Semantics in the Cloud

Scalable Distributed Computing for the Semantic Web and the SHARD Triple-Store

Kurt Rohloff
krohloff@bbn.com
@avometric

Many thanks to:
Gail Mitchell, Doug Reid from BBN
Hanspeter Pfister from Harvard SEAS
Prakash Manghwani
Why?

• Triple-Store Study:
  – “An Evaluation of Triple-Store Technologies for Large Data Stores”, SSWS ’07 (Part of OTM)
  – Great help from OntoText, Franz

• Design Goals (not just scalability!):
  – Scalable – avoid monolithic resource limitations.
  – High Assurance – maintain QoS despite major failures.
  – Cost Effective – only commodity hardware.
  – Modular – strong data separation to maintain provenance
Cloud/Grid/Utility Computing?

- Cloud computing means different things depending on where you play in the stack:
  - Services: PayPal, Google Search
  - Solutions: Google App Engines
  - Storage: Rackspace Cloud Files
  - Infrastructure: Amazon EC2
A Map-Reduce Implementation

- Open implementation of Google’s tech.
  - Developed from Google publications.
  - Heavily pushed by Yahoo, Facebook, etc…
  
  http://hadoop.apache.org/

- Cloudera has great training material
  - Look for VMWare training virtual machine
  
  http://www.cloudera.com/
Some Big Numbers

- Yahoo! Hadoop Clusters: > 82PB, >25k machines (HadoopWorld NYC ’09)
- Google: 40 GB/s GFS read/write load (Jeff Dean, LADIS ’09) [~3,500 TB/day]
- Facebook: 4TB new data per day; DW: 4800 cores, 5.5 PB (Dhruba Borthakur, HadoopWorld)
Map-Reduce, Functionally

• A cloud computing model
• 2 epochs, each run concurrently over many machines:
  Map: split each input line into little pieces of data
  Reduce: recombine little pieces
• There are other models…

Text input:
Cannon to right of them,
Cannon to left of them,
Cannon behind them
New Datastore Models

- File System
  (HDFS: Hadoop Dist. File System)
- Flat Files
- Bigtable, Dynamo, Cassandra, ...
- Triple-Stores
- Database
General Programming of These Systems...

From Experience:

• Inherently multi-threaded
• Toolset still young
  – Not many debugging tools
• Mental models are different...
  – Learn an algorithm, adapt it to M/R
Map-Reduce Triple-Store Proof of Concept

STAND BACK

I'M GOING TO TRY SCIENCE
SHARD Triple-Store

SemWeb Triple Store Built on Hadoop

Design Goals:
• Scalable
• Robust
• Commodity Hardware
More Specifically

• Cloud-based triple-store on HDFS
  – Massively scalable
• SPARQL queries
  – LUBM proof-of-concept
• Basic inferencing
  – subClassOf, subPropertyOf
• Java API
  – Method calls at client
  – Processing in cloud
  – Move results to local machine
HDFS, Physically

Local

Cloud

Client

Name Node

Node 1

Node 2

Node 3

Node 4

Cannon Right
Cannon Left
Cannon Behind

Cannon Right
Cannon Left
Cannon Behind
Robustness?

• Datanode crash?
  – Clients read another copy
  – Background rebalance
• Task fails - Try again
  – Retries possible because of idempotence
• Namenode crash?
  – uh-oh
Triple-Store Operations

• Load data (i.e. select data)
• Persist data (i.e. save to disk)
• Reload triple-store (i.e. restart)
• Run inferencing
• Respond to queries
Query Overview

SELECT ?person
WHERE {
  ?car a :Car .
  ?car :madeIn :Detroit
}

Map: Assign variables for 1st clause
Reduce: Remove duplicates

Map:
1. Assign variables for next clause
2. Map past partial assignments, Key on common variable
Reduce:
1. Join partial assignments on common variable
2. Remove duplicates

Iterate over clauses

Map: Filter on SELECT variables
Reduce: Remove duplicates
Graph Data

- Kurt owns car0
- car0 madeBy Ford
- Kurt livesIn Cambridge
- Ford madeIn Detroit
- Detroit a City
Query Processing

• Initially using BBN-developed query processor
  – Starting interface with Jena
  – Sesame looks feasible.

• SHARD supports “most” of SPARQL.
  – Many unimplemented portions could be handled by query translator.
  – Large performance improvements possible with improved query processing.
SPARQL Query

All people who own a car made in Detroit:
SELECT ?person
WHERE {
  ?car a :Car .
  ?car :madeIn :Detroit .
}
Query Matching

Kurt owns car0 madeIn Detroit livesIn Cambridge a City

?person owns ?car a Car madeIn Detroit

Ford madeBy car0

Detroit a City
MR Triple Store

Triple Store is simple list of triples in HDFS

Triple Store

1\textsuperscript{st} clause
results

2\textsuperscript{nd} clause
results

1\textsuperscript{st} clause
results

2\textsuperscript{nd} clause
results
Test Data

- Standard LUBM benchmark data
  - Artificial data on students, professors, courses, etc… at universities
- Deployed code on Amazon EC2 cloud
  - 19 XL nodes
- 6000 university dataset
  - Approximately 800 million edges in graph
- In general, performed comparably to “industrial” monolithic triple-stores
- Proof o’ Concept: For 6000 universities (approx. 800 million triples):
  Query 1: 404 sec. (approx 0.1 hr.)
  Query 9: 740 sec. (approx 0.2 hr.)
  Query 14: 118 sec. (approx 0.03 hr.)
- Sesame+DAMLDB:
  Query 1: approx 0.1hr,
  Query 9: approx 1 hr
  Query 14: approx. 1 hr
- Jena+DAMLDB for 550 million triples:
  Query 1: approx 0.001 hr,
  Query 9: approx 1 hr
  Query 14: approx. 5 hr
Deficiencies?
Ongoing Research?
Optimizations

• For a single query....
  For a single workflow...
  Across workflows...

• Bring out last century’s DB research! (joins)
  And file system research too! (RAID)

• HadoopDB (Yale)

• Data Formats (yes, in ’10)
Release plans

- Tentative Open-Source release
  - BSD license planned
Thanks!
Questions?

Kurt Rohloff
krohloff@bbn.com
@avometric