Virtuoso as Fedora Resource Index

(August 2013)

Introduction

On https://wiki.duraspace.org/display/FEDORA35/Resource+Index is described what it's Fedora resource index (hereafter RI). This article describes research and testing of other possible implementations of RI than existing standard implementations Mulgara or MPTStore with Postgres, which are currently the only two supported and maintained by Fedora developers. This research is part of project Kramerius (http://code.google.com/p/kramerius/). This research begun approximately in June 2012.

On http://code.google.com/p/kramerius/wiki/fedora are in Czech language described tests, which were done in the project Kramerius – these are some conclusions translated to English:

- The default implementation of resource index Mulgara is for saving more than 100,000 objects unusable.
- Alternative implementation of resource index MPTstore is functional, but supports only a limited subset of the query language iTQL. For example, it does not support composed queries that are used on one of the tabs of administrative interface for indexing documents in system Kramerius.

From these tests and also from other information elsewhere on the Internet for other projects, some of which will be mentioned below, there are grounds for investigation and testing other posibilities of implementation RI. Main goal of this research is to use in the project Kramerius better implementation of RI, if it is possible.

Trippi

First, it was necessary to find out and try whether it is possible to redirect Fedora to a different implementation of the RI than the two above mentioned implementations with Mulgara or MPTStore with Postgres. I found out and tested, that it is possible. The basis for the connection between Fedora and RI is implementation of Trippi (https://github.com/fcrepo and also old documentation on http://trippi.sourceforge.net/). Trippi is a Java library, that is used in Fedora as access point for updating and querying repository used for the implementation of RI. Trippi provides abstract classes and interfaces, which could be implemented for specific repository, that can be used for implementation of RI. Basic abstract class is there TriplestoreConnector, whose implementation (full name of an implementing class) could be given in the configuration of Fedora. I did not found other maintained implementation of Trippi than two standard implementations for the above mentioned Mulgara and MPTStore with Postgres. Only on https://github.com/fcrepo there are also some other old not maintained implementations of Trippi.

I only found on http://journals.tdl.org/jodi/article/view/5396/5885, that in the Library of Alexandria in Egypt they successfully use 4store (http://4store.org/) as RI. I tried to contact them, but they did not answered.

Virtuoso Open-Source Edition

Here are some reasons, why we decided to try to use „Virtuoso Open-Source Edition“ (hereafter VOS) as RI:

- There is also commercial variant of Virtuoso, but open-source VOS seems to be good enough.
- In chapter „2 Related Work“ in the document about the system in the Library of Alexandria (see link above) is written: " The SPAR project (SPAR Project, 2011) at the BNF considers storing the Metadata using RDF in the Virtuoso triple store (Virtuoso Universal Server software, 2011) the least risky approach (Fauduet and Peyrard, 2010) while still accepting a SIP with a METS manifest."
- On http://www.biomedcentral.com/1471-2105/13/S1/S3 are compared five non-commercial repositories (Virtuoso Open-Source Edition, Jena SDB, Jena TDB, SwiftOWLIM and 4store) and there is written: "We identified three groups of queries displaying similar behaviour across the different stores: 1) relatively short response time queries, 2) moderate response time queries and 3) relatively long response time queries. SwiftOWLIM proved to be a winner in the first group, 4Store in the second one and Virtuoso in the third one.“ … „Our analysis showed that some queries behaved idiosyncratically, in a triple store specific manner, mainly with SwiftOWLIM and 4Store. Virtuoso, as expected, displayed a very balanced performance - its load time and its response time for all the tested queries were better than average among the selected stores; it showed a very good scalability and a reasonable run-to-run reproducibility. Jena SDB and Jena TDB were consistently slower than the other three implementations. Our analysis demonstrated that most queries developed for Virtuoso could be successfully used for other implementations."
- VOS can be downloaded in source and binary form for Linux and various Unix's. A Windows binary distribution is also available. For example 4store is not available for Windows.
- Also on http://www.w3.org/wiki/LargeTripleStores we read, that VOS is good.

Source Code of Implementation of Trippi for VOS

My implementation of Trippi for VOS consists of these files:

VirtuosoConnector.java, VirtuosoSessionFactory.java, VirtuosoSession.java, VirtuosoTripleIterator.java and TriplestoreConnector.xml

TriplestoreConnector.xml is my version (for VOS) of standard Fedora configuration file C:\fedora\server\config\spring\TriplestoreConnector.xml. How to use it is described in C:\fedora\server\config\spring\README.txt.

