Cloud computing for Scalability:
The SHARD Triple-Store

Kurt Rohloff
krohloff@bbn.com
@avometric

Many thanks to:
Numerous colleagues from BBN
Hanspeter Pfister from Harvard SEAS
Phil Zeyliger from Cloudera
A Preface

SHARD is released open-source.

• BSD license.
• Look at:
  – My webpage (Search for “SHARD krohloff”)
  – Sourceforge (SHARD-3store)
• Use svn to get code:
  
  svn co https://shard-3store.svn.sourceforge.net/svnroot/shard-3store shard-3store
  
  – Don’t worry - this command is on SourceForge!

Happy to talk offline cloud computing and SHARD

• Use of SHARD, open-source projects, etc…
Cloud/Grid/Utility Computing?

- Services: PayPal, Google Search
- Solutions: Google App Engines
- Storage: Rackspace Cloud Files
- Infrastructure: Amazon EC2
Map-Reduce Cloud Paradigm

• 2 epochs, each run concurrently over many machines:
  Map: split lines into little pieces of data.
  Reduce: new recombination of the little pieces.
• Can be made VERY parallel…
• Word count example:

Text input:
Cannon to right of them,
Cannon to left of them,
Cannon behind them

Cannon:1
To: 1
Right: 1
Of: 1
Them: 1

Cannon:1
To: 1
Left: 1
Of: 1
Them: 1

Cannon:1
Behind: 1
Them: 1

Cannon:3
To: 2
Right: 1
Left: 1
Of: 2
Them: 3

Cannon:1
Behind: 1
Them: 1

Cannon behind them

Cannon to right of them

Cannon to left of them,
A Map-Reduce Implementation

- Open implementation of Google’s tech.
  - Developed from Google publications.
  - VERY large-scale!
  
  http://hadoop.apache.org/

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

- Baked-in robustness makes it practical...
HDFS, Physically

Local
Client

Cloud
Name Node

Node 1
Node 2
Node 3
Node 4

Cannon Right
Cannon Left
Cannon Behind
General Programming of These Systems...

From Experience:

• Inherently multi-threaded.
  – Not many debugging tools.

• Mental models are different…
  – Learn an algorithm, adapt it to M/R.
Map-Reduce Triple-Store Concept
Sample of Relevant Triple-Stores

- Parliament by BBN (from DAPRA DAML.)
- OWLIM by OntoText (several versions.)
- Allegrograph from Franz.
- MySQL and Oracle Solutions.
- LarKC by DERI Galway.
- Hive- and Pig-based experimental triple-stores.
- Etc…
Scale Limitations!

• **Triple-Store Study:**
  – “An Evaluation of Triple-Store Technologies for Large Data Stores”, SSWS '07 (Part of OTM).

• **Benchmarks:**
  – LUBM (Lehigh Univ. BenchMark)
    • Artificial data on students, professors, courses, etc… at universities.
SHARD Design Overview

- Cloud-based triple-store on HDFS.
  - Method calls at client.
  - Processing in cloud via MapReduce jobs.
  - Move results to local machine.

- Massively scalable.
  - Commodity hardware.

- SPARQL queries.
  - Optimize for complex queries with large response sets.

- Basic inferencing.
SPARQL Query

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

Kurt owns car0 madeIn Detroit livesIn Cambridge

?person owns ?car

Car madeBy Ford madeIn Detroit a

City a
HDFS Graph Storage

Graphs saved as flat-file in HDFS:
(Portions of file saved on each data node.)

Kurt owns car0 livesIn Cambridge
Car0 a Car madeBy Ford madeIn Detroit
Cambridge a City
Detroit a City
Query Processing

• BBN-developed query processor.
  – Starting integration with “standard” interfaces
    • Jena, Sesame.

• SHARD supports “most” of SPARQL.
  – Like most commercial triple-stores.

• Large performance improvements possible with improved query reordering.
Iterative Query Response Construction

Source Data

1st clause results

2nd clause results

16
Test Data

• Deployed code on Amazon EC2 cloud.
  – 19 XL nodes.

• 6000 LUBM university dataset.
  – Approximately 800 million edges in graph.

• In general, performed comparably to “industrial” monolithic triple-stores.
Advice from SHARD

• Down to “bare metal” in HDFS for large-scale efficiency.
  – No Berkeley DB, no C-stores, …. Nothing.

• Simple data storage as flat files.
  – Lists of (predicate, object) pairs for every subject by line.
  – Ex: Kurt owns car0 livesin Cambridge

• Simple often really is better…

• Sometimes reinventing the wheel leads to more efficiency…
Thanks!
Questions?
Kurt Rohloff
krohloff@bbn.com
@avometric

Please contact me if you have any interest in using SHARD or becoming a committer!

We are Hiring!
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)
• This past year – see HadoopWorld, NYC ‘10
Hadoop Robustness?

- Datanode crash?
  - Clients read another copy.
  - Background rebalance.
- Task fails - Try again.
  - Retries possible because of idempotence.
- Namenode crash?
  - uh-oh.
Performance Comparison

• 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 0.1 hr,
  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