Accessible, Interoperable, Reusable) principles to that metadata.

Opportunity: Robustly defined and semantically rich metadata can help us to build software which can more completely address the complexities of scientific data. These complexities show up in the sheer variety of possible usage scenarios, only a few of which we can include here. For instance, scientific data can have many consumers with different access, formatting, and organizational requirements. Better metadata can enable per-consumer customizations which are frequently regarded as impractical today because of the wide range of possibilities. Performance improvements can also be enabled through flexible representations of relationship-based metadata. Similarly, semantically richer metadata could drive implementation of the kinds of flexible and granular data protection policies needed in order to support advanced scientific development methodologies such as DevSecOps.

These observations are not in and of themselves novel. The key opportunity we see is that there have been substantial advancements made in understanding open knowledge networks

and/or knowledge graphs as high performance abstractions for encoding complex relationship data. More than just provenance or campaign parameter information, such knowledge graphs can also offer ways to encode the many different *views* over scientific data that might be relevant. Many scientific datasets may have standard prefix tree indices to allow access in a standardized order, but they also store spatial or spatio-temporal data that might be well-represented in an R*-tree for optimized query patterns. Imagine if one could replace a POSIX directory hierarchy with an R*-tree structure, and all of the possible optimizations that might come from a system being able to understand that spatial context for the data. Such an implementation would provide efficient queries for interrelated scientific data not possible on today's systems without support from external services. The ability to embed the intent and the index structures for different queries into a coherent and optimizable knowledge graph representation enables richer automation of services.

One component of investment in new representations of rich metadata is a new ecosystem for portable, reusable, and transferable data services that function over and maintain that metadata. Such data management services could include the following:

- automatically localize a segment of the data archive at a particular location based on access patterns;
- evaluate cost functions which govern whether it's better to store a particular type of data or recreate it on demand;
- apply granular data protection to protect privileged information when making published data sets available;
- track federated facility administration policies to automate and to audit compliance with site-specific requirements;
- aid in boot-strapping a coherent campaign data record from the data hoard of disparate file entities; and
- efficiently shuffle and clean longitudinal scientific data for presentation as Arrow or Pandas data frames as used in AI/ML workflows.

The current context for microservices and service composition techniques for scientific data lags because we have lacked a sufficiently rich and high performance way of marking up the data. Too much of that management has lived in the implicit assumptions of each researcher's or community's chosen implementations. Achieving a shared ecosystem for multi-indexed, richly annotated scientific data will be an essential part of supporting DOE's future science needs.

Timeliness: There's a growing acknowledgement that metadata, provenance, and reproducibility of our large scientific datasets are increasingly important. Today's data sets will only grow larger, and the multi-, inter- and trans-disciplinary teams of the future will need even more support to be able to use them. To make sure we are capturing and expressing all of the relationships and contexts that are relevant for achieving DOE science mission goals, it is critical that we invest in research that moves to a more sustainable data management and policy solution. Formalizing support for the complex relationships between scientists, software, and data in a cohesive metadata system thus becomes both an important method of extracting domain knowledge from experienced researchers who may over time become less active, but also of curating and transferring that knowledge to new generations of researchers.

AI/ML for Storage Parameter Optimization of Large and Complex Data Stores of High Energy Physics Experiments

Author:

* Dr. Peter van Gemmeren (gemmeren@anl.gov, Argonne National Laboratory)
Dr. Alaettin Serhan Mete (Argonne National Laboratory)
Dr. Walter Hopkins (Argonne National Laboratory)

Topic:

Utilizing AI/ML to optimize storage settings.

Challenge:

For the last decade, High Energy Physics (HEP) experiments, such as ATLAS [1] and CMS [2] at the Large Hadron Collider (LHC), provide some of the largest and most complex scientific data stores [3]. This paper focuses on the ATLAS experiment as an example, but circumstances are similar for other HEP experiments.

Having recorded billions of particle collision events with the ATLAS detector, scientific discoveries, such as the discovery of the Higgs Boson in 2012, depend on processing and storing several 100's PB of complex scientific data. HEP experiments typically use C++ to reconstruct detector measurements into physics objects and data representations that use all of C++'s advanced features (including but not limited to, varying size containers, pointers, inheritance and polymorphism) need to be made persistent. Furthermore, given the large amount of data, efficient data compression is mandatory (e.g., ATLAS achieves a compression factor of 3 - 4 and still needs about 200 PB of disk storage).

The most efficient data compression can be achieved when corresponding records from different events are combined into the same compression buffer. Splitting object members into separate compression buffers may increase their compression factor if the buffer size is sufficient. However, it should be noted that these compression buffers will be the smallest read access unit for downstream processing. Therefore, choosing these storage parameters, including compression algorithms and levels, is a complex and manual task that requires balancing metrics such as storage needs, I/O speed and memory requirements not only for the producer but also the consumer.

The next generation of HEP experiments, including ATLAS and CMS at the High Luminosity LHC (HL-LHC) and DUNE will require even more data, making the task of optimizing I/O and storage even more critical and challenging [4].

Opportunity:

ATLAS uses powerful workflow monitoring tools [5] that can deliver deep insights into data consumer workflows, such as their read access patterns. This information is especially useful for physics analysis processes that are far less predictable than upstream production processes (such as reconstruction and simulation). Using methods of AI/ML to learn from data access patterns and efficiencies of these jobs, optimal storage parameter settings for the input data could be derived automatically.

Timeliness or maturity:

LHC and its experiments, including ATLAS and CMS, are scheduled to start the next period of collisions and data-taking, so-called Run 3, in early 2022. The processing framework for ATLAS, Athena has been successfully migrated to support multithreading, and a new analysis model has been established. These changes meant storage settings for new data products had to be determined, which had to be done using limited studies resulting in only approximate optimization. Developing an automated mechanism of determining storage parameters, would be timely to be exercised in Run 3 and deliver results in terms of storage savings and better I/O performance for analysis. However, as Run 3 is the last data taking period before the major luminosity upgrade of HL-LHC, this would present an important testbed as the higher dater rates make such improved optimizations critical.

References

[1] ATLAS Collaboration, 2008, The ATLAS Experiment at the CERN Large Hadron Collider, JINST 3 S08003.

[2] CMS Collaboration, 2008, The CMS Experiment at the CERN LHC, JINST 3 S08004.

[3] van Gemmeren P and Malon D, 2009, The event data store and I/O framework for the ATLAS experiment at the Large Hadron Collider, IEEE International Conference on Cluster Computing, DOI: 10.1109/CLUSTR.2009.5289147

[4] ATLAS Collaboration, 2020, ATLAS HL-LHC Computing Conceptual Design Report, CERN-LHCC-2020-015.

[5] Vukotic I, Gardner R and Bryant L, 2017, Big Data Tools as Applied to ATLAS Event Data, Journal of Physics Conference Series 898(7):072003, DOI:10.1088/1742-6596/898/7/072003

Autonomic Data Management for In-Situ Workflows

Philip Davis, University of Utah, philip.davis@sci.utah.edu
Zhe Wang, Rutgers University, jay.wang@rutgers.edu
Manish Parashar, University of Utah, manish.parashar@utah.edu

Topic: Optimizing data movement for adaptive in-situ workflows

Challenge: Scientific simulations running at extreme scales are generating increasing quantities of data, rapidly making it impractical to store this data for subsequent analysis. The challenges associated with the large data volumes are further exacerbated by current architectural trends in leadership machines towards significantly lower ratios of IO to CPU capacity. As a result, data processing is increasingly being performed *in-situ* using resources on the system where the data is produced, i.e., using in-situ workflows. These workflows can range from tightly coupled where analyses are linked directly to the simulation binary, to loosely coupled where analyses are run in independent processes.

While in-situ workflow provide an approach for effectively realizing the potential of extreme-scales for science, they also present complex runtime management challenges. These challenges include the effective placement of data both across the nodes of the system as well as across a deep memory hierarchy so as to optimize data access times while also minimizing any impact on the execution workflow components.

There are also trade-offs between performance and flexibility in choosing between tightly and loosely coupled analysis. A tightly-coupled analysis does not require coordination or data transfer between data producer and consumer, and is generally more efficient on a per-process basis. However, if the analyses' run time is not uniform across the process group, there is a risk that a relatively small number of long-running processes blocks the simulation, which often includes a numerical solver that imposes synchronization across all the ranks of the simulation. This is potentially catastrophic for a large-scale simulation, as all but a few processes will be idle in such an unbalanced scenario.

Loosely coupled analyses, on the other hand, offer a solution to this problem by decoupling the activity of the analysis routine from the simulation processes. Long running analysis processes can be overlapped with simulation and even moved to idle resources to balance the overall computing load. However, loosely coupled analysis requires additional overhead for data transfer and coordination.

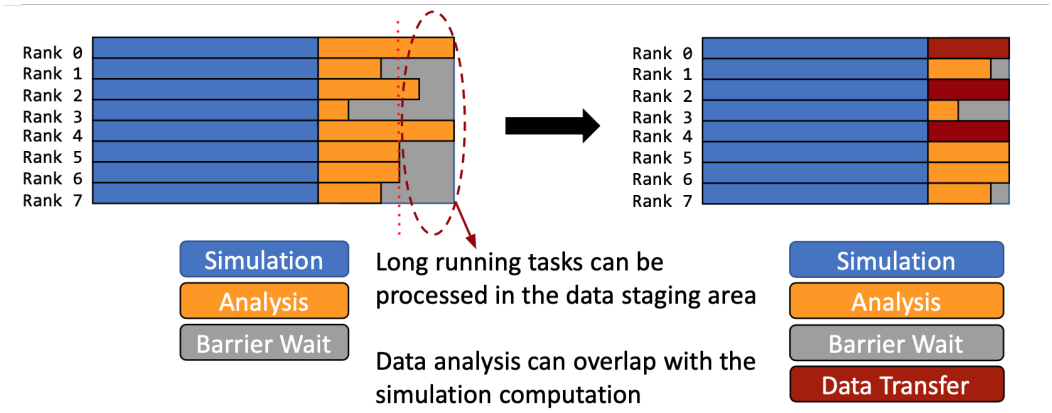


Figure 1: A hybrid approach to autonomous rebalancing of in-situ analysis. Long running analyses are migrated to out-of-process resources, rebalancing the workload of the simulation code.

Recently, progress has been made in optimization for this trade-off through a hybrid tight/loose coupling between simulation and analysis codes[1]. The analysis code can be deployed in either coupling mode and moved back and forth between loose and tight coupling as processing load shifts. This allows a dynamic response to load imbalance and autonomous rebalancing of CPU resources. Figure 1 illustrates this method, showing the process of rebalancing using a off-node resources.

This approach is paradigmatic of a class of performance optimizations that autonomously alter the computation load of a program based on data-driven triggers. More generally, these triggers can be influenced by data and system characteristics and are not known *a priori* and so may not be predictable before runtime. As ML-based optimizations are increasingly applied to data management it is reasonable to expect a proliferation of load-shifting optimizations based on learning about runtime characteristics.

However, as the computation is shifted, data access requirements also shift. Dynamic computation balancing results in dynamic data generation and access patterns. These shifting patterns can result in imbalances in network and memory resource usage that cause secondary performance degradations. Further, since the characteristics that influence load rebalancing are not known *a priori*, it may not be feasible to predict what data management resources will be needed until a workflow is already in progress.

Opportunity: Resource management is crucial to achieving high performance at large scale. Developing methods to coordinate activities between autonomous analysis activity and data management operations is an opportunity to improve the performance of large-scale in-situ workflows.

Notionally, this coordination can be explicit or implicit. Explicit coordination requires the development of interfaces to export understanding from the workload balancer to the data management framework (i.e. what data is needed by an analysis process, and under what circumstances those needs will shift). This is a feasible approach under some load shifting solutions, but could break down for ML-based triggering of load shifting since it may be difficult to articulate the conditions under load shifts through a software interface.

Implicit coordination of data management with autonomous trigger-based rebalancing is, in a sense, fighting fire with fire. Implicit coordination answers the data management challenge with an autonomous, intelligent data management framework. By identifying trends in data access patterns, an intelligent data access framework can anticipate and respond to future data needs by moving data closer to the predicted data consumer[2]. Further resource balancing can be achieved by time-sharing network and memory resources that are shared between data management and simulation tasks.[3]

Timeliness: Rebalancing analysis computation load can be expected to improve performance in the presence of an uneven analysis workload. Some use cases where this is the case are when the compute time of an analysis is data-dependent (such as during an isosurface calculation) or when an analysis involves a random walk or Monte Carlo sampling leading to variable convergence time.

Many ML algorithms suitable for the analysis of simulation data have characteristics that are likely to lead to unbalanced analysis workloads. ML-based scientific computing is an area of rapid development and it is likely that as the use of ML in analyses increases, so too will workload imbalance and the performance risk identified in the Challenge is likely to be an emerging issue for this growing class of workflow. Additionally, ML-based algorithms present a relatively hard version of the problem identified, typically requiring an autonomous data management solution.

References

[1] Z. Wang, P. Subedi, M. Dorier, P. E. Davis, and M. Parashar, "Adaptive placement of data analysis tasks for staging based in-situ processing," in *2021 IEEE 28th International Conference on High Performance Computing, Data, and Analytics (HiPC)*. IEEE, 2021.

[2] P. Subedi, P. E. Davis, and M. Parashar, "Leveraging machine learning for anticipatory data delivery in extreme scale in-situ workflows," in *2019 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 2019, pp. 1–11.

[3] ——, "Rise: Reducing i/o contention in staging-based extreme-scale in-situ workflows," in *2021 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE, 2021, pp. 146–156.

Models and Tools for Composing Complex in-situ Workflows

Philip Davis, University of Utah, philip.davis@sci.utah.edu
Manish Parashar, University of Utah, manish.parashar@utah.edu

Topic: Interfaces for data coupling in complex workflows

Challenge: Computationally addressing current scientific and societal grand challenges is increasingly leading to complex, data-intensive applications workflows rather than monolithic application codes[1]. Furthermore, current architectural trends of large-scale leadership computing systems, such as decreasing ratio of IO to CPU resources, are resulting in *in-situ* formulations of these workflows where analysis and visualization tasks run alongside coupled simulations on the same systems, interacting and exchanging data in real-time mostly using system memory rather than persistent storage [2]. Figure 1 shows an example of a complex workflow integrating multiple simulation, analysis, and visualization components. All components are run simultaneously, and data are exchanged using a coupling framework.

The development of such in-situ workflows has resulted in a positive feedback loop in co-design. For example, as systems software is developed to support in-situ workflows, it is also leading to the generation of new modes of interaction between workflow components. For example, when run in-situ, the results of an analysis program can be used to steer the simulation in ways that wouldn't be possible if the analysis is run *post hoc*. Furthermore, the real-time visualization of simulation results is leading to new opportunities for scientists to interactively direct simulations, reacting in real-time to surprising results. As the software environment supporting in-situ workflows as well as the workflows themselves become more complex through this iterative co-design process, the need for high-level workflow design abstractions and tools have become apparent.

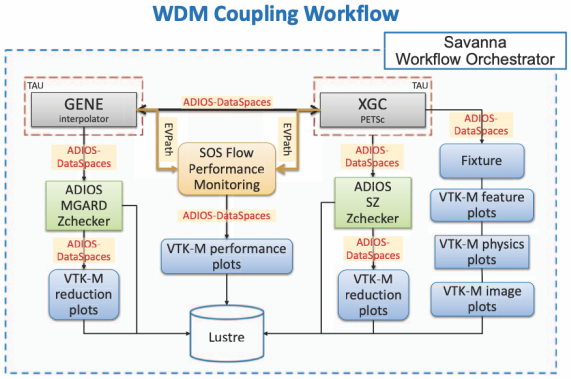


Figure 1: The component interactions of the WDMApp coupled multi-simulation workflow, which includes analysis and visualization routines.[1]

The development of these in-situ workflows is often organic, and as a result, it frequently requires the modification of existing simulation, analysis, and visualization codes/packages to support in-situ composition and operation. Converting existing software packages for use in in-situ workflows has largely been an *ad hoc* process, requiring developer teams familiar with each code to coordinate their efforts generating compatible interfaces between workflow components. Modifying a workflow constructed in such a way requires further coordination, reducing the reusability of the workflow components. Among the complications of coupling codes eien this way, ensuring data compatibility is a critical. Data are generated and consumed with certain units, precision, grid resolution, etc. If an understanding of the parameters of a data set is not shared between data producer and data consumer, then the workflow can fail or (worse) produce spurious results. Self-describing data sets have helped ameliorate this concern to some degree, but this solution require that data validation be implemented and applied on each component, for each interaction.

Recent work has begun to address the issue of workflow development complexity by defining programming models for **composable workflows**[3]. At a high level, these models typically allow abstract Workflow-Level Interfaces (WLI) to be defined and wrapped around application code to create workflow components, and additionally provide semantics for composing workflows that describe the interactions between components using elements of the WLI. These interactions are specified in a **workflow descriptor**. A well-defined workflow interface creates opportunities for resusability of workflow components, as well as multiple implementations of the same WLI (i.e., different software packages that can take on the same role in a workflow). However, further work is needed to standardize and mature composable workflow models.

Opportunity: As complex in-situ workflows become essential to large-scale scientific computing, high-level programming and runtime support that improves scientists' productivity by allowing them to simply and flexibly develop, execute and experiment with these workflows can have a tremendous impact. To this end, workflow descriptor languages are an opportunity to communicate information about the workflow to underlying middleware and system libraries. One of these opportunities is in data compatibility checking. In developing composable workflows, the common data model can be lifted out of application code so that it can be specified once and validated by the workflow framework at compile or run time. Doing this requires semantics for describing the data format provided and expected by each interface, as well as the ways that different aspects of each components' independent data model are related (e.g. what are the relative grid coordinates between two coupled simulation components) and can even include data about workflow intentions (e.g. where in a global data domain should errors be minimized in interpolation).

Providing a common data model that is defined by workflow developers and known *a priori* offers many additional advantages. Beyond validating the compatibility of data exchanges, a data model provides application intelligence that can be leveraged by the developers of the libraries and frameworks that realize composed workflows. As a notional example, if a data producer is generating data at a higher precision than any consumer in the workflow will require, an optimization could be dropping higher excess-precision bits to reduce transfer and storage costs. This sort of optimization can be difficult to provide without a common data model, but if the workflow developer is able to define such a data model, then the workflow framework can implement such an optimization "behind the scenes".

Further, with common understanding about how the data will be generated and consumed, many data translation features of workflow development that are onerous and repetitive to implement in ad hoc in-situ workflows can be integrated into workflow frameworks. This reduces the amount of effort required to develop complex workflows. For example, different workflow components may use overlapping or mismatched grids. Correctly implementing data translation between components is an expensive and error-prone task that must be repeated for every interaction in ad hoc workflow development. Composable workflows create the opportunity to singularly implement and optimize this functionality for a large group of common data transformation tasks. The key to performing the right transformation in an efficient way is an expressive data model. Developing an expressive data model and integrating it into one or more composable workflow frameworks provides an opportunity to bridge the gap between application-level data understanding and opportunistic data management framework optimizations in a way that can be coupled to the workflow logic itself.

Developing a data model that improves the workflow development codesign process will require collaboration between workflow developers, application code developers, and computer scientists. This requires an understanding of not just what type of data parameters fit the needs of code compatibility, but also which parameters provide information that is likely to be useful to data operation optimization.

Timeliness: As in-situ workflow are increasingly used by applications address important scientific grand challenges, their complexity will continue to grow, as will the need for abstractions and tools to support composable workflow development. This is a nascent period for the composable workflow paradigm. Establishing an expressive data model in conjunction with the development of this paradigm has the potential to improve the value of adoption to both workflow developers and data scientists. Without a suitably expressive data model, there is a risk that workflow code developers will continually re-implement the same data transformations (and this is not uncommon in *ad hoc* coupled workflows today). To mitigate this risk, it is desirable to gain understanding of what makes an expressive and useful data model before the paradigm has matured.

At the same time, developments in data management middleware have reduced the development effort by enabling the modular development of data services. This has created new opportunities for a rapid codesign cycle that allows data management optimizations to be quickly prototyped and introduced into workflow frameworks. This decreased development time has increased the ability of data management framework developers to translate data understanding into optimizations in a timely way, increasing the value of data insights that could be gleaned from an effective data model.