VOS has two Java APIs for RDF - „Virtuoso Sesame Provider“ and „Virtuoso Jena Provider“. In implementation of Trippi for VOS I used „Virtuoso Sesame Provider“.

Implementation of Trippi for VOS I created this way:

From Fedora Commons Repository on https://github.com/fcrepo I took several pieces of code from existing implementations of Trippi and also from older implementations of Trippi and than I changed and programmed things, which were necessary to change or to programm. Useful was for example also old (stopped - several years not maintained) implementation of Trippi for Sesame.

In source code I also left commented out my logging statements, which I was using this way:

//logger.error("THIS IS NOT error! constructor VirtuosoSession - begin");

I changed or programmed only things, which were necessary for testing, if it is possible to use VOS as RI. First three things, which may be improved are – see TODO in next pieces of code, which I copied from my implementation of Trippi – they are edited here to fit on page:

```
TriplestoreSessionPool sessionPool =
    //TODO Create something better than SingleSessionPool
    new SingleSessionPool(m_updateSession,
        m_updateSession.listTupleLanguages(),
        m_updateSession.listTripleLanguages());
```

Next thing, which may be improved, is – see TODO:

```
public TripleIterator findTriples(String lang, String queryText)
        throws TrippiException {
    //logger.error(
       //"THIS IS NOT error! findTriples(String lang, String queryText) - begin");
    return new VirtuosoTripleIterator(repository, lang, queryText);
}

public TripleIterator findTriples(SubjectNode subject,
        PredicateNode predicate, ObjectNode object) throws TrippiException {
    //logger.error("THIS IS NOT error! findTriples(SubjectNode subject, - begin");
    return findTriples(SPARQL,
        null //TODO: For inserting triples, deleting triples and
             //SPARQL CONSTRUCT queries I didn't need this function
             //findTriples(SubjectNode subject,    PredicateNode predicate,
             //ObjectNode object)
        );
}
```

Third thing, which may be improved, is:

```
// Implements TriplestoreSession.query(String, String)
public TupleIterator query(String query,
                           String lang) throws TrippiException {
    // TODO: I didn't need this function, so I implemented it the same way as it
      // is implemented in Trippi for MPTstore.
    throw new TrippiException("Unsupported tuple query language: " + lang);
}
```

This implementation of Trippi for VOS I successfully tested on notebook (4 GB RAM, Intel Core i5-2450M CPU 2,5 GHz, Windows 7 64bit) with VOS 6.1.6 (it was latest version - today 2013-08-23 VOS 7.0.0 released 2013-08-05 is latest) and Fedora Commons 3.5 and Java 1.6.

I successfully tried following with it:

- Storing objects into RI VOS via user interface of Fedora C:\fedora\client\bin\fedora-ingest.bat

- Deleting objects from RI VOS via user interface of Fedora C:\fedora\client\bin\fedora-admin.bat
- SPARQL CONSTRUCT queries on data from RI VOS via user interface of Fedora http://localhost:8080/fedora/risearch

But some problems also occured:

Problem with „Virtuoso JDBC 4 Driver“

When I was implementing Trippi for VOS, there were two versions of „Virtuoso Sesame Provider“ - „Virtuoso Sesame 2 Provider“ (virt_sesame2.jar) and „Virtuoso Sesame 3 Provider“ (virt_sesame3.jar). When I wanted to use „Virtuoso JDBC 4 Driver“ (virtjdbc4.jar), that is type 4 JDBC driver, I must use „Virtuoso Sesame 3 Provider“ with it. But there were an incompatibility issue between „Virtuoso Sesame 3 Provider“ and Sesame 3 library „openrdf-sesame-3.0-alpha1-onejar.jar“ (it was latest, but to old version of Sesame 3 library – there were newer Sesame 2 libraries). So it was impossible to use „Virtuoso JDBC 4 Driver“ – so I used type 3 JDBC driver „Virtuoso JDBC 3 Driver“ (virtjdbc3.jar) and „Virtuoso Sesame 2 Provider“ (virt_sesame2.jar) and „openrdf-sesame-2.6.5-onejar.jar“. I also think, that because of bad state of Sesame 3, it wouldn't be a good idea to use „Virtuoso Sesame 3 Provider“.

