Modeling Analytics for Computational Storage

- Motivation
- Near storage opportunities
- Deconstruction of “big data” queries
- Push down to Near Storage
- Workload: TPC-DS
- Modeling Methodologies and Results
Motivation
Motivation: Near storage OLAP

Read IN all that HAY…
Motivation: Near storage OLAP

Where IS that needle ???
SmartSSD

Where IS that needle ???
SmartSSD

Read IN just needle.

Server
Near storage opportunities

- Compression/Decompression;
- Encoding/Decoding;
- Filter;
- Projection;
- Some aggregates (SUM, COUNT);
- SORT;
- Some JOINs.
Deconstruction of “big data” queries

TPC-DS Q44:
“List the best and worst performing products measured by net profit.“
For a specific store.

```
select asceding.rnk, i1.i_product_name best_performing,
       i2.i_product_name worst_performing
from(select *
      from (select item_sk,rank() over (order by rank_col asc) rnk
            from (select ss_item_sk item_sk,avg(ss_net_profit) rank_col
                  from store_sales ss1
                  where ss_store_sk = 2
                  group by ss_item_sk
                  having avg(ss_net_profit) > 0.9*(select avg(ss_net_profit) rank_col
                                    from store_sales
                                    where ss_store_sk = 2  and ss_hdemo_sk is null
                                   group by ss_store_sk))V1)V11
      where rnk < 11) asceding,
      (select *
       from (select item_sk,rank() over (order by rank_col desc) rnk
            from (select ss_item_sk item_sk,avg(ss_net_profit) rank_col
                  from store_sales ss1
                  where ss_store_sk = 2
                  group by ss_item_sk
                  having avg(ss_net_profit) > 0.9*(select avg(ss_net_profit) rank_col
                                    from store_sales
                                    where ss_store_sk = 2  and ss_hdemo_sk is null
                                   group by ss_store_sk))V2)V21
       where rnk < 11) descending,
      item i1,  item i2
      where asceding.rnk = descending.rnk
      and i1.i_item_sk=asceding.item_sk
      and i2.i_item_sk=descending.item_sk
      order by asceding.rnk  limit 100;
```
Push down to Near Storage

Operations pushed down:

- **SCAN**: I/O plus data transformation
- **FILTER**: row selection
- **PROJECTION**: column selection
Two clusters:
- SPARK-SQL
- Presto

TPC-DS sf10,000 (10TB dataset)

99 TPC-DS queries have different characteristics and performance behavior.
Two 8-node Hadoop clusters:
  • SPARK-SQL
  • Presto

One file format – PARQUET:
  • Columnar
  • Designed for OLAP applications
  • READ optimized
  • Self-contained METADATA
  • Existing Parquet Readers can FILTER/PROJECT certain datatypes using statistics in METADATA
Modeling methodologies

[Graph showing scatter plot with points labeled Qxx, P-Qxx, Q-Qxx, indicating SPARK-SQL Query, Analyzed SPARK-SQL Query, and Analyzed Presto Query metrics]

ICPE 2020
## SPARK-SQL modeling:

<table>
<thead>
<tr>
<th>Timestamp</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
<th>17</th>
<th>Note</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage-0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Read dimension table: Scan, Filter, Project, Aggregate</td>
</tr>
<tr>
<td>Stage-1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Read fact table: Scan, Filter, Project, Aggregate</td>
</tr>
<tr>
<td>Stage-2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Read fact table: Scan, Filter, Project, Aggregate</td>
</tr>
<tr>
<td>Stage-3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Sort, Aggregate</td>
</tr>
<tr>
<td>Stage-4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Sort, Aggregate</td>
</tr>
<tr>
<td>Stage-5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Join</td>
</tr>
</tbody>
</table>
## SPARK-SQL modeling:

<table>
<thead>
<tr>
<th>Stage</th>
<th>Steps</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stage-0</td>
<td>Read dimension table</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>: Scan, Filter, Project, Aggregate</td>
<td>1-5</td>
</tr>
<tr>
<td>Stage-1</td>
<td>Read fact table</td>
<td>6</td>
</tr>
<tr>
<td></td>
<td>: Scan, Filter, Project, Aggregate</td>
<td>7-10</td>
</tr>
<tr>
<td>Stage-2</td>
<td>Read fact table</td>
<td>11</td>
</tr>
<tr>
<td></td>
<td>: Scan, Filter, Project, Aggregate</td>
<td>12-15</td>
</tr>
<tr>
<td>Stage-3</td>
<td>Sort, Aggregate</td>
<td>16</td>
</tr>
<tr>
<td>Stage-4</td>
<td>Sort, Aggregate</td>
<td></td>
</tr>
<tr>
<td>Stage-5</td>
<td>Join</td>
<td>17</td>
</tr>
</tbody>
</table>
Presto modeling:

- Run query with original tables. Repeat query with model tables.
- Presto generates same query plan in both cases.
Modeling Results

Near Storage Speedup
10TB dataset size

Geometric Mean:
- Presto: 3.76x
- SPARK-SQL: 2.80x
Modeling Results

### Presto Model Speedup

**Geometric Mean:**
- SF10T: 3.76x
- SF1T: 1.31x

<table>
<thead>
<tr>
<th>Q9</th>
<th>Q13</th>
<th>Q28</th>
<th>Q44</th>
<th>Q49</th>
<th>Q51</th>
<th>Q56</th>
<th>Q75</th>
<th>Q76</th>
<th>Q88</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Speedup (log scale)**
Modeling Results

Presto Q44 at sf10T is the best speed up observed.
- Total bytes READ much smaller with Model – must use LOG SCALE
- Avg CPU utilization 4x smaller
- Response time decreases from 18+ minutes to 19 seconds
- Presto plan for Q44 does not scale
Modeling Results

(A) Q75 Original

(B) Q75 Original

(C) Q75 Model

(D) Q75 Model

23 minutes

3 minutes

10 minutes

3 minutes

CPU Utilization %

nvme* disk Reads (MB/sec)
Conclusion

Modeling Analytics for Computational Storage

- Near Storage optimizations for OLAP NOT universal
- Some queries see significant speedup from Near Storage opportunities
- We covered only basic operations (“low hanging fruit”)
- Other Operations also amenable to Push down to Near Storage

Questions?

Near Storage Speedup
10TB dataset size

Geometric Mean:
- Presto: 3.76x
- SPARK-SQL: 2.80x