

Triplestore Testing in the Cloud with Clojure

Ryan Senior



About Me

- Senior Engineer at Revelytix Inc
- Revelytix Info
 - Strange Loop Sponsor
 - Semantic Web Company
 - <http://revelytix.com>
- Blog: <http://objectcommando.com/blog>
- Twitter: @objcmdo
- Email: senior<dot>ryan<at>gmail<dot>com

Overview

- 6 months of work in 20 minutes!
- Triplestores
 - Basics, varieties, benchmarks
- EC2
 - Rationale, pros/cons
- Testing
 - Testing library overview, code examples
- Incanter
 - Statistical analysis and graphs

Project Overview

- Why did we do this?
- HR-related project for the DoD
- Project is committed to semantic technologies
 - RDF, OWL and SPARQL
- Requirements fairly open ended
 - Datasets are going to get large
 - Queries are going to be complex
- Where do we store RDF data?
- How is it queried?

What is a Triplestore?

- Stores RDF Data
 - But what is RDF?
- Querying RDF Data with SPARQL
- Inferencing
- Why not a plain database?
 - See “Scalable Semantic Web Data...” [1]
- Persistence alternatives
 - Graph databases, trees, linked lists
 - Relational databases with proprietary wrappers

Comparing Triplestores

- Single user SPARQL benchmarks
 - SP² [2]
 - BSBM [3]
- Multi-user Benchmark
 - Revelytix created
 - Based on SP²
 - Scales from 1 to 64 connections
 - Read-only
- Other
 - Admin UIs, inferencing, deployment options, standards compliance

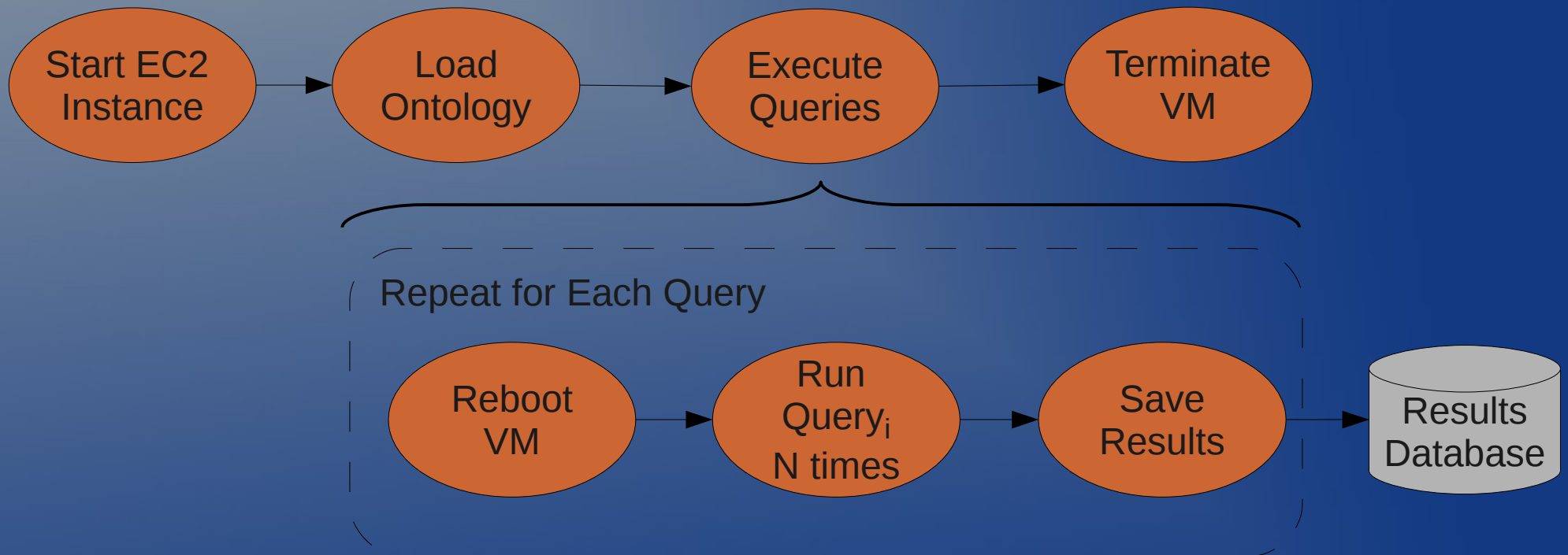
Triplestores are Immature

- No standard API
 - Jena and Sesame
 - No standard approach to integration with vendors
- No standard performance tuning (i.e. indexes)
- Batch Loading
 - Some local, others remote
 - Not all RDF serialized types supported
- Inferencing supported differently
- Bugs