References

[1] A. Bhattacharjee, C.-S. Chang, and J. Dominski, "Exascale computing and whole device model for fusion," *Workshop on Computational Nuclear Science and Engineering Presentations*, 2021.

[2] A. Al-Saadi *et al.*, "Exaworks: Workflows for exascale," *arXiv preprint arXiv:2108.13521*, 2021.

[3] P. E. Davis *et al.*, "Benesh: a programming model for coupled scientific workflows," in *2020 IEEE/ACM 5th International Workshop on Extreme Scale Programming Models and Middleware (ESPM2)*. IEEE, 2020, pp. 1–9.

The promise of computational storage for scientific applications

Philippe Bonnet[*], Ana Gainaru[+], Norbert Podhorszki [+], Scott Klasky [+]
[*] IT University of Copenhagen – phbo@itu.dk
[+] Oak Ridge National Lab - {gainaru, pnorbert, klasky}@ornl.gov

Topic. Interfaces for accessing data

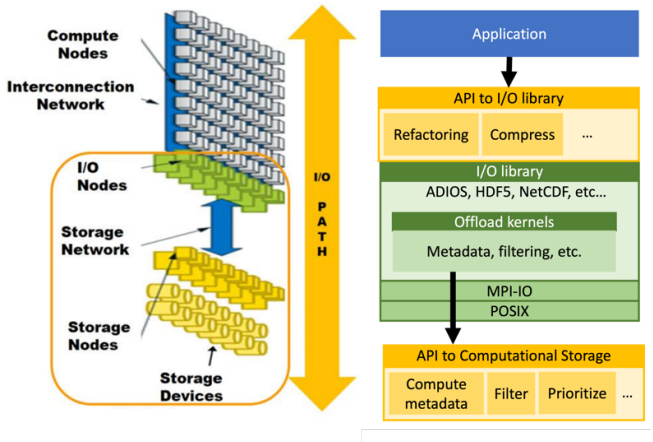
Challenge

Science relies on increasingly accurate digital representations of complex natural phenomenon. Current trends include simulations of complex systems by coupling of codes designed for individual components [1], or learning and continuously updating simulation models based on measurements from large-scale observational and experimental facilities [2], [3]. As a result, ever larger volumes of data are stored and accessed by ensembles of scientific applications with tight requirements in terms of throughput and latency.

This evolution creates new challenges for HPC storage systems. First, we expect new I/O patterns characterized by concurrent flows of (a) random reads resulting from sampling large collections to avoid overfitting when training models, (b) sequential reads resulting from strong coupling of codes, and (c) writes of extremely large data elements (multiple TBs in size), produced by simulation or measurements, in high velocity bursts. This diversity of I/Os will result in new levels of interferences in storage systems and thus a high variability in I/O performance. I/O jitters and I/O throughput variance were identified as bottlenecks years ago [5]. Even with traditional HPC I/O workloads, dominated by sequential writes, these problems result from competing traffic, contention on storage servers or I/O routers, and concurrency limitations in the client compute nodes. They cause suboptimal I/O performance. We expect that new I/O patterns will exacerbate these problems. Getting near-optimal I/O performance will require **dynamic management of I/O interferences in space (managing the I/O path) and time (managing I/O scheduling).**

Second, the volume of data to be stored and retrieved is so large that data compression/refactoring is necessary. Tradtional approaches based on lossy compression sacrifice fidelity for performance. This is no longer adequate for the level of accuracy and performance required by the next generation of scientific applications. Ideally, the most interesting data in a collection can be returned first within a given time constraint. This requires enough **meta-data and appropriate models to take these decisions at run-time**, within storage nodes, without negative impact on I/O performance.

Opportunity



Computational storage denotes the integration of programmable compute resources within storage devices, as opposed to traditional disks that are pre-programmed by their vendors to just read and write data blocks. Computational storage makes it possible to run portions of data pipelines on storage devices, and thus promises reduced data movement, better energy efficiency and reduced costs. These are specialized processing units, including multi-processor system on a chip or data processing units (DPU) equipped with hardware accelerators, that are cheaper and more energy efficient than general-purpose CPUs.

Interfaces to the I/O library and to the storage system are needed in order to utilize computational storage devices to make it possible to program the storage infrastructure [7] to meet the needs of data pipelines and minimize data movement. More specifically, programming the storage infrastructure entails:

1. **Defining new storage interfaces:** Computational storage can expose storage with a variety of interfaces that encapsulate relevant processing [8], e.g., reading quantized samples from a collection, returning refactored samples from a collection, generating metadata when data is written. By using computational storage, I/O libraries will be able to dynamically manage I/O interfaces to define possible in-

transit data queries and conversions. The I/O library could use these interfaces to offload some of the most computational intensive internal kernels (like metadata generation) as well as to define interfaces visible by the application to allow scientists to request application-specific offloads (like reduction, filtering, etc.)

2. **Shipping code from compute nodes to storage devices:** Portions of data pipelines can be shipped to computational storage at run-time to reduce data movement [9]. I/O libraries in the future will be designed to take advantage of this capability and adapt to the requirements of the application and system behavior. Metadata generation and handling and data pre-processing are ideal candidates. Filtering the data needed by an application by checking properties of its metadata before transferring it to the application will help reduce traffic and avoid system level congestion. Moreover, I/O libraries will have the possibility to adapt to the performance of the system (e.g. combined with a refactoring method, the storage could be given rules to prioritize the data and progressively transfer it to the application so that the application does not need to wait for the entire data to be received before starting the execution)

The advantages of desiging interfaces between applications and I/O libraries and between I/O libraries and storage are multi-folded [1]: (i) Offload I/O internal expensive kernels to decreases the application end-to-end execution time; (ii) Allow I/O libraries to adapt to the needs of applications by offloading application-specific data transformations like local metadata reads and data filtering (Such a decoupling of meta-data and data generation is possible for self-describing data formats); (iii) Offloading data refactoring functionalities from the compute nodes to storage devices further reduces I/O traffic between compute and storage nodes; (iv) Allow the I/O library to adapt to the performance of the network and I/O patterns of the application. Ideally, compute nodes should access the most interesting data items in a collection within a given time and traffic budget. This requires that computational storage is able to make relevant predictions. Such predictions could be based on a model instantiated or trained on computational storage, based on locally generated meta-data.

Timeliness

Computational storage is in the process of being standardized. A task force at the Storage Networking Industry Association (SNIA), a trade group representing storage companies, defined terminology and architectures for computational storage in August 2020. They denote the processing units integrated with storage as *computational storage processors*. When combined with traditional storage drives or hubs, they provide Computational Storage Services (CSS) to hosts.

An extension of the NVMe standard for computational storage is expected in 2022. Such a standard will define mechanisms for uploading code to computational storage. We expect that it will be possible to associate computational storage cards (e.g., Bittware 220-U2 or Bittware IA-840F[1]) to existing NVMe SSDs, in the coming year.

The NVMe standard for Solid-State Drives was first defined in 2011. Linux support was introduced in 2013 [11]. NVMe SSDs were introduced with the 4th generation Oak Ridge Leadership Computing Facility in 2018. If we assume a similar process for computational storage, we can expect its introduction in Leadership Computing Facility before 2030. It is time to prepare for the co-design of the storage infrastructure and scientific applications to ensure efficient management of I/O interferences as well as adaptive data refactoring.

References

[1] J. Logan *et al.*, 'Extending the Publish/Subscribe Abstraction for High-Performance I/O and Data Management at Extreme Scale', *Bulletin of the IEEE Technical Committee on Data Engineering*, vol. 43, no. 1, Mar. 2020

[2] T. Pfaff, M. Fortunato, A. Sanchez-Gonzalez, and P. W. Battaglia, 'Learning Mesh-Based Simulation with Graph Networks', *arXiv:2010.03409 [cs]*, Jun. 2021

[3] P. Damme et al., 'DAPHNE: An Open and Extensible System Infrastructure for Integrated Data Analysis Pipelines'. CIDR, 2022.

[4] L. Wan *et al.*, 'Improving I/O Performance for Exascale Applications Through Online Data Layout Reorganization', *IEEE Transactions on Parallel and Distributed Systems*, vol. 33, no. 4, pp. 878–890, 2022

[5] B. Xie *et al.*, 'Characterizing output bottlenecks in a supercomputer', SC, 2012

[6] J. Chen *et al.*, 'Scalable Multigrid-based Hierarchical Scientific Data Refactoring on GPUs', *arXiv:2105.12764 [cs]*, May 2021.

[7] J. Do, S. Sengupta, and S. Swanson, 'Programmable solid-state storage in future cloud datacenters', *Commun. ACM*, vol. 62, no. 6, May 2019.

[8] J. Do, I. L. Picoli, P. Bonnet, and D. B. Lomet, 'Better Database Cost/Performance via Batched I/O on Programmable SSD', *VLDB Journal*, 2021.

[9] P. Bonnet, 'Computational Storage Capabilities'. EU Daphne project deliverable. [Online]. Available: http://daphne-eu.eu/wp-content/uploads/2021/11/Daphne_D6.1_Design-Space-IO-Hierarchy-1.pdf

[10] A. Lerner and P. Bonnet, 'Not your Grandpa's SSD: The Era of Co-Designed Storage Devices', SIGMOD, 2021.

[11] M. Bjørling, J. Axboe, D. W. Nellans, and P. Bonnet, 'Linux block IO: introducing multi-queue SSD access on multi-core systems', *SYSTOR, 2013*.

Revolutionizing the I/O Paradigm for Scientific Data Analytics

Q. Liu[1], S. Klasky[2], J. Chen[2], B. Whitney[2], R. Wang[2], Q. Gong[2], X. Liang[3], L. Wan[2], D. Pugmire[2], K. Moreland[2], N. Podhorszki[2]

1 New Jersey Institute of Technology, qliu@njit.edu, 2 Oak Ridge National Laboratory, {klasky, chenj3, whitneybe, wangr1, gongq, wanl, pugmire, morelandkd}@ornl.gov, 3 Missouri University of Science and Technology, xliang@mst.edu

Topic: Interfaces for accessing data, storage-system architecture design

Challenges

Present-day storage design and I/O methods treat data read from the storage system as opaque byte sequences. This simplistic abstraction, along with the limits it imposes on data management, is problematic and a poor fit for the dynamic needs of scientific applications. For example, application codes may want to retrieve data at different precisions/resolutions when the storage/compute resources for data analysis are highly constrained, or to explore a new feature of the data in an ad-hoc fashion. Such analyses are common in scientific workflows, particularly for datasets shared by broad communities, which are not well supported by the DOE SSIO toolchain today. There are several challenges which must be overcome in order to meet the needs of modern workflows. First, scientific data needs to be refactored into different levels of reduced representations such that approximations to the original data with different levels of accuracy can be dynamically recomposed to satisfy a broad spectrum of analysis needs [3]. A key challenge is that data refactoring needs to be done in an efficient manner that avoids significantly increasing the complexity and overhead of storage and I/O. Given the performance of next-generation storage systems, this is a formidable task. For example, DAOS, the I/O system of the upcoming Aurora system will achieve over 25 TB/s bandwidth. and with refactoring and recomposing, the end-to-end performance needs to be on par with writing the raw data to be useful [4,5]. Second, the hierarchical nature of the refactorization calls for new methods that can take full advantage of the hierarchical storage on DOE systems. Currently, scientists decide which files to move to lower layers in an ad-hoc fashion, before data is purged by the file system. By storing the smaller, lower accuracy representation on faster storage, the time to reach the required accuracy during analysis can be accelerated, compared to retrieving the same amount of data from a slow storage tier. How do we intelligently place refactored data in a way that is best suited to the storage hierarchy and how do we move data across tiers as the usage pattern changes for a particular workflow need to be addressed. Third, by fetching a reduced set of representations, the outcomes of data analysis can be negatively impacted and it requires a theoretical foundation to understand and bound the error so that the data representations can be fully trusted by the scientists.

Opportunities

To address these challenges, a team of researchers consisting of computer scientists, applied mathematicians, and application scientists have started an effort to completely redesign the I/O paradigm for scientific data analytics. As of now, the team has established a theoretical foundation to bound the error for reduced representations [1,2] and further research opportunities include: 1) **at the data generation stage**: fine-grained decomposition of data is needed in order to fully utilize the deep storage hierarchies on HPC systems as well as satisfy various accuracy needs for different data analysis tasks while minimize the I/O costs. Also, data decomposition needs to be device (e.g. accelerator or edge device) friendly at the data source, so that the decomposition can incur a low overhead to the critical path of a scientific run. This requires research that focuses on how to efficiently map data refactoring algorithms to a range of hardware architectures in HPC and edge devices, and how to be portable so that data can be refactored and recomposed on different types of system? 2) **at data storage, management, and retrieval stage**: Given a set of physical quantities evolving across time, different levels of errors after refactorization, and distinct user intentions, how do we maintain the relationships between different quantities of interest, data objects generated by refactorization, and errors associated with those data objects? How do we optimize the data placement of the refactored data based on

user intentions and characteristics of multitier storage hierarchy during the data life cycle, so that we can minimize the time to knowledge? It is necessary to design efficient metadata structures and algorithms to manage refactored data objects across the multitier storage hierarchy. Also, the data placement can affect the performance of both storage and retrieval. The optimal data placement for the refactored hierarchical data on multitier storage systems can be formulated into a combinatorial optimization problem, with the goal of finding the shortest time for fetching all required data levels with constraints on tier capacity and movement overhead. Further, we need to intelligently make data placement decisions leveraging ML/AI techniques. 3) **at the data analysis stage:** Dynamic recomposition of data based upon the user prescribed error tolerance and storage resource availability. The higher the error tolerance and the lower the resource availability is, the less the amount of data needs to be retrieved from the storage, thus greatly reducing the I/O time (and potentially compute time) for data analytics. This allows for a wide range of accuracy and performance needs (e.g., post processing versus near-real time processing for scientific experiments) from DOE applications across a variety of system environments. Given the fine-grained decomposed data, how to efficiently fetch and recompose data in a progressive manner to satisfy different accuracy needs? This requires research efforts that focus on understanding the complex relationship between the improvement of accuracy and the added I/O cost so that domain scientists can make the best decision with constrained resources. A more challenging task is how to enable data analytics to take advantage of the progressive data retrieval capability by incorporating iterative refinement in terms of resolution and accuracy. Finally, to further reduce the I/O and compute overhead of data analysis, research efforts need to be made to allow for partial refinement of the spatiotemporal domain for both data analysis and I/O middleware.

Timeliness

Scientific instruments have been continuously producing large amounts of data at an increased speed. This places tremendous stress on science campaigns as the storage capacity and I/O bandwidth have not grown as fast as the data rate, making it difficult for the data to be archived or moved from edge to HPC devices, and vice versa. Small files can stay on parallel file systems where the bandwidth is fast, however large datasets mostly need to be moved to lower storage tiers (e.g., HPSS) for capacity. For the current storage and I/O capabilities, the majority of data produced by large-scale facilities could never be read back for analysis due to the forbidden cost in data movement. A smart and resource-aware I/O and storage management method is of urgent need so that data analytics can benefit from the improved fidelity of simulation models and sensor readings.

Recently, a range of error-controlled data reduction tools have been integrated to I/O libraries, such as ADIOS, HDF5, allowing data to be reduced at a user-prescribed accuracy. Further combining data refactoring with the error-controlled reduction techniques makes it possible to decompose data into fine-grained representations, with each segment of representations to be stored independently on different storage tiers and be progressively retrieved with guaranteed accuracy to accommodate environments with different resource availability and user-prescribed data fidelity.

References

[1] Mark Ainsworth, Ozan Tugluk, Ben Whitney, and Scott Klasky. Multilevel Techniques for Compression and Reduction of Scientific Data—Quantitative Control of Accuracy in Derived Quantities. SIAM Journal on Scientific Computing 41 (4), A2146–A2171, 2019.

[2] Mark Ainsworth, Ozan Tugluk, Ben Whitney, and Scott Klasky. Multilevel Techniques for Compression and Reduction of Scientific Data—The Unstructured Case. SIAM Journal on Scientific Computing, 42 (2), A1402–A1427, 2020.

[3] Zhenbo Qiao, Qing Liu, Norbert Podhorszki, Scott Klasky, Jieyang Chen, "Taming I/O Variation on QoS-less HPC Storage: What Can Applications do?," Proceedings of the 31st ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis, Nov 17 - 22, 2020, Atlanta, USA.

[4] Tao Lu, Eric Suchyta, Jong Choi, Norbert Podhorszki, Scott Klasky, Qing Liu, Dave Pugmire, Matthew Wolf, Mark Ainsworth, "Canopus: enabling extreme-scale data analytics on big HPC storage via progressive refactoring," Proceedings of the 9th USENIX Hotstorage, p.28-28, July 10-11, 2017, Santa Clara, CA.

[5] Tao Lu, Eric Suchyta, Dave Pugmire, Jong Choi, Scott Klasky, Qing Liu, Norbert Podhorszki, Mark Ainsworth, Matthew Wolf, "Canopus: A Paradigm Shift Towards Elastic Extreme-Scale Data Analytics on HPC Storage," 2017 IEEE International Conference on Cluster Computing (CLUSTER), September, 2017, Honolulu, HI.

The Jot and Tittle of Workflows Interoperability: Towards FAIR Computational Workflows via Metadata APIs

Rafael Ferreira da Silva, Sean R. Wilkinson, Katie Knight, Valentine Anantharaj, Olga A. Kuchar
Oak Ridge National Laboratory, Oak Ridge, TN
silvarf@ornl.gov, wilkinsonsr@ornl.gov, knightke@ornl.gov, anantharajvg@ornl.gov, kucharoa@ornl.gov

Topic: Metadata management infrastructure for scientific workflows to support FAIR principles

Challenge: The FAIR principles [1] have laid a foundation for sharing and publishing digital assets and, in particular, data. The FAIR principles emphasize machine accessibility and that all digital assets should be Findable, Accessible, Interoperable, and Reusable. Scientific workflows encode the methods by which the scientific process is conducted and by which data are created. It is thus important that workflows both support the creation of FAIR data and adhere to the FAIR principles themselves. Workflows are hybrid processual digital assets that can be considered as data or software, or some combination of both. As such, there is a range of considerations to take into account with respect to the FAIR principles [2]. Some perspectives are already well explored in data/software FAIRness, such as descriptive metadata, software metrics, and versioning; however, workflows create unique challenges, such as representing a "complex lifecycle" from specification to execution via a workflow management system.

Workflow systems vary as widely as workflows do, resulting in a variety of approaches that differ in expressivity, execution models, and ecosystems. These differences are mainly due to individual implementations of language, control mechanisms (e.g., fault tolerance, loops), data management mechanisms, execution backends, reproducibility aspects for sharing workflows, and provenance and FAIR metadata capturing. Unfortunately, there is no attempt to develop an approach from a perspective of making interoperable components that require standardized APIs and metadata, which are still open challenges [3]. The need for interoperability is paramount, and it recurs at multiple technical levels (e.g., task, tools, workflows, data, metadata, provenance, and packaging) as well as non-technical levels including semantic, organizational, and legal issues (e.g., licenses compatibility, data sharing policies). The need for interoperability of workflow applications and systems is commonly modeled as a problem of porting applications and data management across systems, which may require anywhere from days to weeks of development effort. Most of the previous approaches for tackling the interoperability problem attempted to develop complete vertical solutions.

Opportunity: Given current efforts for developing FAIR data and software, it is important to first understand what efforts could be adapted to workflow problems. A fundamental tenet of FAIR is the universal availability of machine processable metadata. Developing methods for FAIR workflows requires community engagement: (i) to define principles, policies, and best practices to share workflows; (ii) to standardize metadata representation and collection processes about workflows; (iii) to create developer- and workflow-friendly guidelines and tools; and (iv) to develop shared infrastructure for enabling development, execution, and sharing of FAIR workflows [3]. Additionally, ensuring that provenance can capture the necessary information is key for enabling FAIRness in workflows. As a result, there is an opportunity for developing common APIs that represent a set of workflow library components, so that interoperability could be achieved at the component level, including APIs for defining inputs, storing intermediate results and output data, and automating the capture and annotation of metadata information.

