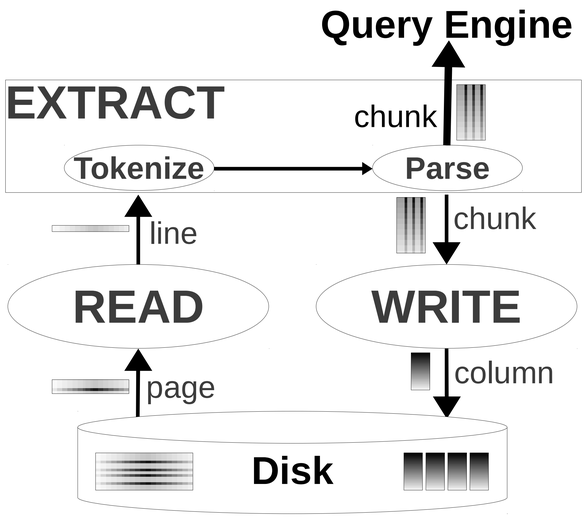
Vertical Partitioning for Query Processing over Raw Data

Weijie Zhao, Yu Cheng, Florin Rusu
Electrical Engineering and Computer Science, University of California, Merced
wzhao23, ycheng4, frusu@ucmerced.edu

Query Processing over Raw Data

The generic procedure to extract tuples with the given schema from the raw file



Mixed Integer Programming

minimize $T_{load} + \sum_{i=1}^{m} w_i \cdot T_i$ subject to constraints:

$$C_1: \sum_{j=1}^{n} save_j \cdot SPF_j \cdot |R| \leq B$$

$$C_2: read_{ij} \leq save_j;\ i = \overline{1,m},\ j = \overline{1,n}$$

$$C_3: save_j \leq p_{0j} \leq t_{0j} \leq raw_0;\ j = \overline{1,n}$$

$$C_4: p_{ij} \leq t_{ij} \leq raw_i;\ i = \overline{1,m},\ j = \overline{1,n}$$

$$C_5: t_{ij} \leq t_{ik};\ i = \overline{0,m},\ j > k = \overline{1,n-1}$$

$$C_6: read_{ij} + p_{ij} = 1;\ i = \overline{1,m},\ j = \overline{1,n},\ A_j \in Q_i$$

T_{load} is defined as:

$$T_{load} = raw_0 \cdot \frac{S_{RAW}}{band_{IO}} + |R| \cdot \sum_{j=1}^{n} \left(t_{0j} \cdot T_{t_j} + p_{0j} \cdot T_{p_j} + save_j \cdot \frac{SPF_j}{band_{IO}} \right)$$

Execution time of a query T_i is a slight modification:

$$T_i = raw_i \cdot \frac{S_{RAW}}{band_{IO}} + |R| \cdot \sum_{j=1}^{n} \left(t_{ij} \cdot T_{t_j} + p_{ij} \cdot T_{p_j} + read_{ij} \cdot \frac{SPF_j}{band_{IO}} \right)$$

Query Coverage

Input: Workload $W = \{Q_1, \ldots, Q_m\}$; storage budget B
Output: Set of attributes $\{A_{j_1}, \ldots, A_{j_k}\}$ to be loaded in processing representation

1: $attsL = \emptyset$; $coveredQ = \emptyset$
2: **while** $\sum_{j \in attsL} SPF_j < B$ **do**
3: $idx = \text{argmax}_{i \notin coveredQ} \left\{ \frac{cost(attsL) - cost(attsL \cup Q_i)}{\sum_{j \in \{attsL \cup Q_i \setminus attsL\}} SPF_j} \right\}$
4: **if** $cost(attsL) - cost(attsL \cup Q_{idx}) \leq 0$ **then break**
5: $coveredQ = coveredQ \cup idx$
6: $attsL = attsL \cup Q_{idx}$
7: **end while**
8: **return** $attsL$

Attribute Usage Frequency

Input: Workload $W = \{Q_1, \ldots, Q_m\}$ of R; storage budget B; set of loaded attributes $saved = \{A_{s_1}, \ldots, A_{s_k}\}$
Output: Set of attributes $\{A_{s_{k+1}}, \ldots, A_{s_{k+t}}\}$ to be loaded in processing representation