Testing Using EC2

- Requirements
 - Many machines, for just a few months
 - High Memory
 - Reproducible System Configuration
- Pros (vs. Physical)
 - Low startup cost, reusable images
 - System lifecycle commands via HTTP
 - Similar to expected production environment
- Cons
 - Occasional outliers, slow IO
 - Calls repeatability of tests into question

Test Lifecycle



Clojure Testing Library

- Needs to support many
 - Triplestores, APIs, Configurations, Benchmarks
- Utilize Java Triplestore libraries
- EC2 Lifecycle Support
 - Launch, reboot, terminate instances
- Handle Failures/Timeouts robustly
- Save test definitions, input/output from tests
- Ability to generate graphs from results

Test Definitions

- Defines a test “template” in Clojure
 - Includes a benchmark, basic configurations
- Allows for “variants”
 - Specifies and/or overrides template config
 - Has triplestore/API/system specific info
- Stored directly in CouchDB
- Example Inputs
 - Triplestore, benchmark, dataset size
 - Client API, query timeout, query repetitions

Extension Points

- Test Fixtures
 - Internal
 - Capturing test running time
 - Capturing exceptions
 - External
 - Starting an EC2 instance
 - Loading data
- Test Points
 - Only External
 - Executing Queries

Implementation Info

- Clojure 1.2
- CouchDB 0.10.0
- clojure-couchdb
- clojure-http
- Incanter
- Leiningen
- Java Libraries
 - Jena, Sesame, AWS JDK

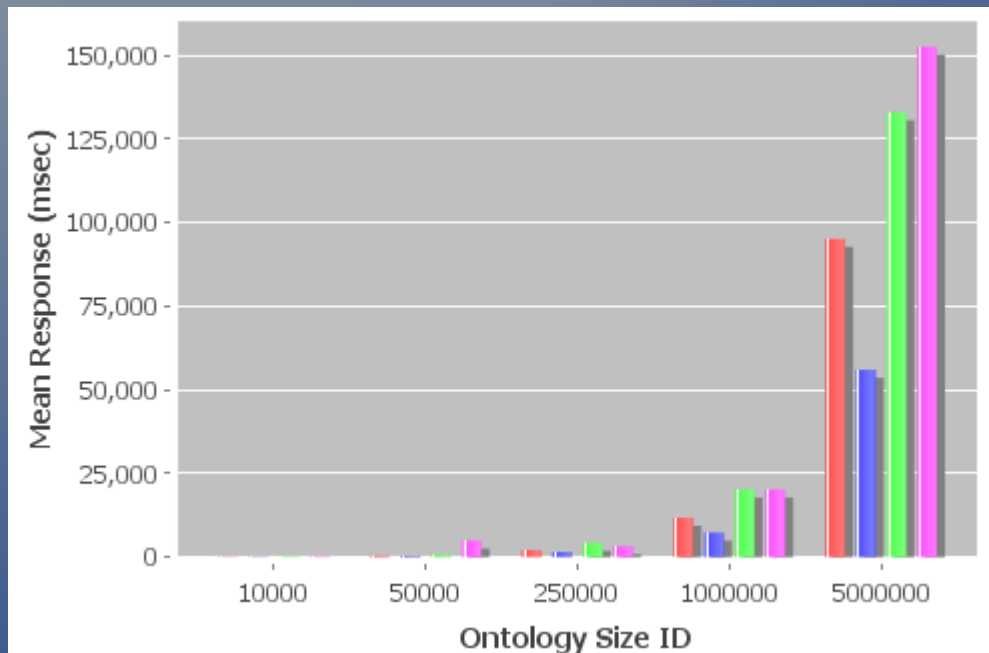
Code Examples

- Test Definition
- Test Result (from CouchDB)
- Test Fixtures
- Test Points
- Protocols
- Adding Timeouts