Today (2013-08-23) on http://virtuoso.openlinksw.com/dataspace/doc/dav/wiki/Main/VOSDownload there is no „Virtuoso Sesame 3 Provider“ – there are two versions of „Virtuoso Sesame Provider“ - „Virtuoso Sesame 2.6.x Provider“ and „Virtuoso Sesame 2.7.x Provider“. May be, that now it is possible to use „Virtuoso JDBC 4 Driver“ with one or both of them.

Problem with older version of VOS

When we wanted to do some tests with VOS on Linux server, where was preinstalled older version of VOS (VOS 6.1.0, which was released 2010-02-03), following exception occured (also on Windows 7), when we were inserting the same objects, which we successfully inserted with VOS 6.1.6 on Windows 7:

```
org.trippi.TrippiException: java.sql.BatchUpdateException: Batch executes only update statements
   at cz.knav.virtuoso.trippi.VirtuosoSession.throwTrippiException(VirtuosoSession.java:65)
[virtuoso.jar:na]
   at cz.knav.virtuoso.trippi.VirtuosoSession.doTriples(VirtuosoSession.java:92) [virtuoso.jar:na]
   at cz.knav.virtuoso.trippi.VirtuosoSession.add(VirtuosoSession.java:47) [virtuoso.jar:na]
   at org.trippi.impl.base.MemUpdateBuffer.writeBatch(Unknown Source) [trippi-core-1.5.4.jar:na]
   at org.trippi.impl.base.MemUpdateBuffer.flush(Unknown Source) [trippi-core-1.5.4.jar:na]
   at org.trippi.impl.base.ConcurrentTriplestoreWriter.flushBuffer(Unknown Source) [trippi-core-
1.5.4.jar:na]
   at org.trippi.impl.base.ConcurrentTriplestoreWriter.run(Unknown Source) [trippi-core-1.5.4.jar:na]
   at java.lang.Thread.run(Thread.java:662) [na:1.6.0_26]
Caused by: org.openrdf.repository.RepositoryException: java.sql.BatchUpdateException: Batch
executes only update statements
   at virtuoso.sesame2.driver.VirtuosoRepositoryConnection.addToQuadStore(Unknown Source)
[virt_sesame2.jar:na]
```

```
   at virtuoso.sesame2.driver.VirtuosoRepositoryConnection.add(Unknown Source)
[virt_sesame2.jar:na]
   at cz.knav.virtuoso.trippi.VirtuosoSession.doTriples(VirtuosoSession.java:87) [virtuoso.jar:na]
   ... 6 common frames omitted
Caused by: java.sql.BatchUpdateException: Batch executes only update statements
   at virtuoso.jdbc3.VirtuosoPreparedStatement.throwBatchUpdateException(Unknown Source)
[virtjdbc3.jar:na]
   at virtuoso.jdbc3.VirtuosoPreparedStatement.executeBatch(Unknown Source) [virtjdbc3.jar:na]
   ... 9 common frames omitted
```

In source code of „Virtuoso JDBC 3 Driver“ on http://www.contexta.cl/sites/contexta-virtuoso-jdbc3-driver/xref/virtuoso/jdbc3/VirtuosoPreparedStatement.html this exception is thrown this way:

```
 if (vresultSet.kindop() == VirtuosoTypes.QT_SELECT)
      throwBatchUpdateException(result,
          "Batch executes only update statements", inx);
result = executeBatchUpdate();
```

We did not care about this problem more and we stopped doing tests with VOS on Linux server.

