Scaling Queries over Big RDF Graphs with Semantic Hash Partitioning

Kisung Lee, Ling Liu (Georgia Institute of Technology)

Motivation

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

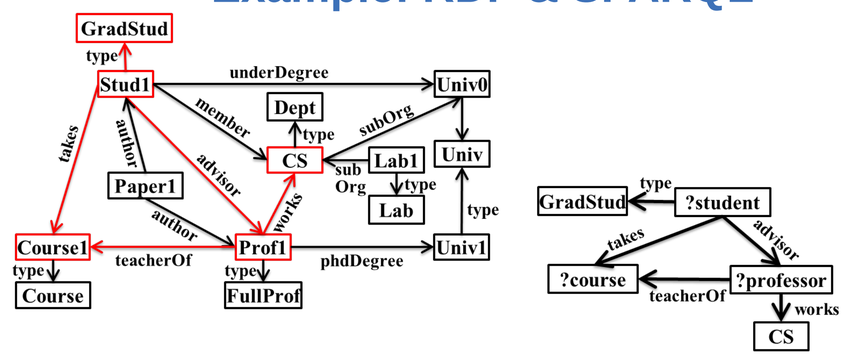
Challenges

➢**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

Example: RDF & SPARQL



SHAPE: Semantic HAsh Partitioning-Enabled RDF System

Goals

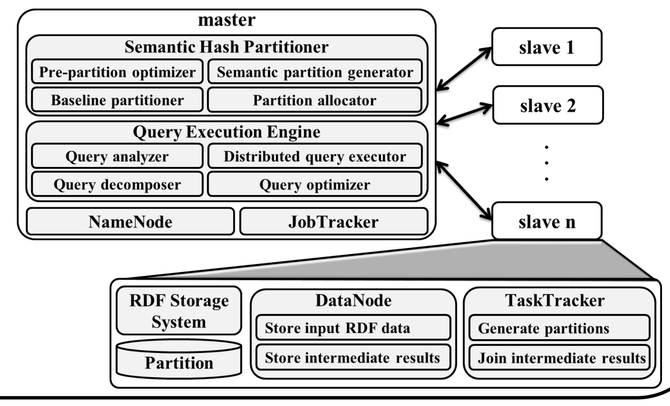
➢Improve distributed RDF query processing performance by **maximizing** local processing and **minimizing** cross-node communication

Contributions

➢**Scalable partitioning** technique through controlled triple replication

➢**Efficient distributed query processing** technique by minimizing the cross-node communication cost

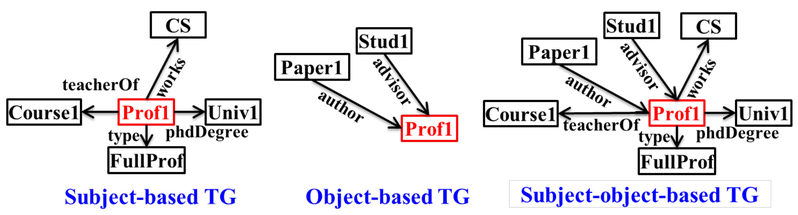
➢Validation through **extensive experiments** using several real-world and benchmark datasets



Semantic Hash Partitioning

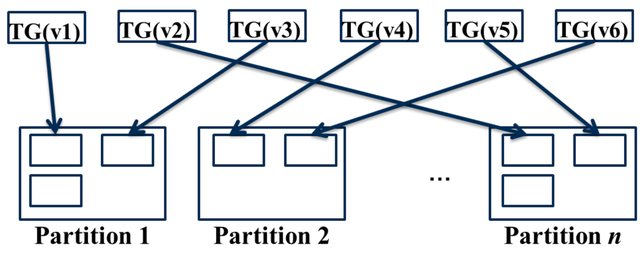
1. Triple Groups

➢Each has an **anchor vertex** and **a set of triples** (edges) connected to the anchor



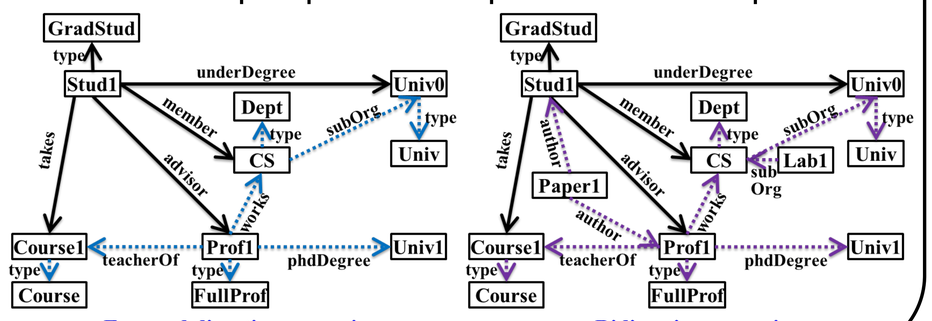
2. Baseline Hash Partitions

➢Grouping the triple groups to generate baseline hash partitions



3. Semantic Hash Partitions

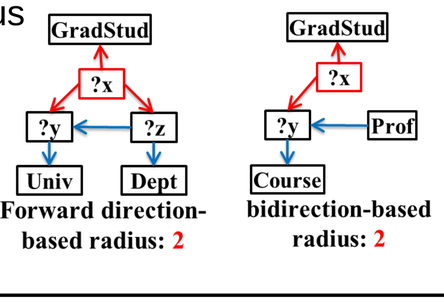
➢*k*-hop expansion: expand each hash partition



Distributed Query Processing

1. Query Analysis

➢Determines whether a given query can be executed without cross-node communication using its radius



2. Query Decomposition

➢Splits the query into a set of sub-queries

3. Distributed Query Execution

➢Executes the query on the semantic hash partitions

Experiments

➢21 machines (Hadoop v1.0.4)

➢RDF-3X v0.3.5 as local RDF system

Dataset	#triples	#subjects
LUBM267M	267M	43M
LUBM534M	534M	87M
LUBM1068M	1068M	174M
SP2B200M	200M	36M
SP2B500M	500M	94M
SP2B1000M	1000M	190M
DBLP	57M	3M
Freebase	101M	23M

➢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

This work is partially sponsored by NSF under Grants IIS-0905493, CNS-1115375, IIP-1230740 and a grant from Intel ISTC on Cloud Computing

Intel Science & Technology Center for Cloud Computing

Carnegie Mellon University · Georgia Tech · intel · PRINCETON UNIVERSITY · UC Berkeley · UNIVERSITY of WASHINGTON