Many provenance models [4] can be implemented or extended to capture the information needed for FAIR workflows. Additionally, FAIR principles are more likely to be followed if the process for capturing these metrics is automated and embedded in workflow systems. In this case, a workflow execution will become FAIR by default, or perhaps with minimal user curation. There is also a tendency to combine the workflow with its execution model and data structures (e.g., the intertwine between the abstract workflow, its execution, and its data management). It is then of the greatest importance to understand which

component in the workflow system architecture accounts for which functionality. Thus, separation of concerns is key for interoperability at many levels, e.g. separation of orchestration of the workflow graph, its execution, data management, and metadata capturing. There is an additional opportunity for exploring how provenance is represented in the metadata of these workflow library components, as change across systems will be important for both knowledge representation and interpretation by humans and machines using the workflow. Furthermore, research into how the metadata itself is versioned will be necessary to understand how these components have changed over time.

Timeliness or maturity: Given the computational demands of many workflows, it is crucial that their execution be not only feasible, effortless, and efficient on large-scale HPC systems (in particular upcoming exascale systems), but also metadata and provenance capturing needs to be automated and comprehensive; thus FAIRness can be attained. Current efforts to apply FAIR principles to data and software (e.g., FAIR4RS and FAIR for Virtual Research Environments) tackle the problem by considering workflows as software. The European EOSC-Life Workflow Collaboratory, for example, has developed a metadata framework for FAIR workflows based on schema.org, RO-Crate, and the common workflow language (CWL), which could lead to standardization of metadata about workflows. On the other hand, most efforts to unify workflow systems and/or their components (in particular data and metadata management) have led to the specialization of some of these standards which may require that other systems conform to that specification, thus resulting in low adoption. Attempts to standardize may also lead to overly generic interfaces that ultimately inhibit usability and lead to hidden incompatibilities. Efforts such as the GA4GH-DREAM have promoted “bake-offs” to compare and identify workflow systems capabilities to define standardization within domains. An open question is whether such attempts should be domain-specific or domain-overarching. As FAIR principle entitled "I1" [1] recommends using a “formal, accessible, shared, and broadly applicable language for knowledge representation”, understanding if and how workflows may need to diverge from such a recommendation is essential.

More recently, the need for distributed computing at scale with heterogeneous resources (HPC, cloud, edge, etc.) has emerged as the demand for processing and storage is continually increasing. This new class of workflow applications uses cross-facility resources (computational, storage, and visualization), and advanced network capabilities for large data movement. Interoperability among resources, and more importantly across facilities, is crucial for enabling seamless workflow executions. Efforts such as the DOE NERSC's Superfacility project provide the mechanisms for bridging experimental and observational instruments with computational and data facilities; the DOE OneID's project provides identity management and federated authentication access to shared/cloud services. Although these solutions are a step forward to enable access to (and to some extent interoperability among) resources/services, provenance capturing and automated metadata extraction is still an open question for this new class of high-profile, cross-facility workflow applications.

References:

[1] Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, Blomberg N, Boiten JW, Santos LB, Bourne PE, Bouwman J. Addendum: The FAIR Guiding Principles for scientific data management and stewardship. Scientific data. 2019;6:6.

[2] Goble C, Cohen-Boulakia S, Soiland-Reyes S, Garijo D, Gil Y, Crusoe MR, Peters K, Schober D. FAIR computational workflows. Data Intelligence. 2020 Jan 1;2(1-2):108-21.

[3] Ferreira da Silva R, Casanova H, Chard K, Altintas I, Badia RM, Balis B, Coleman T, Coppens F, Di Natale F, Enders B, et al. *A Community Roadmap for Scientific Workflows Research and Development*. 2021 IEEE Workshop on Workflows in Support of Large-Scale Science (WORKS), 81–90, 2021.

[4] Oliveira W, Oliveira DD, Braganholo V. Provenance analytics for workflow-based computational experiments: A survey. ACM Computing Surveys (CSUR). 2018 May 23;51(3):1-25.

Management and Storage of Scientific Data in the Context of Edge Computing

R. Sankaran*, S. Shahkarami, W. Gerlach, N. Ferrier, P. Beckman (Argonne National Laboratory)
I. Perez, I. Altintas (University California San Diego)

Topic: Data-management support and Storage-systems for AI@Edge.

1 Introduction

Applications involving voluminous data can often benefit from the computing being performed as close to the data source as possible. This need for computation at the edge arises due to communication constraints, privacy and sensitivity of data, latency and liveliness requirements, or costs of moving data. Machine learning and inference at the edge are growing at a rapid pace. In support of AI@Edge applications, a wide range of computing platforms (hardware and software) with widely varying resources, spanning from intelligent embedded devices like smart cameras, to powerful on-premise systems are being integrated at the source of the data. Devising metadata management infrastructure to support FAIR principles [8], capturing provenance, and providing data management support for AI is increasingly of importance to scientific edge computing.

Edge computing, featuring AI/ML, is rapidly being adopted by a wide range of DOE scientific domains. To address this, Argonne National Laboratory developed a state-of-the-art programmable and networked AI@Edge computing and sensing-actuation platform for science called "Waggle" [5]. Waggle edge-nodes use commodity AI-optimized processors and deep learning to process data directly at the edge and report analyzed results. Waggle has been used in a wide range of scientific projects and supported by academia, industry, and multiple federal agencies. It was the core platform used by the National Science Foundation (NSF) Array of Things (AoT); a DOE-NNSA–funded effort called DAWN, uses Waggle as the common platform for deploying new radiation sensors and urban radiation detection networks; the NSF-funded Sage project [4], based at Northwestern University, has expanded Waggle's AI edge computing capabilities and focused on cyberinfrastructure for "software-defined sensor" networks, whose functions can evolve over time with new measurement or policy demands; Argonne and Exelon corporation are exploring how AI-enabled sensors can improve electrical grid reliability and advance "smart city" technologies; and the DOE Vehicle Technology Office supports a project using Waggle to study traffic patterns at O'Hare Airport.

2 Opportunities and Challenges

A fundamental aspect of AI@Edge is locally sourced sensor data (camera images/video, LiDAR data, audio samples, beamline data etc.) being processed using AI/ML models that may be trained on "similar" data. Due to technical (repeatability of sensor orientation and calibration, local conditions etc.) and logistical reasons (availability of deployment resources), it is almost always the case that the sensors across different edge devices generate data that is different enough that the inference accuracy of an algorithm can vary significantly across the devices. Partial retraining of algorithms at the edge (light-weight training) using both local site-specific samples, and also specifically curated data from other edge devices, has the potential to greatly improve the algorithms, and their inference accuracy. One challenging aspect is maintaining labeled data locally at the device, both for local training, and gathering toward creating the curated datasets for the aggregate of edge devices. This data is often annotated with meta information and other useful contexts, should be cataloged, and searchable for local and global use, and available across the network in a seamless manner.

Machine learning applications are data hungry and require as many data points and features as possible to improve their predictions, which often requires integrating data from different sources. This is true for AI@Edge applications too. Edge computing often is not used in isolation, but through coupling with centralized HPC resources for a variety of reasons including generating a high-level holistic view (inference/analysis) across the whole application space and in-turn steering the computation at the edge, training new AI/ML models for the edge, running large-scale simulations that incorporate edge inferences and sensor data, and monitoring the health, and edge analysis accuracy and drift. A cloud-based (for scalability, geographical availability, latency) centralized resource often plays the role of an orchestrator and data gatherer for edge computing resources. For coupling edge applications to HPC

*corresponding author: rajesh@anl.gov

computations (edge driven HPC and HPC steered edge sensing and computing), seamless, low-latency, on-demand, and bidirectional flow of data mediated by the cloud is of utmost importance.

The above two examples clearly illustrate the necessity of FAIR Principles. Together, edge, cloud, and HPC are a fertile ground for a variety of workflows and triggers. Traditional data access in this continuum still has not matured to provide both static "bulk access" to curated data for analysis, and fast-paced, dynamic, and low-latency access for rapid observations and triggering in a consistent fashion. For example, triggers could include conditions such as "Notify me if I detect more than X events / Y mins", "Run HPC computation if more that X edge devices identify a certain event", "Update the model if prediction accuracy is consistently lower than Y and there is no change in environmental condition", "Notify me if my program fails to meet a certain frequency of measurement", and "Run traffic state analysis model at the edge when more than 10 automobiles are seen in a minute" among others. The storage and data management system has to support powerful metadata queries and scalable time queries with few aggregations to make the above triggers and observations possible in quasi/realtime.

Data systems for managing the AI/ML life-cycle is a fundamentally challenging problem. Can we build a unified data system which manages the complete AI/ML lifecycle, while also negotiating the bandwidth, latency and availability challenges that are unique to edge computing? Such a system would include data management, training, monitoring and actionable feedback from models in production, specifically, understanding the right metrics and presenting them in a meaningful way to users, and retraining. Borrowing from the earlier example of retraining and deploying the AI/ML algorithms to different edge devices, this process involves a number of considerations and constraints including evaluation of the knowledge content of a piece of data, human in the loop or automated labeling, periodic refinement, optimization, and versioning of data sets, and incorporation and tracking of changes in dataset. A few commercially available tools extend parts of this functionality to generic AI/ML applications, and provide motivation for the Edge-to-HPC scenario. For example, Weights & Biases [6] has a data management product focused on AI/ML, Label Box [1] and Label Studio [2] incorporate human-in-the-loop and automatic pre-labeling and model based quality ranking, and Paperspace Gradient [3] provides data management and integration with notebooks for retraining AI/ML models on the cloud.

3 Timeliness, Maturity, and Impact

The traditional workflow adopted by scientists gathering and analyzing data at DOE facilities is to temporarily cache the data at the instrument (edge), and then transfer the data across the network (ESNet etc.) to computing centers for analysis, processing, and visualization. The DOE Report on the 5G Enabled Energy Innovation Workshop [7] identifies the need to reinvent the digital continuum linking the wireless edge to advanced scientific user facilities, data analysis, and high-performance computing (HPC). DOE facilities are rapidly embracing this new paradigm and adding edge computing, from distributed sensors in Oklahoma (ARM) to beam-lines at DOE laboratories. A new breed of data storage, management, and discovery systems are required to usher us into this new world of computing across a vast continuum of diverse systems. DOE with some of the most powerful HPC systems, advanced scientific user facilities adopting edge-computing, field proven AI@Edge systems like Waggle, and a strong history of designing, implementing and operationalizing data systems, is uniquely situated to tackle the challenges and synthesize solutions.

References

[1] Label Box. `https://labelbox.com/`. Accessed: 2021-12-13.

[2] Label Studio. `https://labelstud.io/`. Accessed: 2021-12-13.

[3] Paperspace Gradient. `https://gradient.run/`. Accessed: 2021-12-13.

[4] Sage Cyberinfrastructure. `https://github.com/sagecontinuum/sage`. Accessed: 2021-12-13.

[5] Waggle AI@Edge. `https://github.com/waggle-sensor/waggle`. Accessed: 2021-12-13.

[6] Weights & Biases. `https://wandb.ai/site`. Accessed: 2021-12-13.

[7] 5G enabled energy innovation: Advanced wireless networks for science, workshop report. Technical report, 3 2020. doi: 10.2172/1606538

[8] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne, et al. The fair guiding principles for scientific data management and stewardship. *Scientific data*, 3, 2016.

Integrated data services and workflows approaches for HEP

Saba Sehrish (ssehrish@fnal.gov, contact), Fermi National Accelerator Laboratory
Jim Kowalkowski (jbk@fnal.gov), Fermi National Accelerator Laboratory
Marc Paterno (paterno@fnal.gov), Fermi National Accelerator Laboratory
Wei-keng Liao (wkliao@northwestern.edu), Northwestern University

Topic: Data management support in HEP workflows

Challenges:

Traditional HEP workflows cannot take optimal advantage of HPC resources. These are file-based and developed to work well in grid (high throughput) computing environments. Generating large numbers (typically thousands to hundreds of thousands or more) of small files is a common practice. For example, in a relatively small test case for neutrino candidate selection, we used a sample that consisted of 2000 small files; a machine learning task using a liquid argon time-projection chamber (LArTPC) simulation sample made use of 200 thousand files. Data exchange is based on files sized to fit in archive storage or convenient for output from data acquisition systems. This does not work well with the traditional processing model of HPC. Many workflows have several steps, each of which creates a number of output files that matches the number of input files. While this model has worked for decades in the grid environment, this large number of small files-based approaches is not suited for HPC environments.

In contrast to working with many small files, we also have experience working with very few large files on HPC. Dealing with large files on HPC brings a new set of challenges. First, since HEP applications are not designed to work with large files, how to create a large file effectively from the small files? Second, knowing what data is in the file, and how to organize it for fast indexed access to allow fast access and balanced data distribution? Then, once such a large file is created, placing it where it can be readily accessed, or moving it to a different location is also not straightforward. The largest file we have worked with so far is 4TB (compressed size on disk). Another observation is that HEP data is naturally compressible - either as sparseness because of geometry or measurement data similarity. The challenge here is making this compression work well when there is extensive indexing needed to locate data quickly and transfer pieces (slices) into processing applications. In the current setting, the indexing problem is in user space but should be handled by the data management systems.

Data access patterns have changed in HEP workflows with machine learning applications becoming a significant part of data analysis. Many analyses in LHC and LArTPC experiments use deep learning for problems such as object reconstruction, identification, and calibration. Modern machine learning tools work with data models that are simple to be used in training and languages that are oriented towards productivity. For example, use of HDF5 with Python-based tools is more common in machine learning applications while other applications continue using ROOT IO and C++, which has been the traditional way in HEP.

Usually, HEP data processing phases are executed sequentially. Each phase itself can use both multithreading and multiprocessing. Each phase requires a substantial commitment of resources: possibly tape access, considerable disk storage for inputs and outputs, and much compute time. The granularity of the phases is largely determined by the magnitude of the data handling tasks. A much finer level of granularity would allow better load balancing and more efficient use of computational resources.

An example of workflow is the Exa.TrkX project [1]. Its particle track formation pipeline consists of the following tasks: 1) raw hit data preprocess into feature vectors, 2) embedded network training to identify edges, 3) GNN training to classify doublets and triplets, and 4) track labeling. The output data produced

from one task becomes the input to its subsequent task. In this example, the computational demands of individual tasks are intensive, requiring parallel processing on the DOE leadership supercomputers. Workflow tasks are often developed independently by different scientists, resulting in a disparity in data structures and file layouts being used to store the input and output data. Given the sheer amount of data flowing from one workflow task to another, the computation-bound tasks can become I/O bound. This phenomenon has been observed in many SciDAC applications. To tackle such obstacles, researchers have been developing ad-hoc I/O solutions in hope to reduce the data transition time. An example in the HEP community is PH5Concat.

Opportunity:

Considering the above-mentioned challenges, R&D in developing data services and developing workflow control to make efficient use of HPC systems to allow the complex workflows that are needed for HEP to work in the HPC environments.

Given a more flexible data management system and workflow engine, HEP reconstruction and analysis tasks could be started as soon as their required input data becomes available. R&D into better pipelining of processing tasks – rather than chunking by file – would allow a finer granularity for data access and open the door to improved processing models. Better use of resources by intelligent, active controls, which can dynamically shape the workflow may significantly reduce turnaround time for processing stages that now take months to complete using traditional distributed processing systems. The intelligent shaping of workflow can be used to optimize machine efficiency subject to many constraints, such as memory available, delays caused by startup/shutdown, access to accelerators.

Given that currently indexing needed for data location and distribution is implemented in user space, research on automated indexing i.e., how to define the indexing structure appropriate to locate and bring in data for processing will be beneficial.

Given that there are large variations in runtimes for computational steps in HEP workflows, there might be significant benefits to moving to a task-based approach, one that is specifically tuned and can adapt to changes in CPU time and power needed to process current experimental datasets and is also tightly integrated with the parallel storage systems and available interfaces. We don't have means to use the resources automatically and efficiently we have. Having tools that can deal with varying application load while giving them access to the needed data in a workflow setting can benefit us. This leads us to the scenarios where our workflows will be using a combination of CPUs and accelerators as required by different processing steps and running on the machines that will have both types of architectures available. Being able to optimally run in such a setting will be of great value.

Timeliness and maturity:

We have a relationship and collaboration with ASCR teams and have made significant progress in developing tools to partially address issues in running HEP analysis workflows on currently available HPC systems, including using tools and libraries that are made to work well in the HPC environment. With exascale machines coming online soon and given the increase of complexity and data rates of new HEP detectors that will be deployed over the next decade, now is the time to engage in further research to develop solutions for complete workflows.

[1] Xiangyang Ju, et. al. Performance of a Geometric Deep Learning Pipeline for HL-LHC Particle Tracking. Eur. Phys. J. C 81, 876 (2021).

Characterization and Modeling of HPC I/O Variability through Probabilistic and Explainable AI

Sandeep Madireddy[1] (smadireddy@anl.gov, corresponding author)
Prasanna Balaprakash[1], Michel Kinsy[2], Haryadi S. Gunawi[3]
[1]*Argonne National Laboratory,* [2]*Arizona State University,* [3]*The University of Chicago*

Challenge

Performance variability, manifested as unpredictability in application execution time, is an impediment to efficient resource management and productivity in scientific computing [1]. I/O performance is one of the most prominent underlying contributors to this execution time variability. High performance computing (HPC) storage resources are simultaneously shared by a large number of applications, and I/O behavior in those applications is often characterized by intense bursts of data access interleaved with intervals of computation. This mix of bursty, uncoordinated storage system traffic causes significant fluctuations in the I/O performance perceived by individual applications. The issue is exacerbated by growing complexity in I/O architectures that integrate more heterogeneous storage technologies in order to maximize the price/performance ratio.

The ability to model I/O performance and its variability allows for more accurate prediction of application I/O performance at runtime as well as application- and system-level optimizations to proactively mitigate performance variability. I/O performance models could therefore be leveraged to make more efficient use of the I/O subsystem, a crucial shared resource on HPC systems. However, no well-established method exists for modeling I/O variability on HPC platforms, in part because of the increasingly large-scale and complex I/O subsystem designs that such platforms employ. Several approaches have been employed to address I/O performance modeling, ranging from analytical to empirical models, which lately have focused on machine learning-based approaches. Although several researchers have developed machine learning-based I/O performance models, the critical issue of modeling performance variability still remains elusive.

Opportunity

Modeling performance variability of the heterogeneous storage systems would require a systematic assessment and characterization of the different sources of uncertainties that can effect the I/O performance as well as a mathematical framework that can accommodate the incorporation of such uncertainties to build accurate machine learning models of I/O performance. Modern probabilistic machine learning approaches, especially the Bayesian formalism provides a platform to build performance models with the aforementioned characteristics. In addition, since heterogeneity of the the storage systems can lead to complex interactions between the various sub-components of the storage stack, it is crucial to be able to interpret and understand the decision making process of the I/O performance model to understand the sources of bottlenecks and draw insights that can help build better systems in the future. Explainable AI approaches customized for the probabilistic machine learning employed for I/O performance modeling would be key to achieve this objective.

A possible mathematical framework to model I/O performance ϕ on a given platform [4] is:

$$\phi = f(\alpha, \zeta, \omega), \tag{1}$$

where α represents a set of observable parameters that describe application characteristics, ζ represents a set of observable parameters that describe file system and/or I/O characteristics (e.g., filesystem health, system configuration, node availability, etc.), and behavior (e.g., the behavior

of other applications co-located with the modelled application during its run, contention from resource sharing, etc.) and ω represents unobservable parameters that remain unchanged and/or uncontrolled. The performance modeling problem is to find a function f that models the relationship between ϕ and the parameters (α, ζ). Given the unobservable nature of ω, ϕ is treated as a (possibly multivariate) hidden random variable. The central idea behind this formulation is that for the same values of parameters in α and ζ, we can observe variability in ϕ; we attribute this variability to the hidden random variable ω and model its effect in f. Therefore, for a given input parameter values in (α, ζ), the function f should provide a prediction (as in any other typical modeling approaches) as well as *distributional information* (such as standard deviation, quantiles) that captures the variability in ϕ.

