Scaling Queries over Big RDF Graphs with Semantic Hash Partitioning
Kisung Lee, Ling Liu (Georgia Institute of Technology)

Background
- RDF (Resource Description Framework) is a standard graph-based model for data exchange on the Web and being widely used in many scientific projects, governments, etc.
- SPARQL is a standard query language for RDF and its processing is basically to find a set of sub-graphs satisfying the given graph pattern

Huge and growing size of RDF data makes it hard to store and handle the data on a single machine
- High correlation among data entities (vertices) makes it hard to parallelize the query processing
- Skewed distribution (many high degree vertices) makes it hard to ensure load balancing

Motivation

Example: RDF & SPARQL

Goals
- Improve distributed RDF query processing performance by maximizing local processing and minimizing cross-node communication

Contributions
- Scalable partitioning technique through controlled triple replication
- Efficient distributed query processing technique by minimizing the cross-node communication cost
- Validation through extensive experiments using several real-world and benchmark datasets

SHAPE: Semantic H-ash Partitioning-Enabled RDF System

1. Triple Groups
- Each has an anchor vertex and a set of triples (edges) connected to the anchor

2. Baseline Hash Partitions
- Grouping the triple groups to generate baseline hash partitions

3. Semantic Hash Partitions
- k-hop expansion: expand each hash partition

Experiments
- 21 machines (Hadoop v1.0.4)
- RDF-3X v0.3.5 as local RDF system

Observation 1: SHAPE is faster than the other partitioning techniques for all benchmark queries
Observation 2: SHAPE significantly reduces the graph partitioning and loading time
Observation 3: The decrease of the query processing time is almost proportional to the number of slave servers