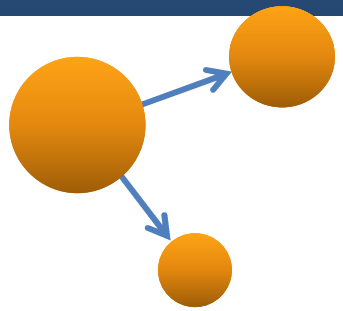
Optimized SPARQL performance management via native API



Date	April 29, 2014.
Team lead	Victor Chernov
E-Mail	vchernov@nitrosbase.com
Time Zone	MSK, UTC+4

Hackathon
Ontology Summit 2014

The Problem Statement and Objective

- Often it is hard to select rather complex data structure by one SPARQL query, but the same time pretty simple to get all the data using a combination of simple queries, and then assemble this complex structure at the application level.
- This may require hundreds of calls to database and hence using endpoint may be ineffective due to HTTP access overhead.
- In such cases it is good if manufacturer provides native API that allows running queries directly, bypassing HTTP.

Objective was to find out the triplestores performance on simple queries through native API.

Event and Participants

- **The event took place worldwide March 29th 2014 at 14:00 – 18:00 MSK with subsequent activities during the next day.**
- **Four people participated :**
  - **Victor Chernov, Russia, vchernov@nitrosbase.com;**
  - **Vladislav Golovkov, Russia, vgolovkov@nitrosbase.com;**
  - **Andrej Andrejev, Sweden, andrej.andrejev@it.uu.se;**
  - **Vladimir Salnikov, Russia, vladimir.salnikov@compilesoft.ru.**

Triplestore selection

- We selected the following triplestores:
  - Virtuoso Universal Server Release 7.1
  - Stardog 2.1.2
  - NitrosBase RDF Storage 1.0 Release Candidate
- Important advantages of the selected triplestores:
  - Very high performance on sp2bench
  - Linux and Windows versions
  - Native API for fast query processing

Benchmark queries

- We've designed an advanced set of queries based on SP²Bench benchmark.
- The queries stress the attention on:
  - Search the small range of values
  - Search the big range of values
  - Sorting
  - Aggregation

Computers

We've set up three computers.

- Computer 1: Intel Core i5-3570 CPU @3.40 Ghz; 32 Gb RAM; SSD: Corsair Force GS 240 Gb; OS:Windows 8.1 x64
- Computer 2 HP Compaq 8100, Intel Core i5; CPU @2.80 GHz, 8Gb RAM; HD: Hitachi ATA (232GB) OS:Windows Server 2008 R2 Standard SP1.
- Computer 3 Intel Core i5 3570, 3400MHz, 16 GB RAM; OCZ Vertex 3 Max 120GB SSD. OS:Windows 7 x64

Experimentation details

- We played with 25 mln triple datasets created by SP²Bench generation utility.
- On each run query was sent to server and full result was read.
- Each query
  - was run 10 times,
  - query performance measured and
  - median value has been taken as the result.

Results

The experiments led to the following conclusions:

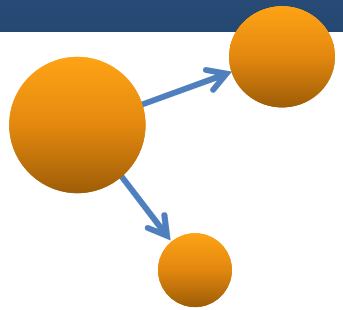
- For ontology applications it is desirable to have direct access to database, not through HTTP protocol.
- Sometimes it is worth to simplify the queries as much as possible and make some processing on the client. What is difficult to do with a single large query is easy to implement with a set of small ones. In those cases triplestore should be able to perform small queries quickly.
- Further performance gain (up to 10 times) could be reached giving the users direct access to database, bypassing SPARQL processing.
- The myth that RDF database is slower than SQL does not work anymore. RDF storages perform fast and can compete with SQL databases.

Final Report

The final report can be downloaded via the following URL:

http://nitrosbase.com/wp-content/uploads/OptimizedSPARQLreportV13.zip

9. Contacts



Victor Chernov
vchernov@nitrosbase.com
+7(985)999-22-43