Problem with one SPARQL CONSTRUCT query

I was testing VOS much more with SPARQL SELECT queries, which I was passing to VOS directly from Java program using „Virtuoso Sesame 2 Provider“ (so not through Fedora and Trippi), than with SPARQL CONSTRUCT queries passed from http://localhost:8080/fedora/risearch. Some SPARQL CONSTRUCT queries on http://localhost:8080/fedora/risearch I tried only to test my Trippi implementation for VOS and with one query (see following part of log file) following problem occured - I did not found solution of this problem:

```
ERROR 2013-02-14 15:13:04.341 [http-8080-2] (VirtuosoSession) THIS IS NOT error! findTriples(String
lang, String queryText) - begin
ERROR 2013-02-14 15:13:04.341 [http-8080-2] (VirtuosoTripleIterator) THIS IS NOT error!
constructor VirtuosoTripleIterator - begin
ERROR 2013-02-14 15:13:04.351 [http-8080-2] (VirtuosoTripleIterator) : SPARQL execute
failed:[construct { ?o ?t ?d } where {
?o <info:fedora/fedora-system:def/model#hasModel> <info:fedora/model:page> .
?o <http://purl.org/dc/elements/1.1/title> ?t .
?o <info:fedora/fedora-system:def/view#lastModifiedDate> ?d .
}]
 Exception:virtuoso.jdbc3.VirtuosoException: SR007: Function sprintf needs a string or UNAME or
NULL as argument 2, not an arg of type INTEGER (189)
org.openrdf.query.QueryEvaluationException: : SPARQL execute failed:[construct { ?o ?t ?d } where {
?o <info:fedora/fedora-system:def/model#hasModel> <info:fedora/model:page> .
?o <http://purl.org/dc/elements/1.1/title> ?t .
?o <info:fedora/fedora-system:def/view#lastModifiedDate> ?d .
}]
```

```
 Exception:virtuoso.jdbc3.VirtuosoException: SR007: Function sprintf needs a string or UNAME or
NULL as argument 2, not an arg of type INTEGER (189)
        at
virtuoso.sesame2.driver.VirtuosoRepositoryConnection.executeSPARQLForGraphResult(Unknown
Source) [virt_sesame2.jar:na]
        at virtuoso.sesame2.driver.VirtuosoRepositoryConnection$2.evaluate(Unknown Source)
[virt_sesame2.jar:na]
        at cz.knav.virtuoso.trippi.VirtuosoTripleIterator.<init>(VirtuosoTripleIterator.java:38)
[VirtuosoTripleIterator.class:na]
        at cz.knav.virtuoso.trippi.VirtuosoSession.findTriples(VirtuosoSession.java:110)
[VirtuosoSession.class:na]
        at org.trippi.impl.base.ConcurrentTriplestoreReader.findTriples(Unknown Source) [trippi-
core-1.5.4.jar:na]
        at
org.fcrepo.server.resourceIndex.ResourceIndexImpl.findTriples(ResourceIndexImpl.java:300) [fcrepo-
server-3.5.jar:na]
        at
org.fcrepo.server.resourceIndex.ResourceIndexModule.findTriples(ResourceIndexModule.java:332)
[fcrepo-server-3.5.jar:na]
        at org.trippi.server.TrippiServer.find(Unknown Source) [trippi-core-1.5.4.jar:na]
        at org.trippi.server.http.TrippiServlet.doFind(Unknown Source) [trippi-core-1.5.4.jar:na]
        at org.trippi.server.http.TrippiServlet.doGet(Unknown Source) [trippi-core-1.5.4.jar:na]
        at org.fcrepo.server.access.RISearchServlet.doGet(RISearchServlet.java:101) [fcrepo-server-
3.5.jar:na]
        at org.trippi.server.http.TrippiServlet.doGet(Unknown Source) [trippi-core-1.5.4.jar:na]
        at org.trippi.server.http.TrippiServlet.doPost(Unknown Source) [trippi-core-1.5.4.jar:na]
        at javax.servlet.http.HttpServlet.service(HttpServlet.java:637) [servlet-api.jar:na]
        at javax.servlet.http.HttpServlet.service(HttpServlet.java:717) [servlet-api.jar:na]
        at
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:290)
[catalina.jar:6.0.35]
        at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
[catalina.jar:6.0.35]
        at
org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:
368) [spring-security-web-3.0.5.RELEASE.jar:3.0.5.RELEASE]
        at
org.springframework.security.web.access.channel.ChannelProcessingFilter.doFilter(ChannelProcessin