The two types of uncertainties that effect empirical models are: *Aleatoric uncertainty* due to inherent randomness in the data which may not be reduced even if more data is collected (e.g., the sensor noise or the inherent randomness of simulation data). The second type is the *epistemic uncertainty* that accounts for the uncertainty which can be explained away with more data (higher spatial/temporal granularity or more telemetry), and/or incorporating domain-informed bias into the modeling exercise. To this end, we identify three main sources of uncertainty that are important to be characterized in the context of the I/O data: *a) Data uncertainty:* Characterize the aleatory uncertainty in the sensor and telemetry data using the procedures such as [3]. This characterization can be used to inform the choice of ω, and ultimately the likelihood function employed [4]. *b) System modeling uncertainty:* This is a form of epistemic uncertainty that arises due to the interaction with the system-wide background traffic that might not have been captured due to the limitation on the diversity and granularity of the application and system-wide metrics that could influence the application performance. *c) Model-form uncertainty:* This is a form of epistemic uncertainty that arises due to the implicit bias in the modeling choices and simplifications made for predicting the application performance, as well as the parameter uncertainty inside each of this model.

This mathematical framework can be effectively modeled using modern probabilistic machine learning approaches such as the Gaussian process and Bayesian deep learning approaches, where the latter promises to scale to large datasets and feature spaces along with uncertainty quantification. After a probabilistic model of I/O performance that can explain the variability is obtained, it is also important to interpret, explain and check the scientific plausibility and consistency of the results to derive scientific findings and actionable insights. These considerations are core elements of explainable ML/AI (XAI) [5], but primarily explored for deterministic models. Developing such XAI techniques for probabilistic models and especially for I/O performance needs further research.

Timeliness: Understanding storage systems and I/O was identified as a key challenge in the 2018 ASCR Workshop on Storage Systems and I/O [1]. The report noted that storage performance measurement and interpretation lags behind computational performance measurement and interpretation despite the increasing importance of data-intensive scientific methods. With recent development in probabilistic machine learning and explainable AI [2] the time is ripe to develop and customize these techniques to model and understand I/O performance variability.

References

[1] R.Ross, et.al. Technical report, USDOE Office of Science (SC)(United States), 2019.
[2] N. Baker, et al. Technical report, USDOE Office of Science (SC), Washington, DC, 2019.
[3] M.Isakov, et.al., Proceedings of ROSS '20.
[4] S. Madireddy et.al., High Performance Computing, pages 184–204, Cham, June 2018.
[5] P. Linardatos, et. al., Entropy, 23(1):18, 2021.

Title: AI-driven Storage Resource Provisioning and Operations: Revisiting Old Assumptions and Meeting New Expectations

Authors: Valentine Anantharaj (anantharajvg@ornl.gov, Oak Ridge National Laboratory), Rafael Ferreira da Silva (silvarf@ornl.gov, Oak Ridge National Laboratory), Ali Butt (butta@cs.vt.edu, Virginia Tech), and Sarp Oral (oralhs@ornl.gov, Oak Ridge National Laboratory), Devesh Tiwari (d.tiwari@northeastern.edu, Northeastern University)

Topic: Storage-system architecture design; Utilizing AI to improve I/O patterns;

Challenge: End-to-end I/O subsystems are complex in nature, especially at large scales. We are designing I/O subsystems based on historical data and assumptions, some even decades old, which may or may not hold true for a system targeted for 4-5 years into the future [3,4,5] and is expected to have an operational life of 5 years beyond that [2]. On top of that, the user workloads and I/O patterns are now changing in unpredictable ways, especially with the advent of AI in large-scale systems and this trend will continue with the integration of edge scientific experiments and instruments. Even today, on a large enough system, multiple applications are running concurrently (e.g., large-scale complex AI workflows that inherently couple various types of tasks such as short ML inference, multi-node simulations, long-running ML model training, etc. [1]), and these different applications are generating a mixed I/O workload utilizing traditional and specialized computing hardware (e.g., GPUs, quantum, neuromorphic chips) observed by the file and storage systems. On one hand, we have a wealth of log and telemetry data coming out of a large-scale compute and I/O system from across all layers of the OS and I/O software stack and hardware components (in terms of variety, velocity, and volume), simply beyond our current capabilities to meaningfully stitch together, analyze, and take action (design or operate). On the other hand, we are not getting enough and high fidelity data in real time from applications and I/O middleware libraries. We have new opportunities to design and operate better I/O subsystems given the data we have, but we are also missing fundamental comprehension of how applications are individually utilizing a given I/O subsystem or as a collection at the system level, and therefore failing to provide actionable feedback (real time or post mortem) to them on how they should improve their I/O behaviors.

Opportunity: To mitigate these data processing challenges, we argue that we need to meaningfully and intelligently reduce and filter the data. We further argue that to effectively operate a large scale storage system using a data driven approach we need to: (1) institutionalize and limit the number of “learning points”, and (2) use the learned models to “control” and achieve certain holistic system-level targets instead of individual-application focused metrics (e.g., system throughput, system-level I/O control congestion). These learning points can be placed on certain I/O servers and routers — instead of collecting data from every single source of the system along the end-to-end I/O path — learning points act as a representative sample and limit the data that needs to be ingested [3,4]. Data collected from these learning points are then fed to the “action controllers” [3,4]. These action controllers can essentially act as “recommendation implementers” to meet certain system-level objectives via better resource allocation (e.g., I/O bandwidth allocation, checkpointing frequency [6]). For example, we envision that these action controllers are embedded into job schedulers and I/O servers and routers to selectively co-schedule application traffic. These components will leverage control-theoretic property with the AI power to ensure that AI power is being harnessed but in a controlled way. This approach will also allow us to develop robust “learning points” and “action controllers” that can rely on extensive system-level benchmarks that can periodically calibrate these “learning points” and “action controllers” with ground truth [3,4]. Unfortunately, developing a representative system-level benchmark is difficult, but having this feedback-based approach (learning points and action controllers) will help us refine the benchmarking process itself and become more useful. In some sense, the benchmarking itself will become automatic and AI-driven, where it helps us achieve target objectives better (e.g., system throughput, I/O bandwidth allocation, checkpointing frequency). We believe that such an intelligent (data and model driven) system-level benchmark will allow us to design better and more cost-effective I/O subsystems.

We also have the opportunity to design a prediction system that would leverage both logs obtained from actual systems, and data that could be obtained from simulations of the system – i.e., a digital twin that could explore unforeseeable scenarios, or how the currently available technologies would perform on novel architectures. By

combining both types of data, it is possible to develop ML models (with acceptable confidence) that could be used to (1) identify current and upcoming system bottlenecks, and then (2) infer the design of novel technologies/solutions to address these challenges.

Timeliness and Maturity: Frontier at ORNL is being deployed today, and within the next two years El Capitan at LLNL and Aurora at ANL will be deployed. All these installations have I/O subsystems, speced and designed 4-5 years ago, are tuned for writing out large volumes of data, from multiple ranks, in the shortest possible time. These requirements may be based on, say writing a dump of the entire system memory in X seconds. This may capture the state of the application in restart and/or analysis files. One of the considerations in the past has been the MTBF of large systems. These stringent performance requirements also resulted in higher procurement costs and increased operational overhead. During the same time period commercial cloud service providers have developed and refined cost-effective approaches toward operational reliability. Over the past decade leadership class systems have become relatively more stable. The emerging class of AI and data intensive applications mostly require efficient and performant data ingress and egress operations. Besides, many digital twin (DT) applications are loosely coupling simulations, analytics, inferencing, synthesis and decision-support capabilities that could take advantage of native support for complex workflows and data management. Application developers and users prefer to think about data in a much more natural way that need not necessarily be I/O-centric. In some instances, the data may need to be desegregated (from files) and then reassembled in multiple ways to support various components of the digital twin applications. The bespoke workflows involved in DT applications result in complex I/O patterns that can occur concurrently during the course of the simulations and learning/inferencing phases. The emerging application needs for data management and DT workflows would need to be supported at multiple levels in the storage hierarchy. This requires intelligent provisioning and management of storage systems.

References:

[1] Ferreira da Silva, R., Casanova, H., Chard, K., Altintas, I., Badia, R. M., Balis, B., et al., A Community Roadmap for Scientific Workflows Research and Development, in 2021 IEEE Workshop on Workflows in Support of Large-Scale Science (WORKS), 81–90, 2021.

[2] Sarp Oral, Sudharshan S. Vazhkudai, Feiyi Wang, Christopher Zimmer, et.al., 2019. End-to-end I/O portfolio for the summit supercomputing ecosystem. In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis (SC '19). Association for Computing Machinery, New York, NY, USA, Article 63, 1–14.

[3] Tirthak Patel, Suren Byna, Glenn K. Lockwood, Nicholas J. Wright, Philip Carns, Rob Ross, and Devesh Tiwari, “Uncovering Access, Reuse, and Sharing Characteristics of I/O-Intensive Files on Large-Scale Production HPC Systems” USENIX FAST 2020.

[4] Tirthak Patel, Suren Byna, Glenn K. Lockwood, and Devesh Tiwari, “Revisiting I/O Behavior in Large-Scale Storage Systems: The Expected and the Unexpected”, Supercomputing (SC) 2019.

[5] Ross, Robert, Ward, Lee, Carns, Philip, Grider, Gary, Klasky, Scott, Koziol, Quincey, et.al., “Storage Systems and Input/Output: Organizing, Storing, and Accessing Data for Scientific Discovery”. Report for the DOE ASCR Workshop on Storage Systems and I/O. [Full Workshop Report]. United States: N. p., 2018. Web.

[6] Devesh Tiwari, Saurabh Gupta; Sudharshan S. Vazhkudai “Lazy Checkpointing: Exploiting Temporal Locality in Failures to Mitigate Checkpointing Overheads on Extreme-Scale Systems”, DSN 2014.

Co-Design to Enable Trustworthy Data Lifecycles for Scientific Computing

Sean Peisert, Berkeley Lab and UC Davis, sppeisert@lbl.gov
Venkatesh Akella. UC Davis and Berkeley Lab, akella@ucdavis.edu
Jason Lowe-Power, UC Davis and Berkeley Lab, jlowepower@ucdavis.edu

Topic: Secure architecture for metadata and provenance to support FAIR principles

Challenge Our main research question is *how to co-design hardware and software security mechanisms to allow scientific computing operators to enforce and ensure trustworthy data pipelines to provide guarantees and evidence, in the form of provenance, about the integrity and confidentiality of data analyzed or created.*

Data useful to science is at risk in the same way as any other kind of data. This is particular true when the data contains sensitivities of some kind and there exists the risk not only of tampering or ransomware but also that of exposure. At the largest scale, scientific computing is characterized by massive datasets, distributed, international collaborations, and HPC centers such as those sponsored by DOE SC/ASCR. The security problem can be particularly acute for HPC centers because such centers host and process data at the largest scale and therefore assume commensurate risk [6] This problem is exacerbated when the data pipeline extends outside the perimeter of the HPC facility. Advanced wireless networking has led to a rapid proliferation of network-connected devices. In the scientific world, this includes computational systems, sensors, and control devices. For example, in DOE SC domains, this includes concerns about the integrity of climate date, confidentiality of geolocation of remote coordinates of field sensors, and exposure of proprietary synthetic biology pipelines. It even now includes closed-loop experimentation in which Earth sensors deployed on remote deployed drones are controlled in conjunction with HPC simulations over a WAN.

All this connectivity introduces new vulnerabilities with each new device connected. A key risk due to this increased vulnerability is the trustworthiness of data collected at the edge. The risk of tampering with and theft of U.S. scientific intellectual property grows exponentially outside HPC facilities because traditional physical controls are no longer available for devices deployed in the field.

Even within HPC facilities, risk to data is still significant. Despite elaborate technical and procedural security protections, traditional enclaves still require implicitly trusting system administrators, and anyone with physical access to the system containing the sensitive data, thereby increasing the risk to and liability of an institution for accepting responsibility for hosting data. This security limitation can significantly weaken the trust relationships involved in sharing data, particularly when groups are large and distributed.

Isolation mechanisms and abstractions to support secure execution have been an active area of work for decades. At the hardware level, virtual memory is widely used for process-level isolation. Trusted execution environments (TEEs) take this further with hardware-based protection mechanisms to create "secure enclaves." ***Storage systems*** also have a central role to play in the challenge of maintaining integrity and confidentiality of data and for supporting ***FAIR data principles***. Existing storage systems for scientific computing scale data do not support provenance tracking and other key properties to of strong data trustworthiness. **Provenance cannot be trusted unless the computational and storage pipeline by which it is trusted, manipulated and stored can be trustsed. New storage systems are needed that are co-designed and integrated with trustworthy computing architectures** to bridge the gap between hardware-enabled TEEs and secure storage systems.

Opportunity We must consider mechanisms for providing security guarantees as the next generation of leading-edge DOE facilities' hardware and software are designed. TEEs can be used to maintain or even increase security over traditional enclaves, at minimal cost to performance in comparison to computing over plaintext. TEEs can isolate computation, preventing even system administrators of the machine in which the computation is running from observing the computation or data being used, generated, and stored by the computation, including even from certain "physical attacks" against the computing system. Therefore, such systems are a means to significantly change trust relationships involved in secure data management.

Common commercial TEEs today include Intel's SGX and AMD's Secure Encrypted Virtualization (SEV), and the recently-announced Arm "Realms." The Linux Foundation's Confidential Computing Consortium, Microsoft Azure's Confidential Computing, AWS's Nitro Enclaves, and Google's recent "Move to Secure the Cloud From Itself" demonstrate the interest in such TEEs. In addition, there exist RISC-V-based

open-source hardware efforts such as Keystone.

However, none of these TEEs have not yet been developed that target scientific computing and are appropriate for the performance requirements and vendor and protocol-specific hardware and software stacks used in HPC. Our own empirical evaluation of commercial TEEs under typical HPC workloads show results [2] that Intel's SGX has fundamental performance limits, and while AMD's SEV has minimal performance degradation on single-node operation, low-latency communication between SEV nodes and with secure HPC storage is currently impossible, making most scientific computing also impossible. Current hardware TEEs and the environments surrounding TEEs, including storage, are designed for either client and IoT devices or cloud systems; whereas HPC systems have different system constraints which should be exploited to co-design a higher-performance and easier-to-use secure environment.

An entirely new TEE architecture tailored for scientific computing is needed, which is our aim. Further, RISC-V provides the opportunity to co-design and demonstrate alternative TEE concepts that overcome the limits of current practice to meet DOE scientific computing needs, including broadening the scope of processors that contain TEEs and also ***specific co-design and integration with next-generation, trusted storage systems***. Our own work in porting Keystone to the gem5 architectural simulator [4] demonstrates the value of the ability to explore new architectural design spaces [1]. RISC-V is also open source and possible to formally verify. We aim [5] to develop approaches to addressing shortcomings of existing TEEs for scientific computing as natural extensions to the way that data is secured in scientific computing environments, and leveraging a hardware/software co-design effort to accomplish, because solutions will clearly require modifications to comute and storage architectures, operating systems, and libraries.

More specifically, we argue that the dichotomy between the usage model (software view) and the implementation on today's hardware architectures (hardware view) is the fundamental obstacle to designing secure HPC systems that needs to be overcome. Software has a single (unified) view of data with an understanding of what is sensitive, whereas a hardware implementation of an application results in data distributed across multiple, fine-grained "silos" in the form of cores, memory, communication, I/O, and storage subsystems. Enforcing *isolation*, the core functionality of a TEE, involves restricting the ability to share with a combination of hardware/software mechanisms such as physical memory protection registers, security monitors, different "modes" of operation, etc.. This bottom-up approach is problematic when an application runs across multiple nodes (especially accelerators) and third-party network, I/O, and storage subsystems.

Our insight is that HPC applications do not benefit by fine-grain resource sharing via time-multiplexing that is offered by today's hardware and OSes. We envision a *data-centric* approach to secure HPC that is based on co-designing the hardware, software, key exchange and attestation protocols around *data enclaves for scientific computing (DESC)*, that delineate data sharing boundaries in memory and storage. Our key idea is to replace fine-grain software compartmentalization with an alignment of architectures around a data-centric view, or how data moves through the system and enforces checks.

Timeliness or Maturity Scientific computing operators have made it clear that there is a need for enhancing data trustworthiness to more robustly support FAIR data principles. Current technical and procedural approaches are functional but leave large gaps both in security and usability. TEEs represent a valuable solution for enabling trustworthy computation and storage without trusting system administrators. Commercial TEEs exist and are used in the cloud but have significant performance limitations for scientific computing. Open-source hardware, such as the RISC-V-based Keystone represents an opportunity to design and build new solutions specific to HPC needs.

References

[1] A. Akram, V. Akella, S. Peisert, and J. Lowe-Power. Enabling Design Space Exploration for RISC-V Secure Compute Environments. In *5th Workshop on Computer Architecture Research with RISC-V (CARRV)*, 2021.

[2] A. Akram, A. Giannakou, V. Akella, J. Lowe-Power, and S. Peisert. Performance Analysis of Scientific Computing Workloads on Trusted Execution Environments. In *IEEE Int'l Parallel & Distributed Processing Symp.*, 2021.

[3] J. Lowe-Power, *et al.* The gem5 simulator: Version 20.0+. *arXiv preprint arXiv:2007.03152*, 2020.

[4] S. Peisert. Trustworthy Scientific Computing. *Communications of the ACM (CACM)*, 64(5), May 2021.

[5] S. Peisert *et al.* ASCR Cybersecurity for Scientific Computing Integrity. Technical Report LBNL-6953E, U.S. Department of Energy Office of Science report, February 2015.

Boosting Scientific Data Access with Usage-Driven Lossy Compression

Sheng Di (Argonne National Laboratory, sdi@anl.gov)

Topic: Storage-system architecture design, Boosting data access and management with compression

Background. Extremely large volumes of data are generated by today's exa-scale scientific applications or advanced instruments, bringing out unprecedented challenges to scientific data management and storage. As such, data reduction turns out to be indispensable for such a data explosion issue. Lossless compression, however, suffers from very low compression ratios (less than 2:1 in most cases) for scientific datasets [1]. By comparison, error-bounded lossy compression is arguably the most promising solution since not only can it significantly reduce the scientific data volumes but it can also respect the data fidelity according to user-defined error controls. Although error-bounded lossy compression has been very effective as verified by many recent studies, *how to combine/integrate this technique in scientific data management and storage efficiently to adapt diverse scientific use-cases was rarely studied*.

Figure 1 demonstrates a typical software stack of data management, representation and storage which includes compression techniques. According to a handful of existing studies [2,3,4], the data compression technique needs to be positioned between the data management layer and the virtual file layer. It plays a critical role for optimizing the overall performance, since it can be called by all upper layers and its compression performance and quality (such as compression ratio) may also considerably affect the efficiency of all lower layers such as parallel data storage.

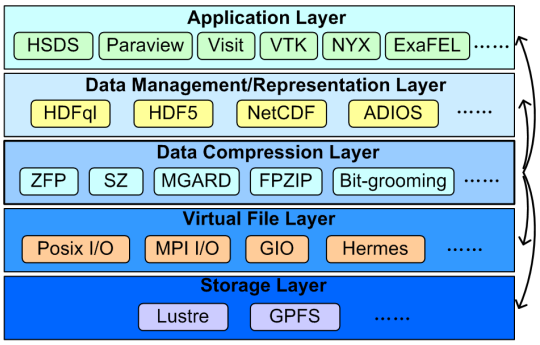


Fig 1. Software Stack of Compression-supported Data Management, Representation and Storage.

State of the arts. A few data management or storage packages/libraries have integrated various lossy compressors. HDF5, for example, leverages the H5Z filter [3] to call third-party compression libraries (such as SZ [5], ZFP [6]) inside its hierarchical data format. pNetCDF provides a utility package [4] to compress the dataset stored in the netcdf format upon user's compression requirement. On the other hand, some prior studies [2,7] have validated that the error-bounded lossy compression can significantly improve the data reading/writing performance if the compressors can be selected appropriately and the compression parameters can be tuned well.

