Technical Report: Memory Management Approach in HTAP Systems

December 30, 2024

1 Symbols

To make the reading process more convenient for readers, we provide a symbol table containing the key symbols used in the paper, allowing readers to easily reference their corresponding meanings.

Symbols	Description
$\mathcal{M}_{\text{total}}$	The total memory available for the whole buffer pool.
$\mathcal{M}_{\text{row}}, \mathcal{M}_{\text{col}}$	The memory allocated to the row and column store buffers, respectively.
$W(C)$	The size of the selected column set.
$\mathcal{P}_{OLTP}$	A model or function that predicts the TPS for the current OLTP workload.
$\mathcal{K}$	The data synchronization strategy.
$t_0, t_1, \ldots, t_n$	Equal-sized time windows, where t_i represents the i-th time window.
θ_{t_i},θ'_{t_i}	Required TPS and its forecasted value for the upcoming time window t_i. The forecasted value (θ'_{t_i}) is predicted by the workload predictor.
$\mathcal{W}\text{ap}(t_i)$, $\mathcal{W}'\text{ap}(t_i)$	The OLAP workload for time window t_i, where $\mathcal{W}\text{ap}(t_i)$ is the actual workload and $\mathcal{W}'\text{ap}(t_i)$ is the forecasted value.
C_{t_i}	The chosen column set in time window t_i.
$\text{Cost}_{\mathcal{W}_{\text{ap}}(t_i)}(C_{t_i})$	The OLAP workload cost at time t_i.
$\text{Cost}_{\Delta}(C_{t_i}, \mathcal{K})$	The delta synchronization cost at time t_i.
$\text{Cost}^{t_i}_{\text{switch}}$	The switching cost incurred when triggering dynamic memory allocation at time window t_i.
$F(t_i)$	The memory reallocation strategy that determines whether to trigger reallocation at the start of time window t_i.

Table 1: Symbol Table (Section 2: Problem Definition)

Symbols	**Description**
$Q_1, Q_2, \ldots, Q_s$	Original queries in the OLAP workload $\mathcal{W}_{\text{ap}}$.
M	The total number of columns in the database.
$C_1, C_2, \ldots, C_M$	The columns in the database.
K	The total number of subqueries in the OLAP workload $\mathcal{W}_{\text{ap}}$.
q_l	The subquery responsible for data scanning and filtering. The original query Q_s may contain more than one subquery.
G_l	The group of columns involved in subquery q_l. $\lvert G_l \rvert$ denotes the number of columns in group G_l.
f_l	The execution frequency of subquery q_l.
$\text{cost}^{q_l}_{\text{row}}$,$\text{cost}^{q_l}_{\text{col}}$	The cost of retrieving data through sequential scan or column scan for subquery q_l, respectively.
x_m	A decision variable indicating whether column C_m is selected.
w_m	The memory usage of column C_m.
z_l	A decision variable indicating whether query q_l is scanning from column storage.
$\text{Cost}_{\text{read}}$,$\text{Cost}_{\text{sync}}$	Costs of reading and synchronizing the delta table, respectively.
$N(t)$	The number of records in the delta table at time t.
w_0, w_1, w_2, w_3	Coefficients for the linear cost models of reading and synchronizing the delta table.
α	The synchronization threshold.
J	The total number of tables in the database.
b_j	The number of UID operations on table j.
ν_j	The number of times delta table j is accessed.
Cost^j_Δ	The total cost associated with reading and synchronizing the delta table of table j.
u_j	A decision variable indicating whether the columns of table j are loaded into memory.
S_j	The set of columns in table j.
$\mathcal{U}(\mathcal{M}_{\text{col}})$	Objective function representing the total cost associated with a given column memory allocation
K'	Reduced number of communities after spectral clustering

Table 2: Symbol Table (Section 3: T^2 FOR STATIC WORKLOADS)

Symbols	Description
Q_s	A specific OLAP query template.
$R_t^{(Q_s)}$	The request rate for query template Q_s at time t.
L	The total number of query templates in the workload.
$\mathbf{R_h}$	The sequence of request rates for all query templates up to time h.
γ	The number of future time intervals to predict.
$\hat{R}_{t+\gamma}^{(Q_s)}$	The predicted request rate for query template Q_s at time interval $t+\gamma$.
ΔC_{t_i}	The set of columns that are present at time interval t_i but not at the previous interval t_{i-1}.
$Tables(\Delta C_{t_i})$	A function returning the set of unique tables containing columns in ΔC_{t_i}.
$\mathrm{Cost}_{\mathrm{row}}(T_j)$	The cost for performing a full table scan on table T_j from the row store.

Table 3: Symbol Table (Section 4: T^2 FOR DYNAMIC WORKLOADS)

2 appendix

2.1 Pseudocode for GACC algorithm

We designed a greedy algorithm for column selection, and the pseudocode is shown below. First, an empty set *selected_columns* is initialized, and the available memory is set to $\mathcal{M}_{\mathrm{col}}$. Then, a greedy strategy is applied for column selection.

The set p contains all column combinations and their associated performance gains, denoted as 'score', which is calculated as $\mathrm{cost}_{\mathrm{row}}^{q_l} - \mathrm{cost}_{\mathrm{col}}^{q_l}$. During the greedy selection process, at each step, the column combination with the highest performance gain to memory cost ratio is selected. However, it is necessary to update the memory cost of each column combination during the selection process, as some columns already included in $selected_columns$ do not incur additional memory costs. Thus, lines 9–13 are used to recalculate the memory cost of each column combination. Lines 14–17 select the column combination with the highest benefit-to-weight ratio. After selecting the best column combination, lines 22–25 update the remaining available memory and the *selected_columns* set. By repeating this process, the available memory is gradually consumed until it is fully utilized.

