Bohr: Similarity Aware Geo-distributed Data Analytics

Hangyu Li, Hong Xu, Sarana Nutanong
City University of Hong Kong
Big Data Analytics

Analysis

Generate
Data are geo-distributed

- US Virginia
- US California
- US Oregon
- Tokyo
- Singapore
- Sydney
- Frankfurt
- San Paulo
Centralizing data is infeasible

• Moving data through wide area networks (WANS) can be extremely slow.

• Data sovereignty laws.

1. Vulimiri et al., NSDI’15
2. Pu et al., SIGCOMM’15
3. Viswanathan et al., OSDI’16
4. Hsieh et al., NSDI’17
Processing queries in-place

Bottleneck site exists.
## Processing queries in-place

<table>
<thead>
<tr>
<th>System</th>
<th>Workload</th>
<th>Task / Data Placement</th>
<th>Optimize aspect</th>
</tr>
</thead>
<tbody>
<tr>
<td>JetStream</td>
<td>streaming</td>
<td>no</td>
<td>WAN bandwidth usage</td>
</tr>
<tr>
<td>Geode</td>
<td>batch</td>
<td>data</td>
<td>WAN bandwidth usage</td>
</tr>
<tr>
<td>Iridium</td>
<td>batch</td>
<td>both</td>
<td>query response time</td>
</tr>
</tbody>
</table>
Processing queries in-place

Iridium (Pu et al. SIGCOMM’15)

- **Recurring** queries, abundant computation resource.
- Transfer data out from the bottleneck site.
- Re-schedule reducer tasks.
Iridium can be improved

- similarity oblivious
- data reduction ratio is constant
Iridium can be improved

- similarity oblivious
- data reduction ratio is constant

<table>
<thead>
<tr>
<th>Site</th>
<th>Intermediate record</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tokyo</td>
<td>2</td>
</tr>
<tr>
<td>Oregon</td>
<td>2</td>
</tr>
<tr>
<td>Seoul</td>
<td>3</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>7</strong></td>
</tr>
</tbody>
</table>
Iridium can be improved

• similarity oblivious

• data reduction ratio is constant

<table>
<thead>
<tr>
<th>Site</th>
<th>Intermediate record</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tokyo</td>
<td>2</td>
</tr>
<tr>
<td>Oregon</td>
<td>1</td>
</tr>
<tr>
<td>Seoul</td>
<td>2</td>
</tr>
<tr>
<td>Total</td>
<td>5</td>
</tr>
</tbody>
</table>
Bohr Design

Key ideas:

• **Recurring** queries, abundant computation resource.

• **Similarity aware** data transfer.

• **Measuring and predicting** the data reduction ratio.
Bohr Design

Site 1:
- OLAP Cube Generation
  - OLAP Cube 1
    - Map 1
      - Combine enabled
    - Reduce 1

Site 2:
- OLAP Cube Generation
  - OLAP Cube 2
    - Map 2
      - Combine enabled
    - Reduce 2

Probes:
- probe
- similar dataset

Global Manager:
- Job Queue

Final result:
- query tasks
Why using OLAP Cube?

- Sorting dataset according to the attribute.
- Using record level similarity score.
Why using OLAP Cube?

Advantages:

• Format the unformatted raw data.

• Multiple dimension cube for one dataset.
Similarity Search & Check

Search:

• OLAP instruction (e.g., dice, slice, roll up) to retrieve the needed attributes.

Check:

• Cross site check: simple probes.
• The probe components.
Reduce tasks scheduling

\[ t : \text{data transfer time.} \quad I : \text{input data size.} \]
\[ R : \text{data reduction ratio.} \quad U : \text{uplink bandwidth.} \]
\[ D : \text{downlink bandwidth.} \quad r : \text{reduce tasks percentage.} \]

\[ \min \quad t \quad \quad (1) \]

Goal: Minimize the data transfer time

\[ (1 - r_i)I_i R_i / U_i \leq t, \quad \forall i, \quad (2) \]

Time to upload data from site i.

\[ r_i (\sum_{j=1}^{N} I_j R_j - I_i R_i) / D_i \leq t, \quad \forall i, \quad (3) \]

Time to download data from other sites.

\[ \sum_i r_i = 1, \quad r_i \geq 0, \quad \forall i. \quad (4) \]
System Implementation

Based on Spark v2.1.0 and Iridium.

Utilize Apache Kylin OLAP data cube to deal with all the OLAP operation.

Use Gurobi Optimizer for optimize the LP function.
Evaluation Methodology

• **Workloads:**
  - TPC-DS.
  - Amplab big data benchmark.

• **Hardware platform:**
  
  A real EC2 deployment with 10 sites (Singapore, Tokyo...).
Evaluation Methodology

- **Baseline:**
  - Iridium (Pu et al. Sigcomm’ 15).

- **Performance metrics:**
  - Query completion time.
  - Data reduction ratio.
Bohr can increase the **data reduction ratio** in all sites.
Bohr saves 30% QCT compare to Iridium.
We propose Bohr:

- **Similarity aware** data transfer.
- **Predicting and measuring** data reduction ratio for each site.

Bohr is **30%** faster than Iridium.
Thank you!
Future work

1. Generalize the basic idea to more workloads.

2. Dealing with the overhead bring by operate OLAP cubes.

3. Break recurring batch workload condition

4. Multiple types of queries accessing one dataset.