Gap analysis and challenges. The existing state-of-the-art data management/storage systems, however, cannot fully leverage lossy compressor's performance and functionalities, introducing a significant gap to the best performance and quality. (1) *Simply calling lossy compression API (as what the existing data management/storage software did) ignores compressor's characteristics, inevitably causing unexpected compression performance or quality.* H5Z filter, for instance, executes the compression in the unit of the field/dataset, so that different chunks of one dataset are limited to the uniform configurations/settings. Another example is that some lossy compressors such as SZ need a specific configuration adjustment when being used on small datasets or chunks, whereas the existing data management/storage is completely unaware of this characteristic. (2) *Various compressors have particular pros and cons depending on different use-cases and diverse datasets, which leaves users a big trouble to determine the best compressors and appropriate settings.* For instance, in-situ data access (such as real-time visualization) requires high parallel decompression speed as decompression time is often a bottleneck compared with other high-speed components; while the online data access web service such as HSDS [8] may need high compression ratios considering the relatively low network bandwidth on WAN. (3) *Some key functionalities offered by lossy compressors require specific integration with data management/storage systems.* For example, progressive lossy compression allows users to reconstruct

data in a progressive way according to different levels of precision. This progressive compression technique generally involves multiple loosely-coupled data representation layers [9], each of which may also involve multiple blocks/chunks for the purpose of random access. As such, how to manage, index and query the corresponding data chunks efficiently is a significant gap. (4) *When all the gaps/issues are combined together, boosting the usage-driven data access performance would turn out to be extremely non-trivial, as illustrated in Figure 2, based on a random access progressive compression example.*

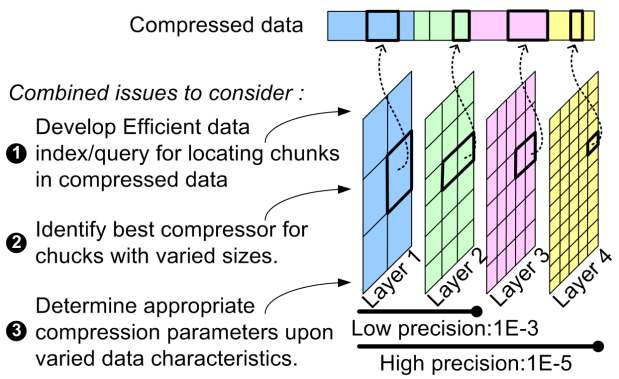


Fig 2. Illustration of Mixed Issues via Progressive Compression

Research opportunities. In order to maximize the lossy compression performance and quality in the data management, representation and storage system, a joint effort between data management/storage experts and lossy compression developers is highly needed. Specific new research opportunities are summarized as follows (but not limited): (1) *How to enable data management/storage layer to suit different use-cases at runtime.* This requires an in-depth investigation of diverse use-cases with domain scientists and flexible/user-friendly interfaces for data access with lossy compression needs. (2) *How to enable the data management/storage layer to be aware of various lossy compressors' characteristics, which may boost the data access performance in turn.* This requires a comprehensive study of lossy compressors' characteristics and their impact to the data access performance. Machine learning or AI techniques will likely be needed. (3) *How to efficiently utilize advanced lossy compression functionalities such as random access and progressive compression in data management/storage*. This requires in-depth understanding of the design and implementation of these advanced functionalities as well as the data management/storage systems. This direction is also well-consistent with the FAIR principles, in that random data access involves *findability* and progressive compression projects the *interoperability*.

Timeliness and potential scientific impact. Error-bounded lossy compression is a very timely technique to resolve the scientific data explosion issue for exa-scale applications, data management and storage systems. How to leverage lossy compression to offer efficient data access and storage service for upper level applications is an open problem which is critical to a wide range of users with diverse use-cases. (1) With use-case aware lossy compression, the data systems can offer significantly improved data storage/access performance for different use-cases adaptively. (2) With compression-characteristic aware design, the system can maximize the capability of lossy compression techniques automatically. (3) With dedicated design and optimization for new advanced lossy compression functionalities, the users/applications/services can access the data more efficiently upon their sophisticated demands (e.g., in-situ data processing/visualization and memory-limited data reconstruction from compressed datasets).

References.

[1] P. Ratanaworabhan, et al. "Fast lossless compression of scientific floating-point data", IEEE DCC06.
[2] X. Liang, et al. "Improving Performance of Data Dumping with Lossy Compression for Scientific Data", Cluster19.
[3] "H5Z: Filter and Compression Interface", https://support.hdfgroup.org/HDF5/doc1.8/RM/RM_H5Z.html
[4] pNetCDF-SZ: https://github.com/Parallel-NetCDF/PnetCDF-SZ
[5] SZ: http://szcompressor.org
[6] ZFP: https://github.com/LLNL/zfp
[7] X. Liang, et al. "Error-Controlled Lossy Compression Optimized for High Compression Ratios", IEEE Bigdata18.
[8] Highly Scalable Data Service (HSDS): https://www.hdfgroup.org/solutions/highly-scalable-data-service-hsds/
[9] X. Liang, et al. "Error-controlled, Progressive, and Adaptable Retrieval of Scientific Data with Multilevel Decomposition", IEEE/ACM SC19.

Title: Enhancing the Performance of Data Management Systems by Closing the Control Loop.

Authors: Vincent Garonne*<vincent.garonne@bnl.gov>, Alexandr Zaytsev*<alezayt@bnl.gov>,
Eric Lancon*<elancon@bnl.gov>, Jerome Lauret*<jeromel@bnl.gov>,
Shigeki Misawa*<misawa@bnl.gov>. Tejas Rao*<raot@bnl.gov>
Brookhaven National Laboratory

Topic:
Optimizing the flow of data in complex workflows by more precisely controlling the flow of information from source to destination.

Challenge:
One of the many responsibilities of a data management system is to allow data to be at the right place at the right time. This task is made more complex as this data movement typically involves orchestrating a disparate array of storage systems, data transfer systems, caching layers, and transfer protocols, each with their own set of unique capabilities and limitations. Given the heterogeneity of systems involved, data management systems tend to be loosely coupled to the component systems in each active data pipeline and have limited visibility into the state of these components (e.g., overloaded, underutilized, stalled). The net result of this situation is that the data management system has limited ability to determine whether the system is running at peak capability and efficiency or is underutilized or operating inefficiently.

On the flip side, each individual component in the data flow pipeline typically operates with little to no information on the overall state of each active data pipeline and more surprisingly, limited visibility in the next system in the pipeline with which it is interacting. To compensate for this limited visibility, the data management system and each component in the pipeline typical run with limits, thresholds, timeouts and heuristics, be they on capacity (bandwidth, storage), resources (tape drives), connections (active transfers), cache lifetimes, and open connection timers, among others. If these controls fail to maintain smooth operations, time outs expire, transfers stall or never complete, or data gets flushed and must be re-acquired from a previous stage of the pipeline. These all lead to inefficient utilization or over provisioning of resources, less than optimal performance, or outright operational failure until the root cause can be determined and operational parameters modified so that the problem does not reoccur.

Opportunity:
As is apparent from the previous section, there are numerous paths to increasing the ability of data management systems to get data where it needs to be more efficiently. Design discussion will clearly be necessary in order to determine the priority for each path and the solution(s) to be pursued. Implementation of these changes will require close cooperation between researchers generating and consuming data, developers of the data management (and possibly the workflow management) systems, storage system developers, “middle ware” developers, and facility operations staff, as changes will need to be made in multiple systems.

Timeliness:

The convergence of two trends in scientific research has created an environment that will spur the development and adoption of I/O optimizing data and workflow management systems. They are:

1. Increased use of data and workflow management systems by researchers in multiple fields
2. The realization that the current status quo in storage systems, hardware technologies, and data management will not cost effectively meet the needs of next generation scientific experiments.

With the increased deployment of data and workflow management systems, the limitations of the current state in the control of the overall pipelines are becoming apparent to more groups. This operational experience will motivate improvements in visibility and control at each stage of the pipeline. The second trend provides both the “carrot” and the “stick” that will motivate the implementation of better controls, as they will reduce operational problems and increase the ability to move data around.

Title: Extending the Usable Range of Tape Systems Beyond Cold Archives

Authors: Vincent Garonne*<vincent.garonne@bnl.gov>, Alexandr Zaytsev*<alezayt@bnl.gov>,
Eric Lancon*<elancon@bnl.gov>, Jerome Lauret*<jeromel@bnl.gov>,
Shigeki Misawa*<misawa@bnl.gov>. Tejas Rao*<raot@bnl.gov>
Brookhaven National Laboratory

Topic: Enhancing data management systems to improve the performance of magnetic tape systems.

Challenge: Multiple generations of magnetic tape technology have satisfied the cold (infrequently or never accessed files) archives and warm (files likely to be accessed at least once a year) data storage requirements for scientific data storage over several decades. However, poor random access performance is making it less suitable for warm data storage as time moves forward. In addition, the increase in the number of research groups requiring warm archive service is exacerbating the problem by increasing the randomness of data on tape and access to data. This is occurring at a time where proportionally larger warm data storage is needed and where the projected rate of change in the cost of magnetic disk storage is preventing it from satisfying the needs of warm storage capacity by itself.

Cost effective use of tape in a warm storage environment requires utilizing the full performance (bandwidth) of the tape drives, which is particularly challenging when reading data from tape. To fully utilize tape drives, the time spent mounting tapes and positioning tape must be small relative to the time spent reading data. With files as the fundamental unit of data access, achieving the goal of extracting full tape drive performance is not possible as file sizes are too small to keep time spent reading data significantly greater than the amount of time spent mounting and positioning a tape. (In order to extract 90% of the performance of current technology tape drives, more than 500 GB of contiguously laid out data would need to be read per tape mount, assuming no time is spent positioning tape.)

Opportunity: The key to increasing the effective utilization of tape systems is to change the unit of data management from files to file aggregates (e.g., dataset) of the appropriate size. If data were written and read to/from a tape in quanta of datasets, the time spent reading data from tape could be made larger than the time spent mounting the tape. If all files in a dataset were written sequentially on tape, time spent positioning the tape would be minimized when the dataset is read back.

Achieving the goal of maximizing tape drive utilization will require effort on multiple fronts. First, researchers must provide input on the access requirements and access “relationships” among the files being stored. Information on what files will be accessed at the same time would need to be captured. Data management systems would need to ensure that data in a given aggregate are faithfully stored on a minimum number of tapes and that read requests for aggregate are transmitted intact to the tape system for efficient recall.

Timeliness: Access to and storage of data has always been needed in scientific research. However, the proliferation of “big data” experiments, the increases in data volumes collected by next generation experiments and the push towards FAIR access to research data have brought light to the fact that better data management and utilization of storage resources is necessary to keep scientific research moving forward. Increased recognition of this fact is key to getting the necessary resources and interests aligned to make progress in this area.

Title: Rethinking Warm Data Storage

Authors: Vincent Garonne*<vincent.garonne@bnl.gov>, Alexandr Zaytsev*<alezayt@bnl.gov>,
Eric Lancon*<elancon@bnl.gov>, Jerome Lauret*<jeromel@bnl.gov>,
Shigeki Misawa*<misawa@bnl.gov>. Tejas Rao*<raot@bnl.gov>
Brookhaven National Laboratory

Topic: Utilizing the SCSI/NVMe Zoned Storage model to make SMR disks viable for scale out nearline/warm storage.

Challenge: Long term projections for disk storage systems show that the cost of these systems will not drop fast enough to keep up with the growth in data volumes in scientific research. Furthermore, while disk capacity continues to increase, improvements in Input/Output operations per second (IOPS) have basically stalled. Keeping disk systems viable as nearline/warm storage will become more difficult over time.

Opportunity: Theoretically, SMR (Shingled Magnetic Recording) disk drives can provide up to 25% higher capacity compared to “conventional” magnetic recording (CMR) drives, be they PMR (Perpendicular), MAMR (Microwave Assisted), or HAMR (Heat Assisted). The downside of SMR drives is that write latencies can be significantly higher than CMR drives, making them unusable in standard disk based storage architectures, e.g., HW/SW RAID arrays, standard disk file systems. However, as SMR and CMR read performance are comparable, if SMR write problems can be mitigated, storage costs can be reduced. Furthermore, if IOPS can be reduced for reads through changes in data access models and data layouts on disk, the limits in IOPS can also be mitigated.

Mitigating the write and IOPS problems might be achieved by building disk storage systems that work more like tape systems, particularly with regards to data placement and write operations. The problems associated with SMR write latencies can be eliminated by writing large blocks sequentially to disk. If properly executed, write performance is not hampered by the shingle nature of the media. This can be achieved by using the Zoned Storage tools that have been developed for SMR disks and Flash memory. Also, if a file is written to a single extent, IOPs needed to read the data are likely to be reduced compared to a block based system. Use of an object based access paradigm can be used to enable the necessary large block sequential writes and reads to disk.

Three potential side benefits for a SMR warm storage system are:

1. Increased performance of any archival tape storage systems that might back end the warm storage. Through the use of an object storage paradigm, the “bucket” could be used as the unit of data that is pushed back to tape, resulting in larger contiguous blocks of data on tape that are likely to be read back at the same time. This would result in more efficient utilization of tape drives, as more data would be read back from a tape, reducing the overhead associated with mounting a tape, and minimal time would be spent seeking to files as the data would be contiguous on tape.

2. Provide a better warm storage layer than would be possible from a tape based system, as seek latencies for the disk system would be substantially lower and access bandwidth scales with storage capacity.
3. Integrate more effectively with flash based caching systems or storage pools that might sit in front of an SMR warm storage system, compared to standard disk storage or nearline tape storage system, for applications requiring lower latency I/O. These flash systems can also be used as staging areas where data can be aggregated into large multi-object or multi-file containers to be colocated on the SMR disk system for better read performance.

Timeliness:

Next generation scientific experiments in multiple fields are expected to generate an order of magnitude more data than current experiments. Extrapolation of current technology trends suggest that current disk based storage systems will not be affordable at the scale that is required if existing usage profiles remain in place. Little to no change in IOPs per disk has resulted in a steady drop in IOPs/TB, increasing the likelihood that there will be issues utilizing disk in the future in the same way they are used now. Multi-actuator drives can mitigate the IOP problem, but at a cost of higher device complexity (and hence cost) and power consumption. These issues are strong motivators for investigation into alternative system architectures, increasing the likelihood that a more suitable solution can be developed and be adopted sufficiently to give the technology some longevity.

Investigation into systems using zoned storage is not isolated to SMR disk drives. Similar techniques are being advanced for flash memory systems, for the same reasons. This is likely to lead to a larger pool of talent knowledgeable with zone based storage, increasing the long term viability of zoned storage solutions.

At this point in time, the basic tools needed to build storage systems based on zoned storage are available in the form of SCSI Zoned Block Commands (ZBC) and Zoned ATA Commands (ZAC) and support for zoned devices is in the Linux kernel. The equivalent API's for Flash are the Zoned Namespace command set in NVMe. These are key enablers for widespread deployment of zone based storage.

References:

G. Gibson and G. Ganger, "Shingled Magnetic Recording for Big Data Applications," Technical Report CMU-PDL-12-105, Parallel Data Lab, Carnegie Mellon University, Pittsburgh, PA, 2012.

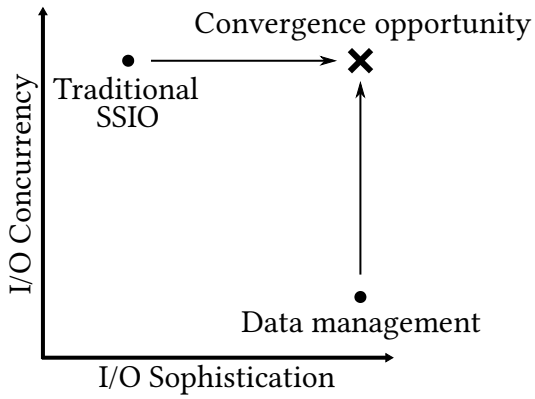
https://zonedstorage.io/ Western Digital Zoned Storage community web site.

M. Bjørling, A. Aghayev, H. Holmberg, A. Ramesh, D. Le Moal, G. R. Ganger and G. Amvrosiadis, "ZNS: Avoiding the Block Interface Tax for Flash-based SSDs", Proceedings of the 2021 USENIX Annual Technical Conference.

SSIO and data management: opportunities for convergence

Spyros Blanas, The Ohio State University

Many scientific datasets are inherently very large and highly structured. Fundamental data management principles, such as declarative querying, data indexing and query optimization have proven to be highly effective in managing petabyte-sized warehouses for structured data in enterprise settings and in the cloud. Despite promising efforts within DoE to develop state-of-the-art data management capabilities that are tailored for scientific data, high-performance scientific applications have been slow to embrace data management technologies.



This has led to a widening gap in how scientific computing and data management achieve high I/O performance. Scientific applications derive their impressive performance from fast and highly parallel I/O, but the fundamental data access pattern is naive: applications typically start by loading all the data from a remote file system to local memory. Enterprise and cloud applications achieve high I/O performance through sophisticated data access methods that carefully consider what data to load to memory and when, but target I/O devices that are many orders of magnitude slower than the typical I/O system of a high-performance computing facility. We posit that the convergence opportunity lies in combining the sophisticated I/O optimizations of data management with the I/O scale and concurrency of high-performance computing.

Finding the pathway to convergence is particularly important, because future I/O scalability is currently challenged by a confluence of paradigm-shifting trends: at the software level, the new reality of asymmetric I/O capabilities has prompted the rethinking of the roles of block-oriented versus object-oriented storage; at the hardware level, the storage stack is becoming deeper and more heterogeneous, requiring a more nuanced understanding of the storage topology; at the application level, the pursuit of AI-assisted science is imposing unforeseen demands on I/O scalability. This presents a unique opportunity for novel research across three thrusts:

1. **Data layout optimization.** A promising research avenue lies in investigating chunking strategies that are more closely tailored to the application access pattern. The data management literature has recently introduced hierarchical partitioning and irregular partitioning strategies for tabular data as a way to minimize I/O for applications with data-dependent accesses. Extend these ideas to multi-dimensional datasets that are common in scientific applications is not trivial and the I/O gains can be even more significant as the dimensionality of the data increases. In particular, more research is needed to understand how to synergistically use both block-based and object-based data stores from applications. Additional optimization opportunities lie in using AI, specifically leveraging unsupervised learning techniques to discover novel access patterns without developer involvement.

2. **Topology-cognizant data placement.** Many scientific applications were developed with the assumption of homogeneous storage: the file system spreads the data in different storage devices, but all devices are equally "far" from every CPU in the cluster. This assumption is challenged by the increasingly heterogeneous nature of compute and storage, that spans from the edge, to the HPC center and to the cloud. Currently theoretical models, algorithms and systems assume a uniform topology; this assumption rarely holds in practice. Future research needs to recognize the need to track the distribution of data across storage devices in a fine-grained manner and systematically consider the impact of the underlying network topology. This necessitates an end-to-end investigation of how one can model, design and deploy topology-aware algorithms for fundamental data processing tasks at large scale.
3. **Near-data AI.** The nascent area of near-data processing is investigating novel ways to bring limited forms of computation, such as computing an aggregate, as close to the physical location of the data. Promising prior results have considered processing-in-memory, namely how one perform limited computation inside the DRAM array or how can some DRAM cells be replaced with simple processing units, and processing-near-flash, where simple computation can be executed at the controller of a flash-based storage device. Further research on how to bring the early stages of ML pipelines closer to where the data is stored would be of particular relevance to data-intensive scientific computing.

Knowledge Graphs for FAIR Data and their Empowerment of Digital Twins Development

Stuart Chalk[1], Dylan Johnson[2], Marshall McDonnell[3], and Jon Fortney[4]

[1]schalk@unf.edu, Department of Chemistry, University of North Florida
[2]n01448636@unf.edu, Department of Chemistry, University of North Florida
[3]mcdonnellmt@ornl.gov, Computer Science and Mathematics Division, Oak Ridge National Laboratory
[4]fortneyjm@ornl.gov, Computer Science and Mathematics Division, Oak Ridge National Laboratory

TOPIC: Metadata management infrastructure to support FAIR principles

I. Challenge

Digital data management has been around for decades. Yet, integrated, connected data that is discoverable and reusable has eluded the largely distributed and heterogeneous scientific data resources of today. This is because it is not just a technological problem. The problem lies in the way scientific data is treated and modeled.

