Fast and Concurrent RDF Queries with RDMA-based Distributed Graph Exploration

Jiaxin Shi, Youyung Yao, Rong Chen, Haibo Chen and Feifei Li

Overview

Wukong: a distributed in-memory RDF store that leverages RDMA-based graph exploration to support fast and concurrent SPARQL queries
- Graph Store: store index vertex, differentiate partitioning, predicate-based KV store
- Query Engine: full-history pruning, in-place/ftk-join execution, work-obligation scheduling
- Communication: one-sided RDMA ops

Architecture

Query Processing

Full-history Pruning
- Observation: the latency of RDMA is relatively insensitive to payload sizes (e.g., 8bytes: 1.56us vs. 2KB: 2.25us)
- Prune non-matching results early
- Avoid final centralized join

Dynamic Execution Mode Switch
- In-place mode: migrate data (ultra-low latency)
- Fork-join mode: migrate execution (parallelism)
- Make decisions in runtime according to #RDMA-ops

Evaluation

Wukong outperforms state-of-the-art systems for both latency and throughput, usually at the scale of orders of magnitude

- Setting:
  - A 6-node cluster (each: 20 cores, 64GB DRAM, 2x1B)
  - Benchmark: LUBM, WSDTS, DBP58, YAGO2
  - Baseline: THAD, Trinity.RDF, RDF3X, BitMat, etc.

Factor Analysis
- BASE: Trinity.RDF-like
- DRDMA: RDMA comm.
- +FIP: enable Full-history
- +IDX: Index-vertex
- +PBS: predicate-based KV
- +DYN: mode switch

A study on concurrent queries processing
- Mixture workload
- 26KB queries/sec
- 99m latency: 5.9ms

Data Model

Graph Model and Indexes & Differentiated Graph Partitioning
- Normal Vertex: randomly assign vertex with all edges (edge-cut)
- (Type & Predicate) Index Vertex: split and replicated (vertex-cut)

RDMA-friendly Predicate-based Store
- Derived from DTM-KV
- Separate a vertex into multiple KV pairs according to predicate

Graph Exploration

Sample RDF & SPARQL

SELECT ?X ?Y ?Z WHERE {
  ?Y takesCourse ?Z .
}

Results

Background

Social Networks, IoT and Business Intelligence apps model data as RDF Graphs and query with SPARQL query language

Existing Solutions:
- Triple join costly join ops
- Redundant im. results
- One-step pruning
- Final centralized join

Setting

Graph exploration
- M: 16 nodes cluster (each: 20 cores, 64GB DRAM, 2x1B)
- Benchmark: LUBM, WSDTS, DBP58, YAGO2
- Baseline: THAD, Trinity.RDF, RDF3X, BitMat, etc.

Factor Analysis
- BASE: Trinity.RDF-like
- RDMA: RDMA comm.
- +FIP: enable Full-history
- +IDX: Index-vertex
- +PBS: predicate-based KV
- +DYN: mode switch

A study on concurrent queries processing
- Mixture workload
- 26KB queries/sec
- 99m latency: 5.9ms

Wukong: a distributed in-memory RDF store that leverages RDMA-based graph exploration to support fast and concurrent SPARQL queries

- Graph Store: index vertex, differentiate partitioning, predicate-based KV store
- Query Engine: full-history pruning, in-place/ftk-join execution, work-obligation scheduling
- Communication: one-sided RDMA ops

Architecture

Query Processing

Full-history Pruning
- Observation: the latency of RDMA is relatively insensitive to payload sizes (e.g., 8bytes: 1.56us vs. 2KB: 2.25us)
- Prune non-matching results early
- Avoid final centralized join

Dynamic Execution Mode Switch
- In-place mode: migrate data (ultra-low latency)
- Fork-join mode: migrate execution (parallelism)
- Make decisions in runtime according to #RDMA-ops

Evaluation

Wukong outperforms state-of-the-art systems for both latency and throughput, usually at the scale of orders of magnitude

- Setting:
  - A 6-node cluster (each: 20 cores, 64GB DRAM, 2x1B)
  - Benchmark: LUBM, WSDTS, DBP58, YAGO2
  - Baseline: THAD, Trinity.RDF, RDF3X, BitMat, etc.

Factor Analysis
- BASE: Trinity.RDF-like
- RDMA: RDMA comm.
- +FIP: enable Full-history
- +IDX: Index-vertex
- +PBS: predicate-based KV
- +DYN: mode switch

A study on concurrent queries processing
- Mixture workload
- 26KB queries/sec
- 99m latency: 5.9ms