FedX: A framework for efficiently evaluating SPARQL queries in a federated environment

CrEDIBLE working days, October 2013

Andreas Schwarte, fluid Operations AG
Outline

- Introduction
- Federated Query Processing
- Optimization techniques in FedX
- Experiments
- Application scenarios
- Experiences & Outlook
Motivation

Query processing involving multiple distributed data sources, e.g. Linked Open Data cloud

Query both data collections in an integrated way
Federated Query Processing

Federation mediator at the server

- Virtual integration of (remote) data sources
- Communication via SPARQL protocol
Federated Query Processing

Example Query from a General domain
Find US presidents and associated news articles

```
SELECT ?President ?Party ?TopicPage WHERE {
  ?President rdf:type dbpedia-yago:PresidentsOfTheUnitedStates .
  ?nytPresident owl:sameAs ?President .
}
```
Federated Query Processing

Query:

```sparql
SELECT ?President ?Party ?TopicPage WHERE {
  ?President rdf:type dbpedia-yago:PresidentsOfTheUnitedStates .
  ?nytPresident owl:sameAs ?President .
  ... 
}
```

Federation Mediator

?President rdf:type dbpedia-yago:PresidentsOfTheUnitedStates .

“Barack Obama”
“George W. Bush”
...
Federated Query Processing

Query:
SELECT ?President ?Party ?TopicPage WHERE {
  ?President rdf:type dbpedia-yago:PresidentsOfTheUnitedStates .
  ?nytPresident owl:sameAs ?President .
  ...}

?nytPresident owl:sameAs “Barack Obama” .

Input:
“Barack Obama”
“George W. Bush”
...

Output:
“Barack Obama”, yago:Obama
“Barack Obama”, nyt:Obama
Federated Query Processing

Query:

```sparql
SELECT ?President ?Party ?TopicPage WHERE {
  ?President rdf:type dbpedia-yago:PresidentsOfTheUnitedStates .

  ...}
```

Input:

```
“Barack Obama”
“George W. Bush”
...
```

Output:

```
“Barack Obama”, yago:Obama
“Barack Obama”, nyt:Obama
“George W. Bush”, nyt:Bush
```

... and so on for the other intermediate mappings and triple patterns ...
FedX Query Processing Model

Scenario:

- Efficient SPARQL query processing on multiple distributed sources
- Full SPARQL 1.1 support
- Data sources are known and accessible as SPARQL endpoints
  - FedX is designed to be fully compatible with SPARQL 1.0 endpoints
- No a-priori knowledge about data sources
  - No local preprocessing of the data sources required
  - No need for pre-computed statistics
- On-demand federation setup
- Read-Only scenarios
Challenges to Federated Query Processing

1) Involve only relevant sources in the evaluation
   Avoid: Subqueries are sent to all sources, although potentially irrelevant

2) Compute joins close to the data
   Avoid: All joins are executed locally in a nested loop fashion

3) Reduce remote communication
   Avoid: Nested loop join that causes many remote requests
Optimization Techniques

1. Source Selection:

Idea:

Triple patterns are annotated with relevant sources
- Sources that can contribute information for a particular triple pattern
- SPARQL ASK requests in conjunction with a local cache
  - After a warm-up period the cache learns the capabilities of the data sources
  ➔ During Source Selection remote requests can be avoided

2. Exclusive Groups:

Idea:

Group triple patterns with the same single relevant source
- Evaluation in a single (remote) subquery
- Push join to the endpoint
Example: Source Selection + Exclusive Groups

```
SELECT ?President ?Party ?TopicPage WHERE {
  ?President rdf:type dbpedia-yago:PresidentsOfTheUnitedStates .
  ?nytPresident owl:sameAs ?President .
}
```

Advantages:

- Avoid sending subqueries to sources that are not relevant
- Delegate joins to the endpoint by forming exclusive groups (i.e. executing the respective patterns in a single subquery)
3. Join Order:

Idea:

Iteratively determine the join order based on count-heuristic:

- Count free variables of triple patterns and groups
- Consider "resolved" variable mappings from earlier iteration

4. Bind Joins:

Idea:

Compute joins in a block nested loop fashion:

- Reduce the number of requests by "vectored" evaluation of a set of input bindings
- Renaming and Post-Processing technique for compliance with SPARQL 1.0
- Optional SPARQL 1.1 implementation using VALUES clause
Optimization Techniques (4)

**Example:** Bind Joins

```sparql
SELECT ?President ?Party ?TopicPage WHERE {
  ?President rdf:type dbpedia:PresidentsOfTheUnitedStates .
  ?nytPresident owl:sameAs ?President .
}
```

Assume that the following intermediate results have been computed as input for the last triple pattern:

**Block Input**

- “Barack Obama”
- “George W. Bush”
- ...