There has been a culture change with ”open data” and the Findable, Accessible, Interoperable, and Reusable (FAIR) data principles to help promote sharing data in a useful way. ”Open data” [1], [2] is the concept that data is made available from scientific studies, whether they are published openly before or after the study is complete. Yet, simply making data open does not provide connections and integrations between the data. Mainly, as pointed out in the original FAIR data principles paper in the context of open data, ”...the data ecosystem, therefore, appears to be moving away from centralization, is becoming more diverse, and less integrated, thereby exacerbating the discovery and re-usability problem for both human and computational stakeholders.” [3] As valuable scientific data is collect in this manner, independently managed “silos” of data are created with very little interaction, connections, or knowledge gained. This compromises the full potential of scientific data, surprisingly, in the age of the digital revolution.

Yet, there is a burden of accurate metadata for FAIR data that requires rethinking how we collect data. FAIR data management is new and requires additional metadata be recorded. Semantic meanings are inherent in our research data. Semantic scientific metadata needs to be captured “at birth” (i.e. from experiments and computations) and as automated as possible.

An ontology is an explicit specification of a conceptualization contextualizing the data and what is used to define the semantic meaning in our research data. [4] A difficult task is determining what ontologies already exist that support a domain and if they capture all of the semantic definitions required of our datasets. Either the standardization of a single ontology or the integration of ontologies between different domains (that connects cross-cutting, interdisciplinary research) is an even bigger challenge.

There are current challenges of capturing semantic metadata “at birth” to enable producing “at birth” FAIR data, how can we do this as automated as possible to lower the burdens on research teams, and long-term challenges of migration of the semantic meanings (i.e. ontologies) in connected, FAIR data.

II. Opportunity

A. Emphasis on creating scientific knowledge graphs

A knowledge graph is a directed, labeled graph where the labels have well-defined, semantic meaning of the relationships (i.e. edges) between nodes (data) of the graph. [5] A knowledge graph is a graph representing explicit connections of the relationships between data and context of the importance (i.e. semantics defined via an ontology). The significance of scientific knowledge graphs can be correlated to the importance highlighted for graph technologies in information technology radars, such as the Gartner 2020 report [6] stating: *”By 2023, graph technologies will facilitate rapid contextualization for decision making in 30% of organizations world-wide.”* and *”It helps data and analytics leaders find unknown relationships in data and review data not easily analyzed with traditional analytics.”*

This latter goal is exactly what we envision for an integrated, connected scientific data. With connected scientific data, we can use machine learning (ML) / artificial intelligence (AI) analytics to determine unknown relationships within the knowledge graph (i.e. “knowledge completion”). Thus, the knowledge graph itself can help with automating semantic annotation of the data. Also, we gain new insights by analyzing the data with ML/AI with deeper context provided by the semantics in a knowledge graph.

Also, what if the initial knowledge graph has a sub-par ontology and requires migration to another? This results in lost time, lost value, and a detriment to momentum of creating FAIR data. Yet, AI and ML analytics tools can again be developed and

used to create an "enhancement service" for the knowledge graph. Examples of current scientific ontologies are both domain-specific, such as the Open Biological and Biomedical Ontology [7], and also more general scientific ontologies, such as the SciData Ontology [8]. Given we invest in creating robust, scientific knowledge graphs using common scientific ontologies, graph ML/AI can be used to optimize data access and connections within the graph. Thus, the knowledge contained in the knowledge graph itself helps to iteratively improve the graph using analytics services. This can include providing feedback of the effectiveness of the current ontology, and visualizing gaps where expansion is needed.

B. Benefits of knowledge graphs with digital twins

One area where the knowledge graph is showing unique benefits is the creation of digital twins. Digital twins are meant to be digital clones which represent real physical systems and are designed to perform in-depth analysis offline. [9] The Internet of Things (IoT) has shown lots of growth through the use of digital twins. IoT digital twins include virtualized sensors and devices that can be used together either fully virtual or with the physical devices as well in a hybrid ecosystem.

The combination of digital twins represented by knowledge graphs has been presented recently and highlights how a knowledge graph *"...enriches the intelligent digital twin by internal linking and referencing, knowledge completion, error detection, collective reasoning and semantic querying capabilities."* [10] Thus, a knowledge graph can be used to enhance a digital twin by utilizing the full potential of the two way data flow combined with data analytics (i.e. ML and AI) to refine the digital twin representation. The approach of using digital twins is seen as a huge growth area for industry. In 2020, the digital twin market was valued at $3.1 billion and is speculated to steeply climb to $48.2 billion by 2026. [11] If knowledge graphs are continually used to enhance digital twins, a similar steep climb should occur with data management infrastructure around knowledge-graph-related data technologies. Scientific data management efforts will need to shift sooner than later to enable a cultural change around how we treat data and utilize its full potential.

III. Timeliness

A. Short-term strategy and gains

Standing up more scientific knowledge graphs, creating semantic data "from birth" at instruments and compute resources, treating data as a "first class citizen", and concentrating efforts into standardized ontologies at organizations will immediately show value for a single organization's data. Specifically, these gains are needed to capture the organization's knowledge (currently stuck in data silos). By creating data services around the knowledge graph asset, easy-to-access, connected data will be provided. This will in turn empower data analytics teams and data scientists across the organization to operate on this data in the knowledge graphs.

B. Long-term strategy and gains

As the short-term strategy is realized, long-term efforts will need to go into adaptability of the knowledge graphs for ensuring continual productivity and also different knowledge graphs through alignment (semantic equivalency) of ontological terms. Also, federation of knowledge graphs between institutions and organizations should be a priority to move toward a unification of the national and/or international scientific data enterprise. Given this vision, a federated data management infrastructure would provide what the FAIR data principles describe as an end goal:*"...more rigorous management and stewardship of these valuable digital resources, to the benefit of the entire academic community."* [3]

References

[1] P. Murray-Rust, "Open data in science," *Nature Precedings*, 2008. [Online]. Available: https://doi.org/10.1038/npre.2008.1526.1

[2] W3C, "Data on the web best practices," 2017. [Online]. Available: "https://www.w3.org/TR/dwbp/"

[3] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. da Silva Santos, P. E. Bourne *et al.*, "The fair guiding principles for scientific data management and stewardship," *Scientific data*, vol. 3, 2016. [Online]. Available: https://doi.org/10.1038/sdata.2016.18

[4] T. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," *International Journal of Human-Computer Studies*, vol. 43, 08 1994. [Online]. Available: https://doi.org/10.1006/ijhc.1995.1081

[5] V. K. Chaudhri, "Stanford university lecture for cs520: What is a knowledge graph?" 2020. [Online]. Available: "https://web.stanford.edu/class/cs520/2020/notes/What_is_a_Knowledge_Graph.html"

[6] L. Goasduff, "Gartner top 10 trends in data and analytics for 2020," 2021. [Online]. Available: https://www.gartner.com/smarterwithgartner/gartner-top-10-trends-in-data-and-analytics-for-2020

[7] B. Smith, M. Ashburner, C. Rosse, J. Bard, W. Bug, W. Ceusters, L. Goldberg, K. Eilbeck, A. Ireland, C. Mungall, N. Leontis, P. Rocca-Serra, A. Ruttenberg, S.-A. Sansone, R. Scheuermann, N. Shah, P. Whetzel, and S. Lewis, "The obo foundry: Coordinated evolution of ontologies to support biomedical data integration," *Nature biotechnology*, vol. 25, pp. 1251–5, 12 2007. [Online]. Available: https://doi.org/10.1038/nbt1346

[8] S. J. Chalk, "Scidata: a data model and ontology for semantic representation of scientific data." *Journal of Cheminformatics*, vol. 8, 2016. [Online]. Available: https://doi.org/10.1186/s13321-016-0168-9

[9] A. Banerjee, R. Dalal, S. Mittal, and K. P. Joshi, "Generating digital twin models using knowledge graphs for industrial production lines," in *Proceedings of the 2017 ACM on Web Science Conference*, ser. WebSci '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 425–430. [Online]. Available: https://doi.org/10.1145/3091478.3162383

[10] N. Sahlab, S. Kamm, T. Müller, N. Jazdi, and M. Weyrich, "Knowledge graphs as enhancers of intelligent digital twins," in *2021 4th IEEE International Conference on Industrial Cyber-Physical Systems (ICPS)*, 2021, pp. 19–24. [Online]. Available: https://doi.org/10.1109/ICPS49255.2021.9468219

[11] IBM, "IBM: What is a Digital Twin?" 2021. [Online]. Available: "https://www.ibm.com/topics/what-is-a-digital-twin

Title: Designing End-to-end HPC Data Reduction by Leveraging Smart Storage and AI Intelligence

Authors: Sudarsun Kannan (Rutgers University), Bing Xie (Oak Ridge National Labs)

1. Challenge

In this post-Moore era, high-performance computing (HPC) applications and workflows are being designed or enhanced to scale across hundreds of systems and generate massive amounts of data. Enhanced data processing capabilities with massively parallel accelerators like GPUs and FPGAs, combined with memory tiers, storage tiers, and network infrastructures [2, 7] have resulted in petabytes of data generated each day. The data increase is also driven by traditional large scale workflows such as molecular dynamics [1], and more recently, AI-driven applications, such as nuclear physics [3] and earth system models [4], which run across thousands of general-purpose cores and accelerators.

In spite of hardware advancement, the increased data volume overwhelms hardware system resources such as compute, memory, storage, and network. Consequently, software components and services, such as runtimes, OS, storage, and network stacks that manage hardware resources, spend a significant fraction of time addressing resource pressure, resulting in a tremendous slowdown in application and workflow performance. Adding more hardware is not the solution because scaling software across new hardware introduces its own set of software bottlenecks.

While reducing data has been an important focus, and several widely used approaches have been employed, in this position paper, we argue that current triggering, sampling, compression, or application-specific filtering solutions are mostly done at the host system and one dimensional. In these approaches, the data reduction is mainly performed at the application layer or the runtime and lacks end-to-end cross-layered designs that include all system software resources and hardware resources (for example, near-storage processing) or the right interface to perform near-storage processing without incurring high-overheads. In this position paper, we specifically focus on using near-storage data reduction (i.e., processing), where modern storage devices are attached with wimpy or general-purpose processors that can perform simple operations. Beyond the use of near-storage processing, we explore the I/O interface support and techniques to extend near-storage processing with the use of AI/ML techniques for identifying what and when to reduce.

2. Opportunity

We see the opportunities to combine near-storage I/O processing and AI algorithms for data reduction, emphasizing the end-to-end design and programming abstractions to make data reduction seamless and without impacting the overall workflow performance.

3. Using Smart Storage to Reduce Data Movement

Our first focus is the support for near-storage processing to reduce data (e.g., applying compression on the data). We believe reducing data and the related overheads to move data between applications and storage and additional overheads such as communication cost when moving data over the network requires expressive file system interfaces that can support near-data processing. Specifically, we focus on introducing novel I/O abstractions as an extension to POSIX and generically applicable for both local and remote near-storage solutions as well as traditional kernel file systems. We take inspiration from seminal CISC (complex instruction set computers) architectures to support aggregation of simple POSIX I/O instructions (operations) and data processing/filtering or reduction operation offloading them to a processor. Intuitively, a CISC-based I/O would allow combining I/O and data processing/filtering operation (e.g., read-modify-write, read-compress-write) to a near-storage file system, thereby significantly reducing overheads between host and (local and remote) storage devices. Beyond offloading I/O operations, CISC-I/O would also support near-storage processing.

To realize CISC-I/O, we plan to extend our cross-layered near-storage designs for HPC systems, CrossFS [5] and CompoundFS [6], orginally designed for supporting in-storage computation for data center key-value stores. As shown in Figure 1, the in-storage computation platform uses Linux Kernel that uses dedicated CPU cores and drivers to emulate the smart storage platform. Applications issue simple POSIX operations (e.g., read or write) that are modified inside the library with data reduction operations (e.g., write + compress) and dispatched for in-storage processing (i.e., compress). Additionally, applications can be given the flexibility to use their custom I/O and data processing operations. To understand the benefits of CISC-I/O for HPC applications, we will extend HDF5, MPI-IO, and POSIX libraries that are traditionally used by most HPC applications and support CISC-I/O operations. We expect first to explore general-purpose data filtering and reduction operations like data compression/decompression and then focus on triggering sampling operations.

As a preliminary study, in Figure 1.B, we show the benefits of using traditional data compression (bars showing host-reduce) that uses host CPUs to fetch and compress the data and our proposed CISC-I/O-based data compression (in-storage-reduce) that performs in-storage compression and decompression without moving data. We use a widely used HPC I/O benchmark, MADBench, that continuously generates I/O data. In the traditional approach, a set of background threads perform compression by loading I/O data constraining the I/O bandwidth from data movement. In contrast,

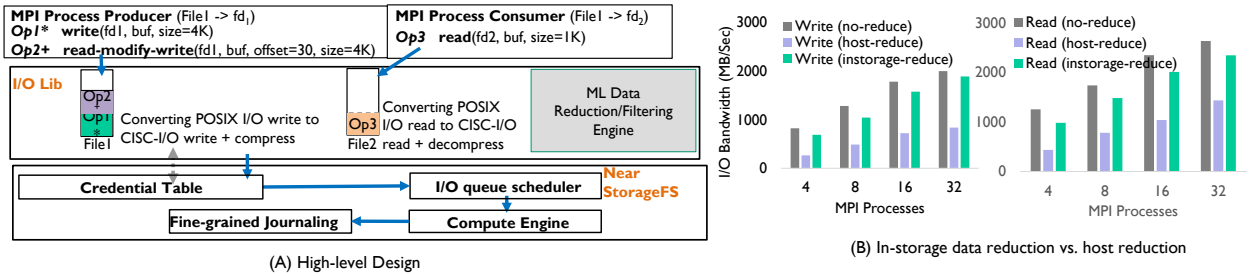


Figure 1: High-level Overview of Smart Storage + AI Data Reduction

our proposed CISC-I/O performs in-place data reduction without impacting application performance and providing comparable performance as no data reduction case (that fails to reduce data size), highlighting the overall benefits.

4. Using AI-based Techniques for Smart Data Reduction.

We next focus on employing AI-based techniques for data reduction in HPC applications instead of the application or a programmer specifying what to reduce or not reduce based on how applications use the data. While one could offload data reduction techniques to smart storage, one opportunity/challenge lies at the application level about what data reduction to apply (e.g., data filtering vs. compression vs. or other operations) that would reduce the I/O cost. Similarly, another opportunity lies in using AI/ML techniques at the application-platform level by identifying data access patterns and deciding whether data reduction is useful or bound to increase overheads (e.g., not to compress data frequently read by an application).

At DoE's experimental facilities and science instruments, we observe that applications are usually submitted as many jobs for debugging, testing, and production runs. The jobs perform the same application logic for each such application but may conduct computations at different scales, process/generate data with different volumes and benefit from different data reduction techniques and strategies. Nevertheless, these different but similar jobs provide the opportunities to 'learn' the behaviors of each application across jobs and scales. We could utilize this information to select the best techniques/strategies to process data reduction transparently, dynamically, and autonomously by estimating the overall data movement overheads and related resource and software bottlenecks and then offloading them for near-storage processing. More specifically, for applications (e.g., nuclear physics and molecular simulation), we propose to build the data-reduction context (e.g., using parameters like data sizes and locations, access patterns, etc.) as a neural network. We will consider the strategies and techniques as actions in deep neural network models to identify the best strategies and techniques via trials and errors. Beyond the application-specific data reduction techniques, we aim to develop a platform-centric solution for smart data reduction. This approach builds a deep-learning (DL) model on the context and features collected from various I/O-intensive jobs across applications. We then aim to learn the effects of different strategies/techniques on different jobs via trials and errors. Specifically, we plan to explore the cost estimation benefit to understand the reduction of smart storage.

5. Timeliness or Maturity:

By proposing a foundational approach to rethink data reduction in HPC systems by combining near-storage data processing enriched with application-explicit and transparent programming interface combined with AI-based techniques for identifying what to filter or reduce, the position paper's research ideas provide an opportunity to efficiently apply intelligent data reduction techniques to a wide range of HPC applications. We believe the ideas will pave the way for new data storage innovation considered an Achilles heel for decades.

References

[1] Gromacs: A message-passing parallel molecular dynamics implementation. *Computer Physics Communications*, 91(1).

[2] Sudarsun Kannan, Ada Gavrilovska, Karsten Schwan, and Dejan Milojicic. Optimizing checkpoints using nvm as virtual memory. In *Proceedings of the 2013 IEEE 27th International Symposium on Parallel and Distributed Processing*, page 29–40, USA, 2013. IEEE Computer Society.

[3] Oak Ridge National Laboratory. Nuclear Structure and Nuclear Reactions, 2021. https://www.olcf.ornl.gov/caar/summit-caar/nuccor/.

[4] Anikesh Pal, Salil Mahajan, and Matthew R Norman. Using deep neural networks as cost-effective surrogate models for super-parameterized E3SM radiative transfer. *Geophysical Research Letters*, 46(11):6069–6079, 2019.

[5] Yujie Ren, Changwoo Min, and Sudarsun Kannan. Crossfs: A cross-layered direct-access file system. In *14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)*, 2020.

[6] Yujie Ren, Jian Zhang, and Sudarsun Kannan. CompoundFS: Compounding I/O operations in firmware file systems. In *HotStorage '20*. USENIX Association, July.

[7] Jeffrey S. Vetter. Preparing for extreme heterogeneity in high performance computing. In *Design, Automation & Test in Europe Conference & Exhibition, DATE 2019, Florence, Italy, March 25-29, 2019*. IEEE, 2019.

- **Title:** Challenges of data management with traditional storage systems.
- **Authors:** Tejas Rao*<raot@bnl.gov>, Vincent Garonne*<vincent.garonne@bnl.gov>, Alexandr Zaytsev*<alezayt@bnl.gov>, Eric Lancon*<elancon@bnl.gov>, Jerome Lauret*<jeromel@bnl.gov>, Shigeki Misawa*<misawa@bnl.gov>.

  * Brookhaven National Laboratory
- **Topic:** Understanding the overlap between traditional storage systems and I/O (SSIO) efforts and data management.
- **Challenge**:

Data storage management ensures that data is available to the users when they need it. Data management is an extensive topic and can include various strategies related to data retention, storage tiers, consolidation of systems/protocols, backup and disaster recovery. With vast amounts of data produced every day, traditional storage systems like SAN and NAS platforms and distributed file systems are being pushed to their limits which hamper scalability. Scientific data is usually semi-structured data which could contain tags, markers or some other method of organizing the data. Traditional file systems are primarily designed for large file transfers and not for quick metadata querying while we observe on home directory storage and scientific semi-structured distributed storage that metadata access is much more frequent than file transfers. There is a need for quick search, storage and retrieval of data present in different file formats. Atomicity, Consistency, Isolation, Durability (ACID) compliant systems and the strategy around them are still important for many scientific applications however ACID compliant storage systems were not designed from the ground up to address several of today's data challenges. The complexity, cost and required performance of using these traditional systems to address these new data challenges would be extremely high.

With effective data management, storage systems perform well across geographic areas. It ensures data security, protects against system failures and preserves data integrity. Proper backup and disaster recovery are important aspects of a data protection strategy.

- Metadata management:

  Metadata generally refers to information about data. It could include traditional user visible information like who owns the data, file attributes, permissions, extended file attributes or custom user defined attributes. Most traditional storage systems are based on a hierarchical namespace storage management. Such hierarchical namespaces are often viewed as inherent limitations on concurrency and usability.

- Storage security:

  Challenges of data storage management include persistent cyberthreats, data management regulations and a distributed workforce. These challenges illustrate why it's so important to implement a comprehensive plan: A storage management strategy should ensure organizations protect their data against data breaches, ransomware and other malware attacks; lack of compliance could lead to hefty fines; and remote workers must know they will have access to files and applications just as they would if in a traditional office environment.