gFilter.java:109) [spring-security-web-3.0.5.RELEASE.jar:3.0.5.RELEASE]
        at
org.springframework.security.web.FilterChainProxy$VirtualFilterChain.doFilter(FilterChainProxy.java:
380) [spring-security-web-3.0.5.RELEASE.jar:3.0.5.RELEASE]
        at org.springframework.security.web.FilterChainProxy.doFilter(FilterChainProxy.java:169)
[spring-security-web-3.0.5.RELEASE.jar:3.0.5.RELEASE]
```

```
        at
org.springframework.web.filter.DelegatingFilterProxy.invokeDelegate(DelegatingFilterProxy.java:237
) [spring-web-3.0.5.RELEASE.jar:3.0.5.RELEASE]
        at
org.springframework.web.filter.DelegatingFilterProxy.doFilter(DelegatingFilterProxy.java:167)
[spring-web-3.0.5.RELEASE.jar:3.0.5.RELEASE]
        at
org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:235)
[catalina.jar:6.0.35]
        at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:206)
[catalina.jar:6.0.35]
        at org.apache.catalina.core.StandardWrapperValve.invoke(StandardWrapperValve.java:233)
[catalina.jar:6.0.35]
        at org.apache.catalina.core.StandardContextValve.invoke(StandardContextValve.java:191)
[catalina.jar:6.0.35]
        at org.apache.catalina.core.StandardHostValve.invoke(StandardHostValve.java:127)
[catalina.jar:6.0.35]
        at org.apache.catalina.valves.ErrorReportValve.invoke(ErrorReportValve.java:102)
[catalina.jar:6.0.35]
        at org.apache.catalina.authenticator.SingleSignOn.invoke(SingleSignOn.java:394)
[catalina.jar:6.0.35]
        at org.apache.catalina.core.StandardEngineValve.invoke(StandardEngineValve.java:109)
[catalina.jar:6.0.35]
        at org.apache.catalina.connector.CoyoteAdapter.service(CoyoteAdapter.java:293)
[catalina.jar:6.0.35]
        at org.apache.coyote.http11.Http11AprProcessor.process(Http11AprProcessor.java:877)
[tomcat-coyote.jar:6.0.35]
        at
org.apache.coyote.http11.Http11AprProtocol$Http11ConnectionHandler.process(Http11AprProtocol
.java:594) [tomcat-coyote.jar:6.0.35]
        at org.apache.tomcat.util.net.AprEndpoint$Worker.run(AprEndpoint.java:1675) [tomcat-
coyote.jar:6.0.35]
        at java.lang.Thread.run(Unknown Source) [na:1.6.0_31]
ERROR 2013-02-14 15:13:04.353 [http-8080-2] (RISearchServlet) Unexpected error servicing API-A
request
org.trippi.TrippiException: : SPARQL execute failed:[construct { ?o ?t ?d } where {
?o <info:fedora/fedora-system:def/model#hasModel> <info:fedora/model:page> .
?o <http://purl.org/dc/elements/1.1/title> ?t .
?o <info:fedora/fedora-system:def/view#lastModifiedDate> ?d .
}]
 Exception:virtuoso.jdbc3.VirtuosoException: SR007: Function sprintf needs a string or UNAME or
NULL as argument 2, not an arg of type INTEGER (189)
        at
cz.knav.virtuoso.trippi.VirtuosoTripleIterator.throwTrippiException(VirtuosoTripleIterator.java:141)
[VirtuosoTripleIterator.class:na]
```

```
        at cz.knav.virtuoso.trippi.VirtuosoTripleIterator.<init>(VirtuosoTripleIterator.java:40)
[VirtuosoTripleIterator.class:na]
        at cz.knav.virtuoso.trippi.VirtuosoSession.findTriples(VirtuosoSession.java:110)
[VirtuosoSession.class:na]
...
```

Conclusion

This first version of Trippi for VOS is functional - it seems, that VOS could be a good RI.

Besides implementing Trippi for VOS, I also tested VOS filled with 108 469 446 triples. I filled VOS directly from Java program using „Virtuoso Sesame 2 Provider“ (so not through Fedora and Trippi) and than I was quering VOS with SPARQL SELECT queries passed from Java program using „Virtuoso Sesame 2 Provider“ (so not through Fedora and Trippi). During filling VOS I did not encountered any problem and during querying VOS only problem, which I encountered, is described with solution on this link: http://www.openlinksw.com/dataspace/doc/dav/wiki/Main/VirtTipsAndTricksHowToHandleBandwidthLimitExceed