**Before (NLJ)**

```sparql
```

**Now:** Evaluation in a single remote request using a SPARQL UNION construct + local post processing.

SPARQL 1.1: VALUES clause implemented, but experiments show that UNION is more efficient
Optimization Example

1. **SPARQL Query**

   Compute *Micronutrients* using Drugbank and KEGG

   ```sparql
   SELECT ?drug ?title WHERE {
     ?drug drugbank:casRegistryNumber ?id .
     ?keggDrug bio2rdf:xRef ?id .
   }
   ```

2. **Unoptimized Internal Representation**

   ![Unoptimized Internal Representation Diagram]

   - 4x Local Join
   - 4x NLJ

3. **Optimized Internal Representation**

   ![Optimized Internal Representation Diagram]

   - Exlusive Group
   - Remote Join
Experiments

Based on FedBench benchmark suite
- 14 queries from the *Cross Domain* (CD) and *Life Science* (LS) collections
- Real-World Data from the Linked Open Data cloud
- Federation with 5 (CD) and 4 (LS) data sources
- Queries vary in complexity, size, structure, and sources involved

Benchmark environment
- HP Proliant 2GHz 4Core, 32GB RAM
- 20GB RAM for server (federation mediator)
- Local copies of the SPARQL endpoint to ensure reproducibility and reliability of the service
  - Provided by the FedBench Framework
Experiments (2)

a) Evaluation times of Cross Domain (CD) and Life Science (LS) queries
### b) Number of requests sent to the endpoints

<table>
<thead>
<tr>
<th>CD1</th>
<th>CD2</th>
<th>CD3</th>
<th>CD4</th>
<th>CD5</th>
<th>CD6</th>
<th>CD7</th>
<th>LS1</th>
<th>LS2</th>
<th>LS3</th>
<th>LS4</th>
<th>LS5</th>
<th>LS6</th>
<th>LS7</th>
</tr>
</thead>
<tbody>
<tr>
<td>27</td>
<td>22</td>
<td>(93,248)</td>
<td>(372,339)</td>
<td>(117,047)</td>
<td>6,183</td>
<td>1,883</td>
<td>13</td>
<td>61</td>
<td>(410)</td>
<td>21,281</td>
<td>16,621</td>
<td>130</td>
<td>(876)</td>
</tr>
<tr>
<td>x</td>
<td>5</td>
<td>(170,579)</td>
<td>22,331</td>
<td>247,343</td>
<td>x</td>
<td>x</td>
<td>1</td>
<td>x</td>
<td>101,386</td>
<td>3</td>
<td>2,666</td>
<td>98</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>2</td>
<td>23</td>
<td>38</td>
<td>18</td>
<td>185</td>
<td>138</td>
<td>1</td>
<td>18</td>
<td>2059</td>
<td>3</td>
<td>458</td>
<td>45</td>
<td></td>
</tr>
</tbody>
</table>

**Runtimes**

- **AliBaba:** $>600s$
- **DARQ:** $>600s$
- **FedX:** 0.109s
Application Scenarios

Bio2RDF scenario:

- 29 datasets with more than 4 billion triples integrated in the Information Workbench
- Structured queries, instance pages, and dashboards
- Example: PubMed publications, Trials, Diseases, etc.

Information Workbench with Bio2RDF federation

In this demonstrator we provide access to various Bio2RDF datasets (see list below) through a FedX federation. In total, this involves 29 data sets with more than four billion triples.