- **Opportunity:**

Develop methods to store science metadata in a scalable manner, and in a standardized and productive way. One possible mechanism would be to decouple metadata storage from the underlying file system, so that they may evolve independently while preserving consistency.

Traditional storage systems rely heavily on Schema-On-Write models which require that data must be validated when it is written and must conform to ACID semantics, instead switching to a loose model of Schema-On-Reads will improve performance and reduce complexity considerably.

Adopting an object storage framework architecture solves multiple problems including data security, complexity, cost and data retention. Adopting such a higher-level model could also be a fundamental breakthrough needed to scale namespace to high concurrency levels.

- **Timeliness**:

SSIO systems are becoming increasingly complex and hierarchical. Organizations today contain large volumes of information that is not actionable or being leveraged for the information it contains. It is important to analyze and correlate large amounts of data quickly and efficiently and with traditional storage systems and its limitations this is not possible. It is important to be able to correlate semi-structured and unstructured data with existing structure.

Storage Abstractions for Data Movement and Interoperability Between SSIO and Workflow Systems

Authors. Tom Peterka (ANL), Dmitriy Morozov (LBNL)
Topics. Data management support for AI and complex workflows; overlap between traditional SSIO efforts and data management.

Challenge. A dichotomy between software for data movement in SSIO and for data movement in scientific workflows exists today. SSIO software consists of mature libraries such as NetCDF, pNetCDF, and HDF5 that many computational science communities use in application domains, either directly or through high-level libraries such as HighFive. In contrast, workflow data movement software such as Conduit, Decaf, or ADIOS is usually custom built as part of a workflow system and not used elsewhere.

Opportunity. From a user's perspective, data movement in SSIO and data movement in workflows appear to have similar objectives: to move data in parallel, efficiently, with minimal user intervention. The only difference is the destination: a storage target in SSIO or another task in a workflow. Such similarity offers an opportunity for interoperability between data flow in SSIO and in workflow systems, an opportunity that can and should be addressed in the following three ways. **(1) SSIO researchers should develop data movement abstractions for their libraries**, analogous to the HDF5 virtual object layer (VOL) for HDF5. **(2) Workflow researchers should develop data movement libraries for workflow tools on top of the SSIO abstraction layers**, customizing data movement for particular cases specific to workflows on HPC systems. **(3) SSIO and workflow researchers should co-design (1) and (2) to minimize redundant research and maximize common functionality between SSIO and workflows**.

State of the Art. There are at least two approaches to increasing interoperability between software libraries: (de)composition and abstraction. Decomposing monolithic software stacks into smaller pieces and re-assembling them in modular ways is the approach taken by the Mochi project and advocated in the "composable ISDM" PRD of the report on the DOE workshop on In Situ Data Management. The alternative approach presented in this document is to provide abstraction layers (middleware) accommodating different uses of a software library. This is the approach taken by HDF5 VOL and DataElevator, which applies HDF5 VOL to staging burst buffers. We argue that interoperability between SSIO and HPC workflows can be achieved by applying analogous data movement abstractions to in situ data movement. Composability and abstraction can also be combined, by composing interoperable abstraction layers. For example, we are currently experimenting with climate codes issuing NetCDF-4 calls, which are mapped to HDF5, and then capturing those HDF5 calls with HDF5 VOL for the purpose of redirecting data in situ to tasks in a workflow.

Timing. The costs of continuing to ignore the potential synergy in SSIO and workflow software systems are lost performance, reduced usability, and duplication of effort on the part of computer scientists and end users. Legacy SSIO libraries such as HDF5 are now starting to support virtual object layers, and developers are beginning to write abstractions on top of those capabilities. With new storage services such as Mochi and Daos beginning to emerge, now is the time for R&D of similar virtual abstractions for those services. Doing this later as an afterthought is much more difficult than designing such capability into the SSIO service from the outset. Now is the time to take advantage of this opportunity and co-design such abstraction layers together with experts from the ASCR HPC workflow community.

New Research Directions. New research is needed to provide abstraction layers so that the same user interface can direct data movement between workflow tasks as well as accessing storage. The functionality listed below can come from an SSIO abstraction, from workflow plugins on top of the SSIO abstraction, or in an ideal scenario be co-designed by both SSIO and workflow teams working together.

- Redistributing data between different numbers of producer / consumer tasks with varying resources (e.g., number of MPI processes) between tasks.
- Interchanging physical file storage with in-memory or in-network communication per data object, per pair of tasks in a workflow.
- Support for deep and shallow copies of data, configurable per data object, per workflow task pair.
- Support for distributed-memory and shared-memory data communication. The latter requires thread safety of abstraction layers and underlying libraries.
- Abstraction layers in modern C++, that are well-documented and actively supported.
- Composability of abstraction layers for even greater interoperability.

Potential Scientific Benefit. The above research would require several years of initial investigation funded by ASCR research, followed by sustaining investments through programs such as SciDAC in order to increase technological readiness to the level of production usage. Short-term metrics for success after 3 years would include prototype development of one or more abstraction layers and their demonstrated use in scientific workflows in addition to traditional SSIO use cases. Long-term success would be measured by usage in DOE science applications by end-users. The beneficiaries of the proposed research direction are computer scientists and domain scientists alike. Computer scientists would gain expertise in developing interoperable abstraction layers, and would save time by reusing and composing software rather than re-inventing it. Domain scientists would minimize changes to their codes in order to switch seamlessly between data movement destinations, using a consistent data model for both storage and workflow use-cases.

Title: FAIR Data Principles at Data Centers

Authors: Vincent Garonne*<vincent.garonne@bnl.gov>, Jerome Lauret*<jeromel@bnl.gov>, Alexandr Zaytsev*<alezayt@bnl.gov>, Eric Lancon*<elancon@bnl.gov>, Shigeki Misawa*<misawa@bnl.gov>. Tejas Rao*<raot@bnl.gov>

* Brookhaven National Laboratory

Topic: Devising metadata management infrastructure to support FAIR principles (Findability, Accessibility, Interoperability, and Reusability).

Challenge:

Supporting and implementing FAIR principles at data centers puts an emphasis on rapid automatic data discovery and efficient data access by users in scientific studies, while favoring the future reuse of data. For scientific collaborations it is common that metadata and its definition are globally distributed in heterogeneous catalogs, data repositories or stored with the data itself on different administrative domains. Research defined metadata, e.g., datasets, publications, are usually in data catalogs while storage defined ones, e.g., size, checksum, url, last access are stored with the data on storage. The inherent challenge for a data center is to provide a scalable, extensible and interoperable FAIR interface for the data and metadata stored on its domain and the respective services. The aim of such an interface is twofold: (1) to allow all users to further benefit from FAIR principles for data access, extraction and reuse (2) to enable Data Centers to gather more information and organize data accordingly to respond more efficiently to user needs like identifying patterns in data usage.

Opportunity:

Enabling FAIR data principles at Data Centers has the potential to accelerate science by providing our scientific communities the abilities to efficiently locate access and reuse data no matter their geo-location. As examples, integrating generic standards tools and principles will demonstrate its effectiveness with real use cases and production workflows; and introduce measurable benefits in the future evolution of Data Center architectures, especially for the storage design and metadata support across heterogeneous infrastructures.

Timeliness:

FAIR principles is an active topic for many big data sciences like weather, geo or health sciences. In an era where not only collaborations but experiments are no longer confined to a single place, this key to distributed science. As different implementations, tools and standards have been developed, it emphasizes the need to review the solutions and to take advantage of them in a context of data centers and cross-disciplinary support. A set of case study examples should be developed and maintained to demonstrate that providing FAIR data principles can increase the impact of Data centers by increasing data reuse and thereby return on investment in Data

centers. With FAIR principles, Data Centers can truly Federate at all levels of workflows and storage.

References

- Zenodo (http://zenodo.org/)
- https://www.nature.com/articles/sdata201618
- https://www.osti.gov/biblio/1606031-big-federal-data-centers-implementing-fair-data-principles-arm-data-center-example
- FAIR sharing www.fairsharing.org : Curated registry for standards
- CEDAR https://cedar.metadatacenter.org: Create, manage and fill metadata templates
- https://github.com/FAIRDataTeam/FAIRDataPoint : Expose metadata in a FAIR way
- https://github.com/FAIRDataTeam/FAIRDataPoint/wiki/FAIR-Data-PointSpecification FAIRifie

A Well-Designed Interface is a Trojan Horse for New Capabilities in Data Management and Data-intensive Processing

E. W. Bethel,* B. Loring, O. Rübel, G. Weber
LBNL
N. Ferrier, J. Insley, V. Mateevitsi, S. Rizzi
ANL
P. O'Leary, U. Ayachit, C. Wetterer-Nelson
Kitware, Inc.
E. Duque, B. Whitlock
Intelligent Light

1 Context and Focus

Our central message is that a well designed interface enables access to new types of data and processing capabilities not originally envisioned by it's architects, capabilities that are broadly applicable across a broad cross-section of DOE mission science codes. We observe that there are numerous challenges and opportunities in the areas of simplifying access to a rapidly growing collection of heterogeneous software tools in the scientific computing ecosystem (§2), and the inescapable reality of challenges in using heterogeneous computational platforms now and in the future for data-intensive processing (§3).

2 The Interface Enables Access to Diverse, Heterogeneous Processing Capabilities

Challenge. There has been an explosive growth in useful data-intensive software tools, but brittle interfaces inside of codes may be unable to keep pace with evolving technologies. The past decade has witnessed a rapid growth in the diversity of software tools that play important roles in DOE mission science. These include tools for AI/ML, for data analytics, performance measurement and analysis, code coupling, workflow environments, I/O, platform portable programming and more. The rapid growth in tools often results in increased complexity for code developers who must contend with and rectify differences in APIs, data models, execution models, and parallelization strategies. Given that we expect the first exascale systems to be put into service during 2022, we can expect additional unknown challenges will arise due to the nature of those systems.

Opportunity. Lowering the barriers to using heterogeneous tools for data intensive workloads will have broad benefit. The complexity of adopting and using a growing and changing ecosystem of scientific software tools is multi-faceted. When focusing on data-intensive scientific workloads, which in particular can benefit from third-party methods from research programs and industry, efforts to reduce complexity will likely be directed towards simplifying and streamlining the use of diverse software for data-intensive scientific analysis/learning/understanding pipelines where multiple codes and tools are coupled, and run in parallel on DOE HPC platforms.

Timeliness, Maturity, Impact. The SENSEI project [1] is an example of early success where a SENSEI-instrumented code may leverage any number of different parallel endpoints (Fig. 1), including user-written Python code [3, 6]. While SENSEI's strengths lie in its embrace of diversity in data-centric tools, the broader community will benefit from a compendium of knowledge that provides exemplars and curated, pre-configured examples of use cases and tool combinations important to DOE science. A potential impact of this type of breakthrough would be an acceleration in development and deployment time, where scientific software developers and users would have a ready-made set of "recipes" or "motifs" from which to quickly implement common processing patterns that involve use of 3rd party tools for AI/ML, for iterative data-intensive pipelines, and bidirectional data movement and execution control. Because all sciences are increasingly data-driven, lowering complexity of using diverse toolsets will have broad impact across DOE mission science.

*Corresponding author: ewbethel@lbl.gov

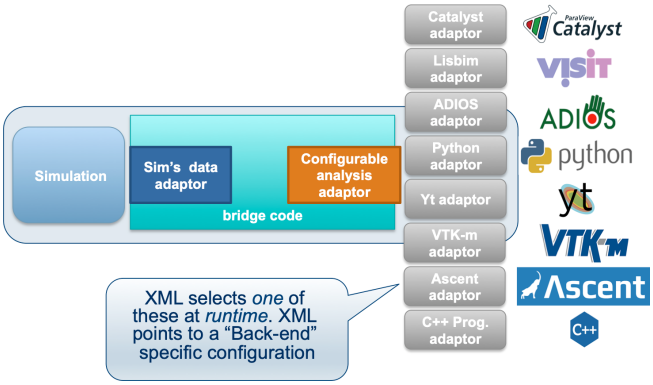


Figure 1: SENSEI's design promotes diversity of computational endpoints, where a user may switch between different endpoints with a runtime configuration file. The endpoints may include existing parallel applications, like VisIt, as well as user-written parallel Python that implements iterative scientific analysis pipelines. Image courtesy B. Loring.

3 The Interface Enables Access to and Use of Heterogeneous Computational Platforms

Challenge: Use of heterogeneous computational platforms for multistage, data-intensive scientific analytics pipelines is not a straightforward endeavor. While many code teams are leveraging GPU platforms for work in computational and data sciences, numerous issues emerge when considering the more complex task of using heterogeneous computational platforms for multistage scientific data-centric pipelines. These include the fact that different stages of the pipeline may not use the same approach for parallelization (domain decomposition), may need to run at different levels of concurrency and/or thread blocking factor due to underlying algorithmic characteristics, and data may need to be repartitioned and moved from producer to consumer as in the case with M-to-N processing pipelines [5]. Furthermore, different stages of these pipelines may be implemented using completely different technologies: some portions may be distributed-memory parallel with MPI, other portions may be targeted at a specific device, e.g, NVIDIA GPUs using CUDA.

Opportunity. Simplify and streamline the use of heterogeneous computational resources for multistage scientific data-centric pipelines. Given that the exascale-class platforms will all consist of heterogeneous platforms, efforts that target simplifying the use of these resources will be of broad benefit across many DOE science programs. Here, "simplifying use of" should be interpreted broadly, but with an emphasis on encouraging the diversity of tools and approaches that are in use now and that will emerge in the future.

Timeliness, Maturity, Impact. Significant effort has been directed towards the problem of code platform portability, with language extension models like SYCL [2] and Kokkos [4]. While these help with individual codes, they don't address the challenges that arise when combining multiple heterogeneous processing stages into a pipeline, where the pipeline stages are diverse and consisting of many potentially different types of code implementations. Advances in awareness and management of data residency and placement is a critical part of the computational landscape needed to support DOE mission science. Other efforts have shown the benefit of careful data partitioning and proximity portability in an M-to-N context [5], ideas that are likely to have applicability and benefit in heterogeneous computational environments. Together, these concepts of platform portability and proximity portability hold promise to be of benefit for data-centric multi-stage scientific pipelines on heterogeneous computational platforms.

References

[1] SENSEI – Scalable in situ analysis and visualization. `http://www.sensei-insitu.org`, last accessed Nov. 2021.

[2] SYCL 2020 Specification revision 4. `https://www.khronos.org/files/sycl/sycl-2020-reference-guide.pdf`, last accessed Nov. 2021.

[3] U. Ayachit, B. Whitlock, M. Wolf, B. Loring, B. Geveci, D. Lonie, and E. W. Bethel. The SENSEI Generic In Situ Interface. In *Proceedings of In Situ Infrastructures for Enabling Extreme-scale Analysis and Visualization (ISAV 2016)*, Nov. 2016. LBNL-1007263. doi: 10.1109/ISAV.2016.13

[4] H. Carter Edwards, C. R. Trott, and D. Sunderland. Kokkos: Enabling manycore performance portability through polymorphic memory access patterns. *Journal of Parallel and Distributed Computing*, 74(12), 7 2014. doi: 10.1016/j.jpdc.2014.07.003

[5] B. Loring, J. Gu, N. Ferrier, S. Rizzi, S. Shudler, J. Kress, J. Logan, M. Wolf, and E. W. Bethel. Improving performance of m-to-n processing and data redistribution in in transit analysis and visualization. In *EuroGraphics Symposium on Parallel Graphics and Visualization (EGPGV)*. Norrköping, Sweden, May 2020.

[6] B. Loring, A. Myers, D. Camp, and E. W. Bethel. Python-based in situ analysis and visualization. In *Proceedings of the Workshop on In Situ Infrastructures for Enabling Extreme-Scale Analysis and Visualization - ISAV '18*. ACM Press, 2018. doi: 10.1145/3281464.3281465

Compression-Assisted Data Management in Exascale Scientific Workflow

Xin Liang (Missouri S&T), Dingwen Tao (Washington State University), Jieyang Chen (Oak Ridge National Laboratory), Qing Liu (New Jersey Institute of Technology)
Contact: xliang@mst.edu

Topics: Data management support for AI and complex workflows.

Challenges: The increasing complexity in scientific workflow and unprecedented data volumes from scientific simulations and instruments are revolutionizing the I/O workloads in current and future High-Performance Computing (HPC) systems. Unlike the simple access patterns in traditional workflows, current scientific workflows may involve extensive interactions between data producers (e.g., scientific applications) and consumers (e.g., data analytics). This is further complicated by the large volume of data that needs to be exchanged, where the increasing gap between computing capability and data movement bandwidth will pose huge challenges in data transmission and analytics. Although error-controlled lossy compression has become a common way to reduce the size of scientific data and has been proved to be efficient for multiple applications [1, 2, 3], several challenges exist for deploying it for data management in exascale scientific workflows. First, identifying a proper compression method for data movement is challenging and should be dynamic [4]. Although data compression plays a trade-off between data preparation time and transmission time, the impact on each component varies a lot among different compression algorithms. This becomes more challenging considering the variability in I/O and storage systems. Second, since compression adds new complexity in data access latency, it is important to revise the data placement and prefetch strategies accordingly in current data management and storage systems. This will ask for careful decision-making in a finer granularity, compared to that of lossless compression methods which incur less variability in compression ratios. Furthermore, coordinating the data management among multiple workflows when they are running on the same system is very challenging, because dynamic workflows can be added and introduce new contentions for shared resources such as aggregated burst buffers at any time. Third, adding compression creates new difficulties in configuring the workflow, since the best configuration might change depending on the compression methods adopted. For instance, a previously in-transit analysis may be performed in situ if in-memory compression is used to reduce the memory overhead. In addition, the impact of compression differs in a write-heavy workflow (e.g., scientific applications for data generation) and a read-heavy workflow (e.g., training phase of deep neural networks), because of the asymmetric performance of data compression and decompression.

Opportunity: As leveraging lossy data compression becomes a trend for exascale scientific applications, data management and storage systems need to be rethinked and redesigned to address the above challenges. This imposes several research opportunities, which can be summarized as the following questions. 1) *Compression-assisted data movement: how can we determine the proper method for compression to accelerate data movement?* This requires understanding the speed-ratio trade-offs for various compression algorithms and building accurate performance models for both compression and data movement. The former can be achieved by a careful decomposition of compression stages, where each stage will be modeled with the combination of theoretical analysis and sampling methodologies; the latter can be estimated using performance monitors along with AI models. Then, a preferred data compression method can be identified based on the performance models. For instance, homomorphic compression, where computation can be performed in compressed representations, shall be used for aggregated data movement such as Allreduce in MPI to reduce the data preparation cost. This will make a big difference on how data is exchanged in in-situ, in-transit, or staging analytics, and lays a foundation

for the following tasks. 2) *Compression-assisted data placement and prefetch: how should we place and prefetch compressed data in hierarchical storage systems?* This includes both determining the proper compression method for each data segment and how to manage them dynamically in the storage hierarchy while the workflow is running. A simple way for the placement strategy is to revise a performance model in [5] with finer granularity and better cost model to account for the variability in lossy compression. With the evidence that associating adaptive lossless compression methods with different levels of storage hierarchy improves I/O performance, it is promising that adopting lossy compression will further boost the performance due to its flexibility. For instance, lossy compression can enable in-memory computation with a decompression-computing-compression process due to the high compression ratios, which allows for faster analytics [1] and/or larger-scale applications [2]. As for data management coordination across multiple workflows, a possible way is to combine the application-centric methods with the data-centric methods [6], where the former has proactive knowledge while the latter persists a global view of data flow. Such a method is expected to make better decisions with additional information. 3) *Compression-assisted workflow configuration: how can we appropriately configure the workflow with compression?* This can be simplified to an optimization problem with proper parameterization on the compression methods and workflow design, and solved by hyperparameter search when data placement decisions are considered unchanged. A better estimation needs to consider the interaction between data placement and workflow design, which requires an iterative process such as expectation-maximization method or evolutionary algorithm to identify the best option. To consider the impact of other running workflows, the data placement strategy may need to leverage other information such as the global data flow pattern and resource utilizations as well.