Incanter

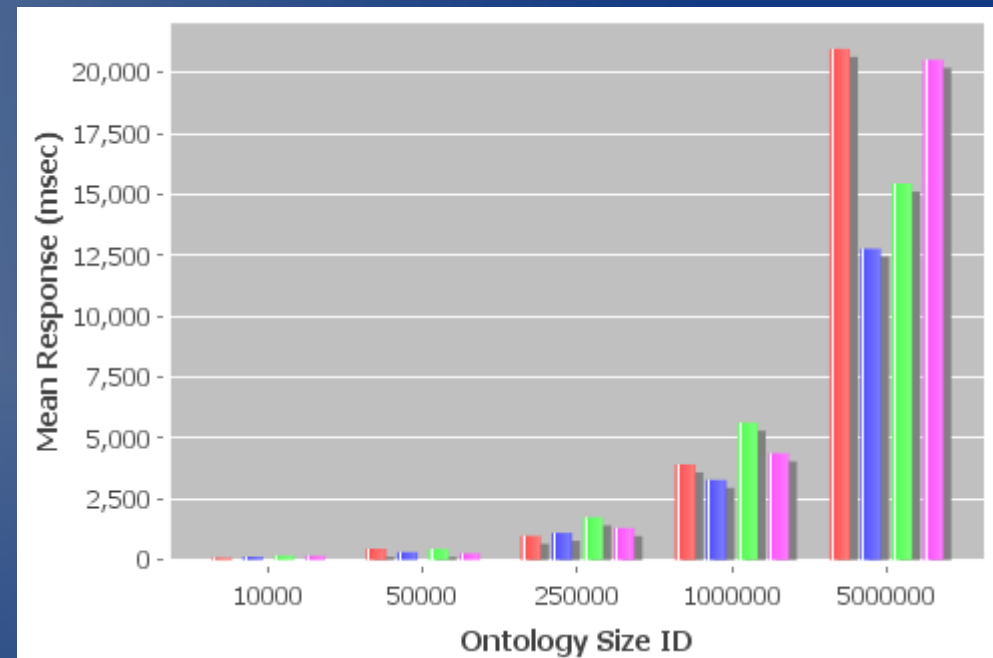
- What is it?
 - Statistical analysis and graphing in Clojure
- Used extensively in analyzing EC2 data
- Graphing comparisons of triplestores
- Refining data through statistical calculations
 - t-stat, mean, standard deviation
- Captured in a report for the DoD
- Example graphs
 - Names omitted to protect the innocent

