**Motivation**

Social Networks, Internet of Things and Business intelligence applications model data as RDF graphs and query with SPARQL query language.

**Heterogeneity:** selective and non-selective queries

Problem: inefficient query processing on massive data parallelism and lack of execution isolation leads to 1) sub-optimal response time and 2) workload interference.

**Graph Store**

- **Predicate-based grouping:** partition the key space into multiple segments, which are identified by the combination of predicate and direction (e.g., p(id.d)).
- **Caching RDF store:** splits each segment into multiple fixed-size blocks and allows to store them into discontinuous regions of the cache on GPU.
- **Replacement policy:** uses a look-ahead LRU-based policy to decide where to store the new prefetched value and key blocks.

**Distributed Processing**

- **Parallel sub-query generation:** leverages GPU to fast break history tables
- **Direct sub-query distribution:** adopts GPUDirect RDMA to avoid unnecessary data copy

**Computation Model**

Basic approach:
- Leverage GPUs to exploit data parallelism in non-selective queries

**Query Execution**

How we improve memory and time efficiency step by step:
- **all graph scale:** potential GPU memory overflow
- **per-query scale:** only cache the necessary data retained in GPU memory before running a query.
- **per-pattern scale:** only prefetches the triples with a certain predicate used by the next triple pattern.
- **Pipeline:** overlap the data movement and query execution time
- **per-piece scale:** further split predicates into multiple fixed-size blocks and cached them in a best-effort way.

**Case Study**

**Background**

We use Lehigh University Benchmark (LUBM) which includes both selective queries and non-selective queries. We compare our system with the state-of-art systems Wukong and TriAD.

**Settings**

- **Hardware Settings**
  - 5-node cluster, 12 cores each
  - 56Gbps InfiniBand NIC

**Benchmark Settings**

- Single query latencies
- Mixed concurrent latency CDF

**Results**

<table>
<thead>
<tr>
<th>Performance (msec) on LUBM-1024D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test</td>
</tr>
<tr>
<td>------</td>
</tr>
<tr>
<td>Time</td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

**Summary**

**Wukong+G:**
1. GPU-based RDF Query Execution
2. GPU-friendly RDF Graph Store
3. GPU & RDMA-accelerated Query Distribution

*This work was done while the author was in SJTU.*