Performance Testing

VIVO's Triple Store Options

As VIVO continues its support of alternate triple stores, it is important to have a clear picture of the performance of various VIVO/triple-store configurations. This page will document the procedures and results of testing against these configurations.

System details

- VIVO 1.11.0
    - with logging patch
    - with developer properties "loggingRDFService" enabled (see details below)
    - with inferencing disabled
    - with search indexing disabled
- Java -version: 11.0.6
- JAVA_OPTS="${JAVA_OPTS} -Xms2G -Xmx8G -XX:MaxNewSize=2G"

Ingest testing

This test is designed to measure the amount of time taken to ingest a standard data set. The data set used in this test is the published OpenVIVO content found in the vivo-project/sample-data GitHub repository.

Test procedure

The following procedure was taken for each of the test runs:

1. Stop VIVO
2. Clear triple store prior to test
3. Clear Tomcat logs
4. Start VIVO
5. Log in as vivo_root
6. Verify no content in VIVO
7. Site Admin -> Add or Remove RDF Data
    - From local download: openvivo.ttl

After upload has completed, analyze the VIVO log(s)

1. Total time for ingest determined by "grepping" for "ingest" in the vivo.all.log(s)
    a. There should be two lines, like the following:

```
2020-02-26 22:45:18,938 INFO  [RDFUploadController] Start ingest: 2020-02-27T03:45:18.937813Z
2020-02-27 00:08:27,242 INFO  [RDFUploadController] Stop ingest: 2020-02-27T05:08:27.242238Z,
total time: PT1H23M8.304425S
```

2. Time for each method invoked on the RDFService implementation
    a. The attached script is run over a concatenation of all vivo.all.log files created during the ingest process
    b. The script produces a report of total times for each RDFService method, like the following:

```
   calls      sec   sec/call              method
==================================================
   8502    483.01    0.0568        changeSetUpdate
1389889   1895.31    0.0014   sparqlConstructQuery
   7056     16.72    0.0024      sparqlSelectQuery
   4261      3.52    0.0008         sparqlAskQuery
     14      0.04    0.0029      isEquivalentGraph
Total time: 2398.603 sec (~39 mins, or ~0 hrs)
```

Enabling developer properties

1. Update file `$VIVO_HOME/config/developer.properties`, ensuring the following options are enabled/uncommented

```
developer.enabled = true
developer.loggingRDFService.enable = true
developer.loggingRDFService.queryRestriction = .*
developer.loggingRDFService.stackRestriction = .*
```

Test results

TDB

Run 1

1. Total time: *12min 42sec*

```
2020-02-26 21:53:03,478 INFO  [RDFUploadController] Start ingest: 2020-02-27T02:53:03.478638Z
2020-02-26 22:05:46,016 INFO  [RDFUploadController] Stop ingest: 2020-02-27T03:05:46.015668Z, total
time: PT12M42.53703S
```

2. Method invocation times

```
   calls      sec    sec/call               method
=================================================
    8502     48.97     0.0058        changeSetUpdate
 1406755    380.57     0.0003 sparqlConstructQuery
   10101     11.72     0.0012     sparqlSelectQuery
   12354      2.15     0.0002        sparqlAskQuery
      14      0.83     0.0592      isEquivalentGraph
Total time: 444.245 sec (~7 mins, or ~0 hrs)
```

SDB

Run 1

1. Total time: *1hr 23min 8sec*

```
2020-02-26 22:45:18,938 INFO  [RDFUploadController] Start ingest: 2020-02-27T03:45:18.937813Z
2020-02-27 00:08:27,242 INFO  [RDFUploadController] Stop ingest: 2020-02-27T05:08:27.242238Z, total
time: PT1H23M8.304425S
```

2. Method invocation times

```
   calls      sec    sec/call               method
=================================================
    8502    483.01     0.0568        changeSetUpdate
 1389889   1895.31     0.0014 sparqlConstructQuery
    7056     16.72     0.0024     sparqlSelectQuery
    4261      3.52     0.0008        sparqlAskQuery
      14      0.04     0.0029      isEquivalentGraph
Total time: 2398.603 sec (~39 mins, or ~0 hrs)
```

Fuseki (local, backed by TDB)

Run 1

1. Total time: *1hr 11min 0sec*

```
2020-02-27 20:58:05,486 INFO  [RDFUploadController] Start ingest: 2020-02-28T01:58:05.486176Z
2020-02-27 22:09:05,833 INFO  [RDFUploadController] Stop ingest: 2020-02-28T03:09:05.829769Z, total
time: PT1H11M0.343593S
```

2. Method invocation times

```
   calls       sec    sec/call                   method
==================================================
    1302    176.50      0.1356        changeSetUpdate
 1387044   2697.63      0.0019 sparqlConstructQuery
    6791    107.68      0.0159      sparqlSelectQuery
    2868     13.65      0.0048         sparqlAskQuery
      14      0.86      0.0615       isEquivalentGraph
Total time: 2996.323 sec (~49 mins, or ~0 hrs)
```

Read testing

This test is designed to measure the amount of time taken to read a fixed data set. The data used in this test is the published OpenVIVO content found in the vivo-project/sample-data GitHub repository, previously ingested into VIVO... and for this test, read by the VIVO Scholars application in the process of Scholars populating its dedicated Solr index.

Test procedure

The OpenVIVO test data is initially ingested into VIVO as described in the previous "Ingest testing" procedure. After ingest to VIVO, the VIVO Scholars application is started with a connection to the VIVO data store. During VIVO Scholar's start-up procedure, it reads content from VIVO's data store in order to populate its dedicated Solr index.

These "read tests" capture the timing of the time it takes VIVO Scholar to update its Solr index.