Overview of datasets

The datasets can be downloaded by clicking the link in the first column.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>SkosConcept</th>
<th>Instance type</th>
<th>Interlinking page</th>
<th>Example instance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Biocad</td>
<td>12,666,813</td>
<td>biocad-2-protein</td>
<td>Biocad Start</td>
<td>BCL-2 antigen, Bcl-2/adenovirus E3 Bcl-2-related protein</td>
</tr>
<tr>
<td>Cell Map</td>
<td>149,232</td>
<td>biocad-2-protein, biocad-2-pathway</td>
<td>Biocad Start</td>
<td></td>
</tr>
<tr>
<td>Chabi</td>
<td>649,481</td>
<td>sbioComp4-Concept</td>
<td>Chabi Start</td>
<td></td>
</tr>
<tr>
<td>Databases</td>
<td>153,320</td>
<td>sbioPharm-Drugs</td>
<td>Vizara</td>
<td></td>
</tr>
<tr>
<td>OpenData</td>
<td>70,512,494</td>
<td>dmpedia:Protein, dmpedia:Drug</td>
<td>Whispa</td>
<td></td>
</tr>
<tr>
<td>OAEaW</td>
<td>111,869</td>
<td>sbioComp4-Concept</td>
<td>OAEaW Start</td>
<td></td>
</tr>
<tr>
<td>OAEaW</td>
<td>75,502</td>
<td>Disease:organism, Disease:disorder</td>
<td>OAEaW Gene Ontology</td>
<td></td>
</tr>
<tr>
<td>OAEaW</td>
<td>517,023</td>
<td>drugbank:nc-drugs, drugbank:nc-targets</td>
<td>Drugbank Gene Ontology</td>
<td></td>
</tr>
<tr>
<td>OAEaW</td>
<td>16,583,157</td>
<td>biomedical:Gene</td>
<td>GeneSee Gene Ontology</td>
<td>FEK3</td>
</tr>
<tr>
<td>OpenBiomed</td>
<td>330,333</td>
<td>sbioComp4-Concept</td>
<td>OpenBiomed</td>
<td></td>
</tr>
<tr>
<td>GeneWise</td>
<td>1,024,877</td>
<td>GeneWise</td>
<td>GeneWise Gene Ontology</td>
<td></td>
</tr>
<tr>
<td>lHCC</td>
<td>22,463,395</td>
<td>lHCC</td>
<td>lHCC Start</td>
<td></td>
</tr>
<tr>
<td>lHCC</td>
<td>1,061,257</td>
<td>lHCC:2-protein</td>
<td>Cyto-dependent kinase inhibitor 1</td>
<td></td>
</tr>
<tr>
<td>lHCC</td>
<td>327,275</td>
<td>lHCC:2-protein</td>
<td>Cyclin-dependent kinase inhibitor</td>
<td></td>
</tr>
<tr>
<td>lHCC</td>
<td>83,148</td>
<td>lHCC:2-protein</td>
<td>Cyclin-dependent protein kinase 1</td>
<td></td>
</tr>
<tr>
<td>lHCC</td>
<td>16,609,323</td>
<td>lHCC:2-protein</td>
<td>3B protein-kinase activated nucleotide exchange factor</td>
<td></td>
</tr>
<tr>
<td>lHCC</td>
<td>2,360,956</td>
<td>lHCC:2-protein</td>
<td>3B protein-kinase activated nucleotide exchange factor</td>
<td></td>
</tr>
<tr>
<td>lHCC</td>
<td>316,077</td>
<td>lHCC:2-protein</td>
<td>3B protein-kinase activated nucleotide exchange factor</td>
<td></td>
</tr>
<tr>
<td>lHCC</td>
<td>7,031,916</td>
<td>lHCC:2-protein</td>
<td>3B protein-kinase activated nucleotide exchange factor</td>
<td></td>
</tr>
<tr>
<td>lHCC</td>
<td>2,841,213</td>
<td>lHCC:2-protein</td>
<td>3B protein-kinase activated nucleotide exchange factor</td>
<td></td>
</tr>
<tr>
<td>lHCC</td>
<td>2,783,613</td>
<td>lHCC:2-protein</td>
<td>3B protein-kinase activated nucleotide exchange factor</td>
<td></td>
</tr>
<tr>
<td>lHCC</td>
<td>810,746</td>
<td>lHCC:2-protein</td>
<td>3B protein-kinase activated nucleotide exchange factor</td>
<td></td>
</tr>
<tr>
<td>lHCC</td>
<td>544,373</td>
<td>lHCC:2-protein</td>
<td>3B protein-kinase activated nucleotide exchange factor</td>
<td></td>
</tr>
<tr>
<td>lHCC</td>
<td>3,781,818,537</td>
<td>lHCC:2-protein</td>
<td>3B protein-kinase activated nucleotide exchange factor</td>
<td></td>
</tr>
<tr>
<td>lHCC</td>
<td>984,684</td>
<td>lHCC:2-protein</td>
<td>3B protein-kinase activated nucleotide exchange factor</td>
<td></td>
</tr>
<tr>
<td>lHCC</td>
<td>195,999</td>
<td>lHCC:2-protein</td>
<td>3B protein-kinase activated nucleotide exchange factor</td>
<td></td>
</tr>
<tr>
<td>lHCC</td>
<td>4,820</td>
<td>lHCC:2-protein</td>
<td>3B protein-kinase activated nucleotide exchange factor</td>
<td></td>
</tr>
<tr>
<td>lHCC</td>
<td>1,81,432</td>
<td>lHCC:2-protein</td>
<td>3B protein-kinase activated nucleotide exchange factor</td>
<td></td>
</tr>
<tr>
<td>lHCC</td>
<td>2,304,060,040</td>
<td>lHCC:2-protein</td>
<td>3B protein-kinase activated nucleotide exchange factor</td>
<td></td>
</tr>
</tbody>
</table>

