Workload-Driven Horizontal Partitioning and Pruning for Large HTAP Systems

Paris, 16 April 2018

Martin Boissier & Daniel Kurzynski
Project Goal

• Background:
  • In our research, we focus HTAP-optimized main memory-resident databases for modern enterprise systems
  • Own research database called Hyrise*, an open source columnar HTAP database

• Current Situation:
  • NUMA architectures pose new challenges for data distribution, scheduling, and query execution
  • Optimizing for NUMA has large impact (cf. [1, 2])

• Our Goal:
  • Evaluate various partitioning approaches to distribute a table’s data to $n$ NUMA nodes with the goal to maximize tuples skipped (i.e., data skipping)

[1] Leis et al., Morsel-Driven Parallelism: A NUMA-Aware Query Evaluation Framework for the Many-Core Age. SIGMOD ’14
[2] Pandis et al., Data-Oriented Transaction Execution. VLDB ’10

* Hyrise on GitHub: https://github.com/hyrise/hyrise
Modern NUMA Systems
Modern NUMA Systems

- For optimal performance on NUMA systems:
  - data shall be equally distributed
  - processing shall be data-local *

- We see **two problems** with this statement:
  - Equal distribution works fine for both TPC-C and TPC-H
    *it does not for real-world systems*
  - Partitioning elimination/pruning is considered an orthogonal topic
    *it should not be*

- This projects evaluates means to “combine both worlds”

* Cf. “NUMA commandments”, Albutui et al., Massively Parallel Sort-Merge Joins in Main Memory Multi-Core Database Systems. PVLDB 2012
Workload-Driven Partitioning

- We evaluated several partitioning approaches
- almost all of them to be too simple
- Aggressive Data Skipping by Sun et al. is one exception [3]
Aggressive Data Skipping

- Approach initially motivated for large-scale systems like Spark
- We misused the approach to create partitioning schemes
  - Configurable by the number of partitions to yield
  - We limit partition count to number of NUMA nodes
- The process
  - Parse workload and extract relevant selections + frequent item set mining
  - Scan data for distribution of features
  - Merge features to create partitions

Created Partitioning Scheme

- clustered and non-trivial
  - Feature 1:
    \[
    \text{mandt} = 2 \ & \ koart \ <> \ 'k' \ & \ koart = 'd'
    \]
  - Feature 2:
    \[
    \text{mandt} = 2 \ & \ koart = 'k' \ & \ bukrs = '9999'
    \]
  - ...
- Created partitions are freely defined by 15 features
Aggressive Data Skipping

• Problem:
  • Objective is the number of pruned tuples
  • Heavily favors selective and frequent Queries
    • i.e., OLTP queries
  • Many ways to adapt for multiple workload classes
    • Weighting by runtime, what-if based query costs, ...

• Idea: execute partitioning twice and merge
Merging Partition Schemes

- Analyze Workload
- Extract Features
- Vectorize Tuples
- OLTP Features
- OLAP Features
- OLTP Queries
- OLTP Partitioning Scheme
- Build Partitioning Scheme
- OLAP Queries
- OLAP Partitioning Scheme
- Combine Partitioning Schemes
- Final Partitioning Scheme
Results (i)

<table>
<thead>
<tr>
<th>Dataset</th>
<th>CNF Queries</th>
<th>DNF Queries</th>
<th>OLAP Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training Queries (May)</td>
<td>81.85% 82.14% 11.53%</td>
<td>83.33% 82.48% 11.68%</td>
<td>87.80% 89.56% 11.80%</td>
</tr>
<tr>
<td>Evaluation Queries (May)</td>
<td>88.96% 81.96% 11.80%</td>
<td>82.48% 83.33% 11.80%</td>
<td>88.96% 81.96% 11.80%</td>
</tr>
<tr>
<td>Evaluation Queries (December)</td>
<td>94.03% 94.06% 11.80%</td>
<td>86.51% 86.80% 11.80%</td>
<td>90.00% 90.03% 11.80%</td>
</tr>
</tbody>
</table>
Results (ii)

The diagram shows the number of queries accessed and skipped across different partitions. The x-axis represents the partition number, while the y-axis represents the number of queries. The diagram uses two colors: light blue for accessing queries and dark blue for skipping queries. Each bar represents a partition, with the height indicating the number of queries in each category.
My Personal Outlook

- **Self-driving trend & distributed systems** force us to put more emphasis on (re-)partitioning

- **What is missing?**
  - More work on skew-aware and pruning-optimized partitioning
  - **Proper cost models** for adept partitioning schemes
  - Personally, I doubt AI & DL will solve these problems for us
  - More emphasis on repartitioning (not only for NUMA systems! [4])