Algorithm 1 Greedy Algorithm for Column Combinations (GACC)

```
Initialize selected_columns ← ∅
Initialize available_cost ← M_col
while available_cost > 0 do
    best_ratio ← 0
    best_combination ← None
    for combination ∈ p do
        total_score ← p[combination]['score']
        total_cost ← 0
        for column ∈ combination do
            if column ∉ selected_columns then
                total_cost ← total_cost + w[column]
            end if
        end for
        if total_cost ≤ available_cost and total_score/total_cost > best_ratio then
            best_ratio ← total_score/total_cost
            best_combination ← {(col, w[col]) : col ∈ combination \ selected_columns}
        end if
    end for
    if best_combination = None then
        break
    end if
    for (column, cost) ∈ best_combination do
        Add column to selected_columns
        available_cost ← available_cost − cost
    end for
end while
return selected_columns
```

2.2 Supplementing the Optimal Data Synchronization Strategy

We provide a more detailed derivation process, compared to the paper, to explain how the synchronization threshold is obtained.

We model this problem by selecting a time period $[0, T]$. We assume that data updates are uniformly distributed within this time period, meaning the number of records in the delta table grows linearly at a constant rate of $r = \frac{b}{T}$, where b is the total number of data update records during the time period $[0, T]$. A synchronization is performed whenever the number of records reaches the threshold α. Given the assumption of a constant data growth rate, the time interval between each synchronization is uniform. Since synchronization is triggered each time the number of records reaches α, there will be $\frac{b}{\alpha}$ synchronization events over the time period $[0, T]$. The cost of a single synchronization operation is $w_2\alpha + w_3$, resulting in a total synchronization cost during the time period $[0, T]$ of:

$$\text{Cost}_{\text{sync}} = \frac{b}{\alpha} \times (w_2\alpha + w_3)$$

Assume that reading events are uniformly distributed within the time period $[0, T]$. There are a total of ν reads in the time period $[0, T]$. Due to the assumption of a constant data growth rate, the time interval between each synchronization trigger is also the same, denoted as $T_\alpha = \frac{\alpha}{r}$. The expected time for each read is

$$E[\text{Cost}_{\text{read}}(t)] = \int_0^{T_\alpha} \text{Cost}_{read}(t)h(t)dt$$

where $h(t)$ is the probability density function of t, which is $\frac{1}{T_\alpha}$ in the case of a uniform distribution. Based on the previous discussion, $N(t) = rt$ and we have:

$$\begin{aligned} E[\text{Cost}_{read}(t)] &= \int_0^{T_\alpha} (w_0 rt + w_1)\frac{1}{T_\alpha}dt = \frac{w_0 r}{T_\alpha}\int_0^{T_\alpha} tdt + \frac{w_1}{T_\alpha}\int_0^{T_\alpha} dt \\ &= \frac{w_0 r}{T_\alpha} \cdot \frac{t^2}{2}\bigg|_0^{T_\alpha} + \frac{w_1}{T_\alpha} \cdot t\bigg|_0^{T_\alpha} = \frac{w_0 r \frac{\alpha}{r}}{2} + w_1 = \frac{w_0 \alpha}{2} + w_1 \end{aligned}$$

During each synchronization cycle of T_α, $\nu\frac{T_\alpha}{T}$ reads will occur. The total read cost during T_α is

$$\text{Cost}_{\text{read}}^{T_\alpha} = \nu\frac{T_\alpha}{T} \times (\frac{w_0\alpha}{2} + w_1)$$

The overall cost, Cost_Δ, is the sum of the synchronization and reading costs in each synchronization cycle, given by $\text{Cost}_\Delta = \text{Cost}_{\text{sync}} + \text{Cost}_{\text{read}}$:

$$\begin{aligned} \text{Cost}_\Delta &= \frac{b}{\alpha}(w_2\alpha + w_3) + \frac{b}{\alpha} \cdot \nu\frac{T_\alpha}{T} \times (\frac{w_0\alpha}{2} + w_1) \\ &= \frac{b}{\alpha}(w_2\alpha + w_3) + \nu(\frac{w_0\alpha}{2} + w_1) \end{aligned} \tag{1}$$

To find the value of α that minimizes Cost_Δ, we need to take the derivative of Cost_Δ with respect to α and set it to zero:

$$\begin{aligned} \frac{d\text{Cost}_\Delta}{d\alpha} &= \frac{d}{d\alpha}(\frac{b}{\alpha}(w_2\alpha + w_3) + \nu(\frac{w_0\alpha}{2} + w_1)) \\ &= \frac{dbw_2}{d\alpha} + \frac{d\frac{bw_3}{\alpha}}{d\alpha} + \frac{d\frac{\nu w_0\alpha}{2}}{d\alpha} + \frac{d\nu w_1}{d\alpha} = -\frac{bw_3}{\alpha^2} + \frac{\nu w_0}{2} = 0 \end{aligned}$$

By solving the equation above, value of α can be obtained:

$$\alpha = \sqrt{\frac{2bw_3}{\nu w_0}} \tag{2}$$