Introduction

- Growing interest in exploiting RDF data for decision-making
- Need: Scalable and cost-effective processing techniques
- MapReduce based parallel data processing
- Ad hoc analysis on extremely large data sets
- Systems supporting high-level data flow languages
- BUT support relational style operators for structured data processing workloads with few join operations

RDF Graph Processing on MapReduce

- SPARQL Graph Pattern Matching
- Join-intensive!!!
  - Query: Retrieve Vendor and Review details of product
  - Offers delivered within three days
- MapReduce computational model with map and reduce phases
  - Multiple join workloads
  - Multiple MR cycles
  - High I/O and communication costs

Algebraic Optimization

- Re-interpret star-joins SJ1, SJ2, SJ3 using grouping operation resulting in a set of TripleGroups
  - Computed in 1 MR cycle vs. 3 MR cycles
- An intermediate Nested TripleGroup Algebra (NTGA) that supports special operators for TripleGroup based processing of RDF graphs
- Compact representation for intermediate data using RDFMap
- Extended Pig’s computational infrastructure to support NTGA operators – RAPID+

Graph Pattern Matching using NTGA

- Data model based on nested TripleGroups - more naturally capture graphs
- TripleGroup based Operators – loadFilter, StarGroupFilter, RDFJoin etc.
- Re-factored and coalesced to minimize MR cycles

RDF Graph Processing on MapReduce

- RDFMap – property-based index
  - RDFMap – property-based index

Mapping NTGA → Relational Algebra (lossless transformation)

- Comparative evaluation
  - BSBM benchmark dataset with RAPID+, Pig, Pig_Opt
  - Up to 60% performance gain with NTGA operators

Ongoing and Future Work

- Extending the optimizations to grouping and aggregation phases
- Structure and cost based optimizations

References