Distributed Querying over Compressed Property Graphs

Swapnil Gandhi, Sayandip Sarkar, Abhilash Sharma and Yogesh Simmhan

**INTRODUCTION**

- Increasing trend toward representing semi-structured data as **Property Graphs** using compact in-memory store.
- **Challenges:**
  - **Low latency** interactive querying on compressed data
  - **Scaling** to Large Graphs (Billions of Vertices and Edges)
  - **High Throughput** Graphs Keep Growing in Size

**GoDB**

- Builds upon GoFFish subgraph-centric batch processing framework
- **Bulk Sync Parallel** Execution Model
- Common graph query types: VE, Path, BFS, Reachability
- Heuristics based **Cost Model** to select best distributed query execution plan

**GoDB-X**

- Harnesses query execution model of GoDB
- **Uses Succinct** as an underlying Data Store
- Reduces memory footprint of Java objects but incurs higher traversal costs
- Offers an interesting **trade-off** between distributed memory utilized and query latency time

**DATA-MODEL**

- **Critical Feature:** Storage of incidence and properties by vertex/edge pairs
- **Attributes:**
  - **ID:** Unique, Deterministic
  - **Weight:**

**DATASET**

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Vertices</th>
<th>Edges</th>
<th>Properties</th>
<th>On-disk Size</th>
<th>Number of Select Queries</th>
<th>Number of Report Queries</th>
</tr>
</thead>
<tbody>
<tr>
<td>CTP</td>
<td>3.7 M</td>
<td>16.5 M</td>
<td>2</td>
<td>1 GB</td>
<td>300</td>
<td>300</td>
</tr>
<tr>
<td>GTR</td>
<td>170 M</td>
<td>1.2 B</td>
<td>4</td>
<td>38 GB</td>
<td>120</td>
<td>480</td>
</tr>
</tbody>
</table>

**EXPERIMENTAL EVALUATION**

- Space-time trade-off of Succinct Data Store for different Sampling Rate w. We adopt w=0.2 in our experiments

**CITP DATASET**

- Though by using compressed data model we were able to see >75% Reduction in Query Execution Time for Centralized GoDB-X. For distributed (4 VMs) GoDB-X we see execution time to be marginally slower than GoDB. Static Overheads of querying compressed data + network communication cost out-weigh any advantages offered by parallel execution.

**GTR DATASET**

- >50% Reduction in Query Execution Time for Distributed GoDB-X. As more operations per Machine are required for GTR, we see advantages of parallel execution out-weighting communication costs and static overheads.

**ON-GOING WORK**

- **Improvements in Data Model**
  - **Cost Model Inclusion**
  - Support query over Time-Series Graph with compound predicates
  - Concurrent Queries Optimization

**SUMMARY**

- Fewer Machines ➔ Smaller Memory Footprint ➔ Reduced Communication Costs ➔ Lower Query Latency