FedX – The Bigger Picture

Information Workbench:
Integration of Virtualized Data Sources as a Service
(incl. Enterprise data sources)

Application Layer

Virtualization Layer

Data Layer

Semantic Wiki
Collaboration
Reporting & Analytics
Visual Exploration

Transparent & On-Demand Integration of Data Sources

Data Registries
CKAN, data.gov, etc.
+ Enterprise Data

Information Workbench:
Integration of Virtualized Data Sources as a Service
(incl. Enterprise data sources)
Experiences & Outlook

Federation in practice

- Requires reliable federation members
  - SPARQL endpoints in controlled environments (local intranet)
  - Hard to deal with unreachable / broken endpoints
- Works best for queries with clearly separated vocabulary / namespaces
- Linking between datasets needs to be improved
- Query performance quite efficient and good for static applications (e.g. dashboarding)
  - Not yet suitable for highly interactive applications

Outlook

- Statistics layer to improve source selection and join ordering
- Support for write scenarios
- New join strategies (Hash Join instead of BNLJ)
- Component to prune subqueries by namespace
Thank you!

Contact

fluid Operations AG
Altrottstr. 31
Walldorf
Germany

+49 (0) 6227 358087-0
www.fluidops.com
contact@fluidOps.com

Further information on FedEx
http://www.fluidops.com/fedx
References

FedX: Optimization Techniques for Federated Query Processing on Linked Data
Andreas Schwarte, Peter Haase, Katja Hose, Ralf Schenkel, Michael Schmidt.
In Proc. ISWC 2011, Bonn (Germany).

FedBench: A Benchmark Suite for Federated Semantic Data Query Processing
Michael Schmidt, Olaf Görlitz, Peter Haase, Günter Ladwig, Andreas Schwarte, Thanh Tran.
In Proc. ISWC 2011, Bonn (Germany).

An Experience Report of Large Scale Federations
Andreas Schwarte, Peter Haase, Michael Schmidt, Katja Hose, Ralf Schenkel
http://arxiv.org/abs/1210.5403

FedSearch: efficiently combining structured queries and full-text search in a SPARQL federation
Andriy Nikolov, Andreas Schwarte, Christian Hütter
ISWC 2013, Sidney (Australia).
The fluidOps Platform

Sample Solutions
- IT Transparency & Data Center Intelligence for Large Hosters
- Public Portal for Complex Demo Application Landscapes
- Image Library Management for Complex Demo Application Landscapes
- Scalable End-User Access to Big Data
- Linked Data Approach to Drug Research
- Dynamic Semantic Publishing @
- Protein Engineering Portal

SDK
- API
- Configuration
- Data
- Data Providers
- Ontology
- Rules
- Templates
- Widgets
- Workflows

Partner Products & Solutions
- Optique
- Statoll
- Siemens
- Life Sciences
- Media
- NewProt

Product
- eCloudManager
- Information Workbench
  For a World Where All Data is Linked.

Platform
- fluidOps Platform
  Cloud Management
  Flexible & Data-driven UI
  Semantic Data Management

For a World Where All Data is Linked.