Timeliness and maturity: Scientific applications and instruments are producing more data than that can be stored, transmitted, and analyzed, causing problems in many aspects of the data management and storage systems. Error-controlled lossy data compression significantly reduces the data volumes while preserving necessary information, providing a direct solution to address the data challenge. Nevertheless, there is limited research and development effort for the adoption of lossy compression in data management and storage systems. As both lossy compression and interfaces for storage hardware are becoming more and more mature and necessary, it is crucial to explore such usage at the current stage. This will allow for better utilization of the existing hardware in HPC systems, reducing the time to insights for various scientific disciplines. Meanwhile, it will enable the execution and exploration of larger-scale problems, opening possibilities for novel scientific discoveries.

Reference:

[1] Gok, Ali Murat, et al. "Pastri: Error-bounded lossy compression for two-electron integrals in quantum chemistry." 2018 IEEE international conference on cluster computing (CLUSTER). IEEE, 2018.
[2] Wu, Xin-Chuan, et al. "Full-state quantum circuit simulation by using data compression." Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis. 2019.
[3] Cappello, Franck, et al. "Use cases of lossy compression for floating-point data in scientific data sets." The International Journal of High Performance Computing Applications 33.6 (2019): 1201-1220.
[4] Liang, Xin, et al. "Improving performance of data dumping with lossy compression for scientific simulation." 2019 IEEE International Conference on Cluster Computing (CLUSTER). IEEE, 2019.
[5] Devarajan, Hariharan, et al. "Hcompress: Hierarchical data compression for multi-tiered storage environments." 2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS). IEEE, 2020.
[6] Devarajan, Hariharan, Anthony Kougkas, and Xian-He Sun. "Hfetch: Hierarchical data prefetching for scientific workflows in multi-tiered storage environments." 2020 IEEE International Parallel and Distributed Processing Symposium (IPDPS). IEEE, 2020.

Identifying Root Causes of I/O Performance Deficit on HPC Systems through Holistic I/O Stack Analysis

Xu Liu (xliu88@ncsu.edu)
North Carolina State University

Lipeng Wan (wanl@ornl.gov)
Oak Ridge National Laboratory

High-bandwidth, large-capacity storage systems have been built to absorb big scientific data. However, applications running on high-performance computing (HPC) systems still often experience I/O performance deficit [2], which significantly degrade scientific applications' overall execution efficiency. Therefore, identifying and understanding why these I/O performance issues occur is important for data-intensive applications running at Exascale.

1. Challenges

On common HPC systems, the first layer of the I/O software stack is usually the I/O library which hides the complexity of the parallel file system (PFS) from the application. Some of the I/O libraries like ADIOS directly call POSIX functions to interact with the PFS, while the others like HDF5 call MPI-IO library first which then triggers POSIX function calls. The client-side library of the PFS is the second layer of the I/O software stack. The POSIX I/O requests are rearranged at this layer and forwarded to the server-side library of the PFS through the network. Once the I/O requests are further aggregated and reorganized by the server-side library, they are passed to the local file system of the storage server through the OS kernel and finally committed to the storage devices. As we can see, the I/O software stack in HPC environments can be deep and complex.

Since the actual I/O stack for each application running on the same HPC system might be different (e.g. some libraries are only used by a certain application), in order to be generalizable, we abstract the cross-layer behavior as the interactions between the upper layer and lower layer. Typically, the lower layer provides necessary services (i.e., APIs) to allocate resources and schedule tasks created by different programs in the upper layer to maximize the throughput. The upper layer expresses necessary semantics (e.g., algorithms) via invoking the APIs provided by the lower layer to minimize its latency. Software bugs, inappropriate configurations, and resource contentions can easily introduce defects in the entire stack, which makes identifying the root causes of I/O inefficiencies challenging.

Particularly, we highlight three potential issues in I/O stack that can cause performance issues. First, the upper layers may use suboptimal algorithms that invoke heavy, useless, or unnecessary APIs from the lower layers. Second, the APIs provided by the lower layers are not well implemented, which incurs inefficiencies upon usage. Third, the upper and lower layers might have different performance requirements. For example, the I/O requests from the upper layer are small and more latency-sensitive, while the lower layer is designed for achieving high I/O throughput. Such a mismatch can lead to inefficient I/O bandwidth utilization.

2. Opportunities

In our vision, to understand how the I/O requests are propagated along the entire software stack and identify the root causes of any I/O performance deficit, a novel profiling tool is needed. This tool can penetrate the abstractions (i.e., upper and lower layers) along the I/O path and correlate the behaviors across multiple layers. For example, by using this tool, the user can understand the causal relationship between the API calls of I/O libraries and the operations conducted by the PFS internally. Moreover, if the timing information is also captured, this tool can also locate at which layer in the I/O stack the performance bottleneck occurs. We foresee the following two techniques are the keys to make implementing this profiling tool possible.

- *Top-down correlation.* One can monitor the execution in a higher-level layer in the system stack. When an event (e.g., PMU sample) occurs, one can query the state maintained in the lower-level layer and correlate the event in the higher-level layer with it. It requires some bookkeepings in the lower-level layer and the state query interface.
- *Bottom-up correlation.* One can also monitor the execution in a lower-level layer. When an event occurs, one can correlate this event with the activity in the higher-level layer. This will be especially useful for understanding the semantics involved in the inefficiencies in the lower-level layers.

In practice, although there are still many technical challenges need to be addressed to fully support the cross-layer I/O behavior analysis, a variety of existing tools can be leveraged as building blocks. On the software side, one can employ Intel Pin binary rewriter, Dyninst, eBPF, and LLVM compiler to analyze software behaviors. On the hardware side, one can rely on performance monitoring units (PMU) and debug registers to capture architecture-related events. Thanks to these software tools and hardware capabilities that are widely supported by most of the HPC architectures, capturing the behavior in each individual layer of the I/O stack is promising. However, to profile the entire I/O stack and correlate the I/O related operations across layers without incurring too much overhead remains a challenging problem. Novel algorithms and techniques are needed to achieve this final objective.

3. Timeliness

This research is timely. As the applications, I/O stack, and OS become mature under the support of ECP, understanding the performance across the I/O stack layers is an urgent task. The existing performance analysis tools mostly target applications, which are not mature to give a holistic view for the HPC I/O stack. Today, the evolution of various measurement techniques such as performance monitoring units, eBPF, and binary analysis engines makes this research feasible. This research will complement the coarse-grained tracing tool, Darshan [1].

References Cited

[1] CARNS, P., HARMS, K., ALLCOCK, W., BACON, C., LANG, S., LATHAM, R., AND ROSS, R. Understanding and improving computational science storage access through continuous characterization. *ACM Trans. Storage 7*, 3 (oct 2011).

[2] PAUL, A. K., FAALAND, O., MOODY, A., GONSIOROWSKI, E., MOHROR, K., AND BUTT, A. R. Understanding hpc application i/o behavior using system level statistics. In *2020 IEEE 27th International Conference on High Performance Computing, Data, and Analytics* (2020).

Fast Dataset Discovery Strategies using Efficient Metadata Search

Yong Chen [1], Wei Zhang [2], Houjun Tang [3]
[1] Texas Tech University, yong.chen@ttu.edu
[2] Oracle and Texas Tech University, X-Spirit.Zhang@ttu.edu
[3] Lawrence Berkeley National Laboratory, htang4@lbl.gov

Topic: This position paper discusses metadata management infrastructure to support FAIR principles and the overlap between traditional storage systems and I/O (SSIO) efforts and data management.

Overview: Large-scale scientific applications increasingly need to manipulate enormous volumes of data, which imposes substantial challenges for efficiently finding the data that scientists require for performing their research and discoveries. The need to address dataset search challenges has been acknowledged by numerous recent initiatives, including FAIR principles, Research Data Alliance, and a prior DOE ASCR SSIO Workshop [1]. This problem, if unaddressed, could hinder scientific advancement across many fields. In existing practices, the metadata stored in datasets is a typical source for understanding and locating desired scientific data. Many scientific domains store experimental, observational, and simulation datasets in self-describing data formats, such as netCDF, HDF5, ROOT, FITS, Zarr, N5, PnetCDF, and ADIOS-BP. The primary advantage of self-describing file formats is that they allow scientists to store data and metadata together side by side using attributes. The metadata can be accessed as a collection of attributes that are in the form of key-value pairs. Each attribute key-value pair consists of a key representing the attribute name and a value representing the metadata attribute. The metadata in self-describing files provides detailed descriptive information about the internal data objects.

Challenges and Research Gaps: As metadata is stored alongside the datasets in self-describing data formats, the problem of searching over massive datasets, which are typically stored in thousands to millions of files and organized in deep directory hierarchies, can be achieved by performing metadata search over the metadata attributes. However, there are no efficient search strategies for the massive amounts of heterogeneous metadata stored in self-describing formats and file systems. Additionally, a dataset search is typically for exact search without the inclusion of any semantic meaning. Furthermore, datasets are managed in an isolated manner without any relationships. Efficient metadata search remains a challenging, open problem which we discuss below in more detail.

First, current data management lacks self-contained, interoperable search solutions. Scientists commonly issue structured queries against metadata attributes to find the corresponding matched data objects. In order to perform structured queries, scientists often rely on external metadata search systems based on a database management system, such as BIMM, EMPRESS 2.0, JAMO, or SPOT Suite. In these dataset search solutions, the metadata in the self-describing data formats has to be extracted and stored in an external data infrastructure (i.e., the database) along with the indexes built for dataset search purposes. However, these solutions employ an external data infrastructure that is disjoint from the self-describing data files and, as such, conflicts with the goals of the self-describing and self-contained data management paradigm. Moreover, these solutions introduce additional overhead in deployment and require constant maintenance in order to update the external database when data is changed. Furthermore, initial deployment of the database system for metadata search requires reading the metadata from the self-describing files and then loading it into the database. This process duplicates the metadata into two places and hence leads to storage redundancy. More importantly, metadata databases have to be either allocated or migrated when the self-describing files are transferred to a different system, which is a tedious process. This area deserves in-depth examination to develop a self-contained indexing, querying, and searching solution for scientific datasets.

Second, current data management methodologies suffer from poor metadata quality and a lack of semantic understanding. In existing systems, scientists are forced to issue structured queries with query conditions that exactly match the metadata attribute names and/or the attribute values. In other words, a successful search can only be achieved by locating exact or partial lexical matches between the metadata attributes and queries. As a result, these solutions are often criticized as inefficient. The root cause of these limitations is the inability to capture the semantic

relationships between the content of the metadata and the query keywords, which precludes performing queries at the semantic level. There are many constraints with lexical data search methods for self-describing files. Due to the specific structure of the metadata, such approaches often require scientists to navigate through their dataset schema and understand the metadata attributes to be able to query and search datasets that are of interest. This renders it impractical for scientists to mine a large number of datasets due to inconsistent metadata attributes and naming schema (e.g., two datasets can describe the exactly same observation, but with two different metadata attributes as "speed" and "velocity", respectively). Users may spend an inordinate amount of time manually locating attributes or need to rely on external resources or repositories to assist, even before starting their experiments. This challenge of metadata quality and lack of semantic meaning also needs to be addressed in order to deliver an efficient data management solution at scale.

Third, current data management practices utilize datasets in an isolated manner and are unable to exploit links between them. Scientific datasets are typically stored and managed in an isolated manner without capturing relationships among them. Self-describing datasets store metadata related to only the data in the constituent files. There is no metadata relating to other files or datasets describing their correlations. This isolation is a barrier for leveraging patterns and relationships among datasets for effective searches. In fact, the concept of linked data has attracted an increasing amount of attention in the web domain wherein insights from mining oceans of web pages on the Internet has shown promising results. Current linked data principles, however, are difficult to leverage in scientific datasets due to different schemas (or structures) and naming conventions used by different organizations, lack standard specification, and the complexity in binary data formats and data changes. This area deserves in-depth investigation in order to develop a linked data management solution for scientific datasets.

Opportunities: To address these aforementioned challenges, four topics of research should be conducted by the research community. First, the community needs to develop an efficient indexing strategy for heterogeneous metadata that is stored in self-describing file formats and file systems. Second, we need to explore the usage of AI, machine learning, and natural language processing methods for extracting semantic meanings in metadata attributes. Third, we should develop methods to build and manage relationships among datasets that can be searched as linked data. Fourth, we should design and develop innovative search APIs in high-level I/O libraries (i.e., HDF5, netCDF, etc.) that can utilize these methods. Numerous concepts and research efforts have started exploring these areas. For example, we have developed a Metadata Indexing and Querying Service to deliver efficient indexing and querying for scientific datasets [2], along with an open-source prototype available [3]. Other research efforts in the web domain, such as RDF (Resource Description Framework), knowledge graph models, and the usage of AI and natural language processing approaches in mining semantics provide an inspiration to the scientific community.

Timeliness: In this position paper, we argue that efficient metadata search is a critical, challenging, and open problem in current scientific data management and storage practices. However, numerous cutting-edge research efforts have shown promising potential for addressing these challenges by offering self-contained indexing and querying, integrating semantics for searching, etc. Given the growing importance of collecting rich metadata and maintaining FAIR compliance, these gaps deserve the data management community's attention and collective effort. The return on investment in these investigations will be profound and result in the ability to efficiently find data and improve scientific productivity.

References

[1] R. Ross, L. Ward, P. Carns, G. Grider, S. Klasky, Q. Koziol, G. Lockwood, K. Mohror, B. Settlemyer, and M. Wolf. Storage Systems and Input/Output: Organizing, Storing, and Accessing Data for Scientific Discovery. Report for the DOE ASCR Workshop on Storage Systems and I/O. https://www.osti.gov/servlets/purl/1491994. 2018.
[2] W. Zhang, S. Byna, H. Tang, B. Williams and Y. Chen. MIQS: Metadata Indexing and Querying Service for Self-describing File Formats. SC'19, 2019.
[3] MIQS Prototype Software Release. https://bitbucket.org/berkeleylab/miqs/src/master/.

FAIR Data and Model Service for AI

Zhengchun Liu (zhengchun.liu@anl.gov), Ahsan Ali, Rajkumar Kettimuthu, Ian Foster
Data Science and Learning division, Argonne National Laboratory
Nageswara Rao, Oak Ridge National Laboratory

Topic: Data-management support for AI and complex workflows.

Challenge Data generated by experiments, simulations, and digital twins, and ML models derived from those data for use in digital twins, are used on multiple time and distance scales. For example, data from an in-situ experiment must be delivered quasi-instantaneously to the AI model trainer that updates the digital twin used to choose the next experiment [2]. Here, speed, as to be provided by the data-management service, is of the essence. Those same data and/or AI models trained may also have value for the next experiment, for the construction and updating of other AI models, and to other scientists. Thus, data and AI models need also to be FAIR. These characteristics will requires a **FAIR Data and Model Service** (`FAIR-DMS`), which will provide *publication*, *enrichment*, *discovery*, and *access* capabilities for the DevOps of AI based applications.

Opportunity When move to Software 2.0 where expertise and even physics are represented implicitly by data instead of explicitly coded in a programming language, the `FAIR-DMS` needs to consider the unique characteristics of the AI/data-driven component of the software being developed. For example, requirements on `FAIR-DMS` including (but not limited to):

- Machine learning is an exploratory engineering technique, its development requires exploring different architecture, parameters and data. The ability to track numerous ML training experiments will be needed to accelerate the development and testing process.
- Each science problem is unique from a certainty point of view which causes lack of data to update/retrain ML model. As physics are encoded and represented in data (possibly from different experiments), `FAIR-DMS` needs to version and index different data automatically (e.g., using AI but should relax the demanding on meta-data).
- The successes of foundation models for natural language processing have shown that knowledge/physics/expertise encoded in the "black-box" neural network can be versatile. Thus, a ML model management will be needed in the `FAIR-DMS` to index trained ML models and rank them in a way for developer to find the best model for fine-tuning and transfer learning to other downstream tasks.

In addition, to advance the state of the art in the use of ML/AI methods in science, we will also need to investigate methods for the synthesis of content-based descriptors for scientific datasets. Scientific image repositories e.g., TomoBank [1], digital rocks portal [3], PSI public data repository [4]) typically rely on human-supplied annotations for indexing and search [1, 3, 4]. Successful AI relies on a number of factors including a large corpus of data. While data management can improve the development of AI applications, AI can be used to improve data management in areas such as data ingestion and query performance. As shown in Figure 1, an automatic image indexing and searching service based on learned representations can be achieved with AI techniques such as using self-supervised representation learning, transfer learning used to extract representations

automatically from images, sequences, and molecular graphs. The basic philosophy underlying this architecture is a transformation from the data-rich representation of explicit image pixels to a compact, semantic-rich representation of visually salient characteristics. We expect that by reducing the need for user-supplied metadata, this solution may encourage data contributions.

Thus, data management systems and AI are synergistic. When AI becomes embedded throughout the data management system, it has the potential to improve database query accuracy and performance, as well as optimize system resources, reducing the burden on human admins while improving data access for data scientists and developers. When AI gets embedded at the data layer, it creates a synergistic relationship between the underlying data management system and the development of AI applications, which has the potential to impact the entire data life-cycle. Data architects should ensure that they are delivering data infrastructure that is capable of supporting the rapid development of AI applications through direct support for machine learning tools and frameworks and the acceleration of the complete AI data pipeline.

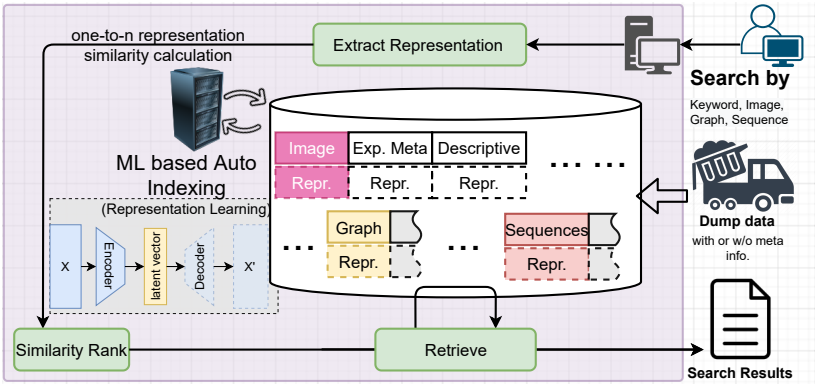


Figure 1: A search engine for scientific data powered by AI techniques for the DevOps of AI based scientific applications.

Timeliness Content-based data retrieval allows users to (also) search for datum that are similar to supplied datum. Emerging research and commercial systems enable content-based image retrieval, a technique which explicitly manages image assets by directly representing their visual attributes. There are currently two dominant interface types for searching and browsing large image collections: keyword-based search, and searching by overall similarity to sample images.

References

[1] F. De Carlo, D. Gürsoy, D. J. Ching, K. J. Batenburg, W. Ludwig, L. Mancini, F. Marone, R. Mokso, D. M. Pelt, J. Sijbers, et al. TomoBank: A tomographic data repository for computational x-ray science. *Measurement Science and Technology*, 29(3):034004, 2018.

[2] Z. Liu, A. Ali, P. Kenesei, A. Miceli, H. Sharma, N. Schwarz, D. Trujillo, H. Yoo, R. Coffee, N. Layad, J. Thayer, R. Herbst, C. Yoon, and I. Foster. Bridge data center ai systems with edge computing for actionable information retrieval. In *The 3rd Annual Workshop on Extreme-Scale Experiment-in-the-Loop Computing*, 2021.

[3] M. Prodanovic, M. Esteva, R. A. Ketcham, M. Hanlon, M. Pettengill, A. Ranganath, and A. Venkatesh. Digital Rocks Portal: Preservation, Sharing, Remote Visualization and Automated Analysis of Imaged Datasets. In *AGU Fall Meeting Abstracts*, volume 2016, Dec. 2016.

[4] C. Spurin, T. Bultreys, M. Rücker, G. Garfi, C. M. Schlepütz, V. Novak, S. Berg, M. J. Blunt, and S. Krevor. Real-time imaging reveals distinct pore scale dynamics during transient and equilibrium subsurface multiphase flow. *Earth and Space Science Open Archive ESSOAr*, 2020.