SP² Benchmark Performance Testing Sample Results



Query 2

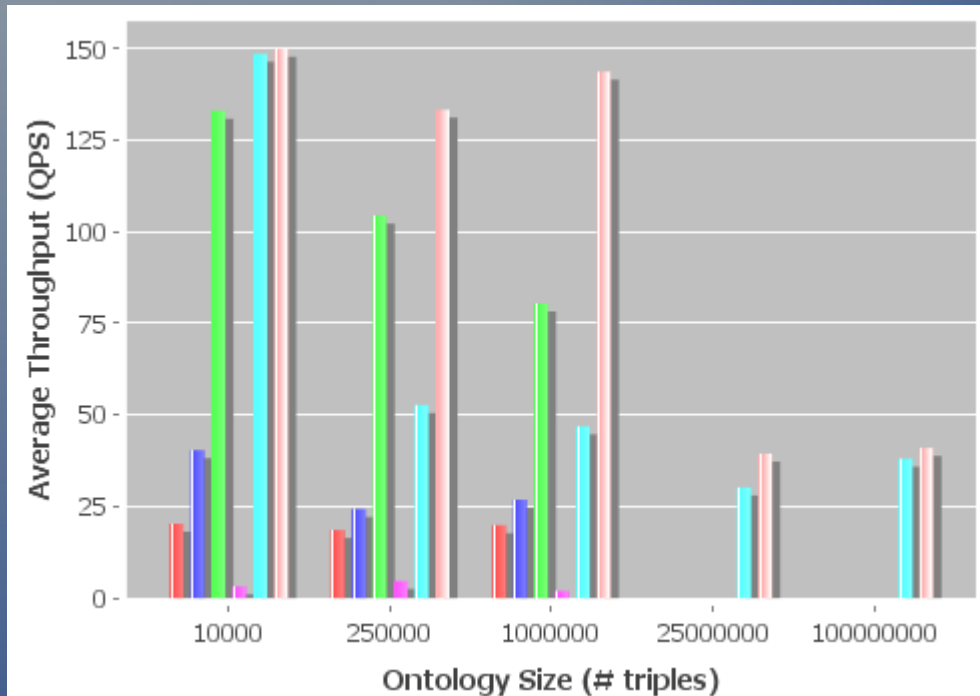
Grouped by triplestore



Query 3a

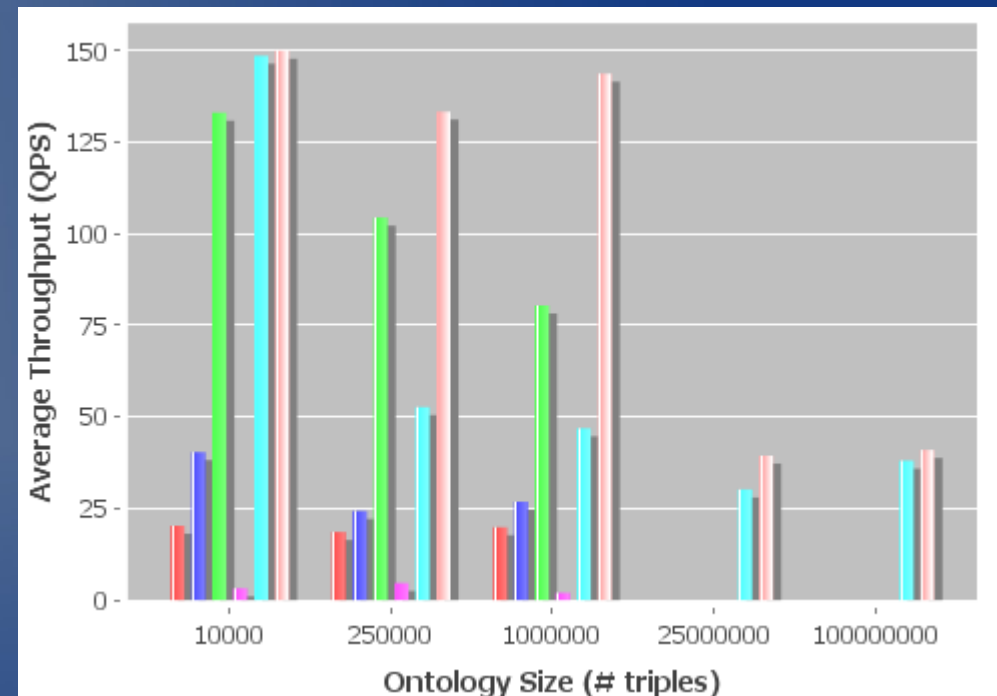
Grouped by triplestore

BSBM Benchmark Performance Testing Sample Results



Query 1

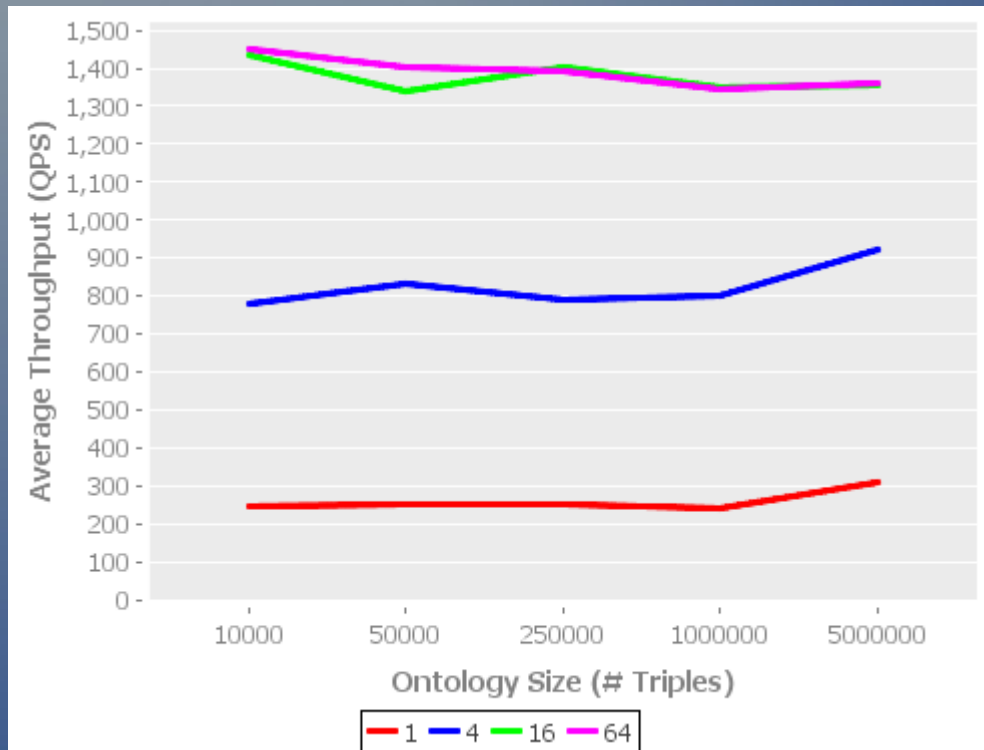
Grouped by triplestore



Query 2

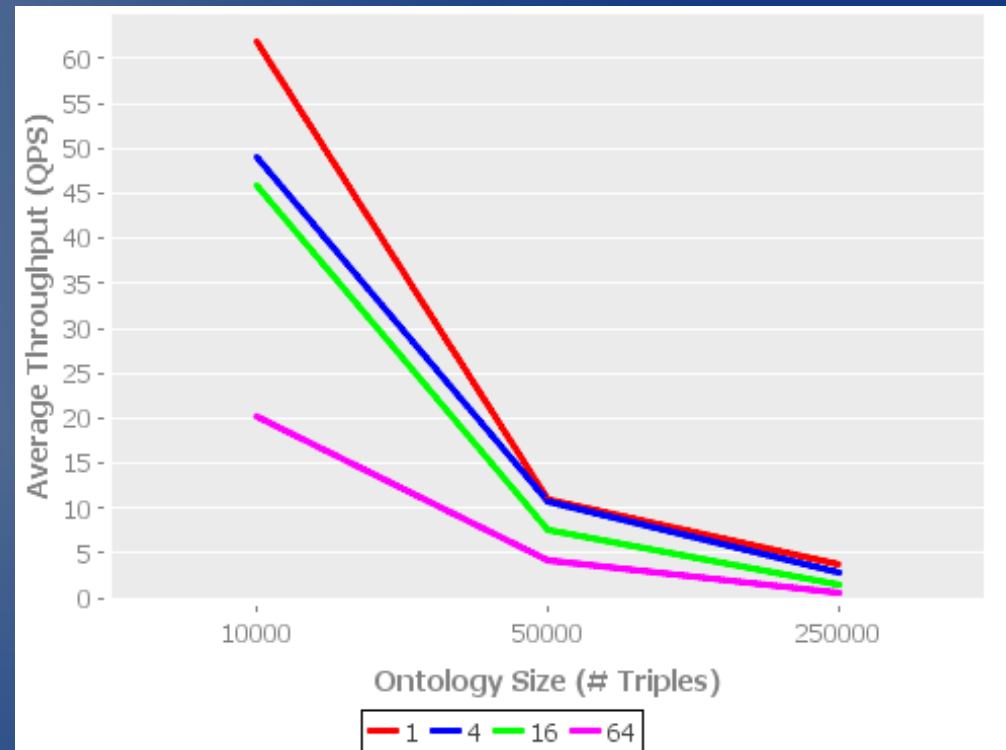
Grouped by triplestore

SP² Concurrent Performance Testing Sample Results



Query Family 1

Grouped by # connections



Query Family 8

Grouped by # connections

References

- 1 – Scalable Semantic Web Data Management Using Vertical Partitioning - <http://bit.ly/b2W5jY>
- 2 – SP² Benchmark - <http://bit.ly/bkYJ4>
- 3 – BSBM Benchmark - <http://bit.ly/K8nLA>