1: $attsL = saved$
2: **while** $\sum_{j \in attsL} SPF_j < B$ **do**
3: $idx = \text{argmax}_{j \notin attsL} \{ cost(attsL) - cost(attsL \cup A_j) \}$
4: $attsL = attsL \cup idx$
5: **end while**
6: **return** $attsL$

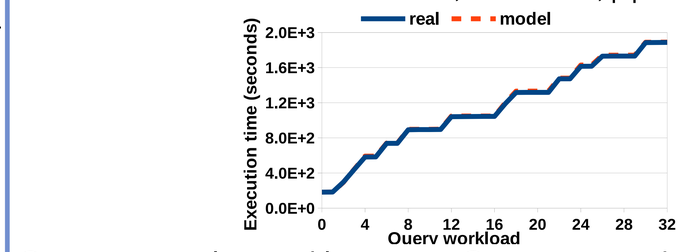
Heuristic Algorithm

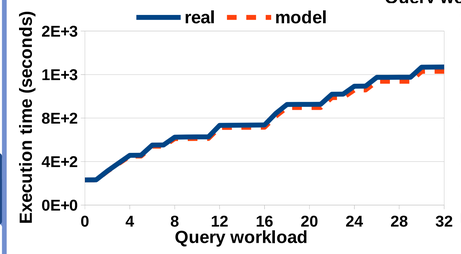
Input: Workload $W = \{Q_1, \ldots, Q_m\}$; storage budget B
Output: Set of attributes $\{A_{j_1}, \ldots, A_{j_k}\}$ to be loaded in processing representation

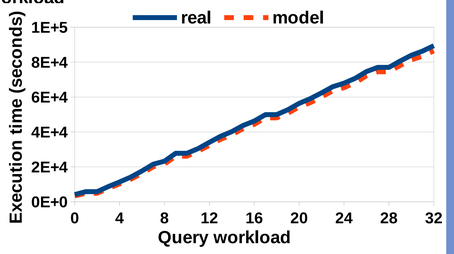
1: $obj_{min} = \infty$
2: **for** $i = 0$; $i = i + \delta$; $i \leq B$ **do**
3: $attsL_q = Query\ coverage(W, i)$
4: $attsL_f = Attribute\ usage\ frequency(W, \Delta_q, attsL_q)$
5: $attsL = attsL_q \cup attsL_f$
6: $obj = cost(attsL)$
7: **if** $obj < obj_{min}$ **then**
8: $obj_{min} = obj$
9: $attsL_{min} = attsL$
10: **end if**
11: **end for**
12: **return** $attsL_{min}$

Experiments

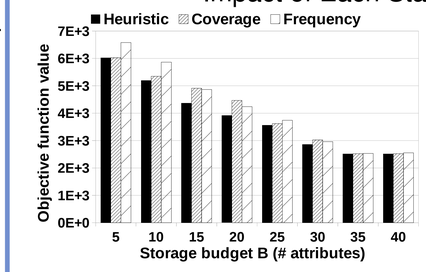
Model validation: serial CSV, serial FITS, pipeline JSON.

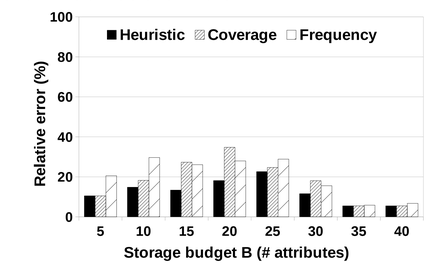




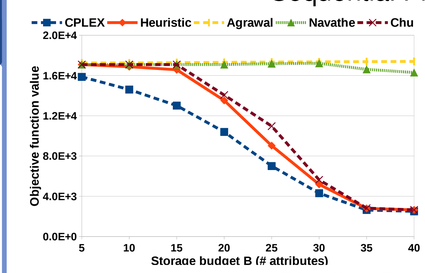


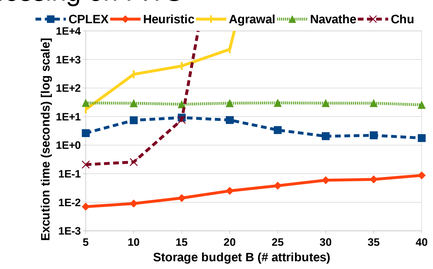
Impact of Each Stages of Heuristic on CSV





Sequential Processing on FITS





Pipeline Processing on JSON