

Performance Measurements for Fedora GSearch version 2.3

By Gert Schmeltz Pedersen and Morten Sørensen, DTU Library

Introduction

We have performed a set of measurements of GSearch 2.3, running on a platform suitable for production use. We have analyzed the variables and found seven relevant and interesting test dimensions. We have used Apache JMeter as test framework, creating and running a test plan.

We first describe the test environment. Then we explain the test dimensions and the test plan. We have extracted measurement data from JMeter and present them in a set of measurement tables. Finally, we conclude on the findings.

Many more conclusions and insights might be gathered from the data and more refined test plans be created and run. We encourage others to follow up and report on results. The test plan and the measurement spreadsheet may be downloaded, see the References.

Test Environment

Hardware:

- A virtual server (VMWare ESXi) with one 2.93GHz Intel Xeon CPU and 3GB ram
- The physical VMWare Host is a 2x6 core IBM HS22 Bladeserver with 96GB ram
- The Disk system under the virtual system is an IBM Storwize v7000 SAN

Software:

- Debian 6 Squeeze with openjdk-6-jdk (6b18-1.8.9-0.1~squeeze1)
- Tomcat6 (6.0.28-9+squeeze1)
- Fedora Commons 3.5
- Fedora GSearch 2.3
- Tomcat has 2 of the 3GB ram (-Xmx2048m)
- Apache JMeter

Test Plan

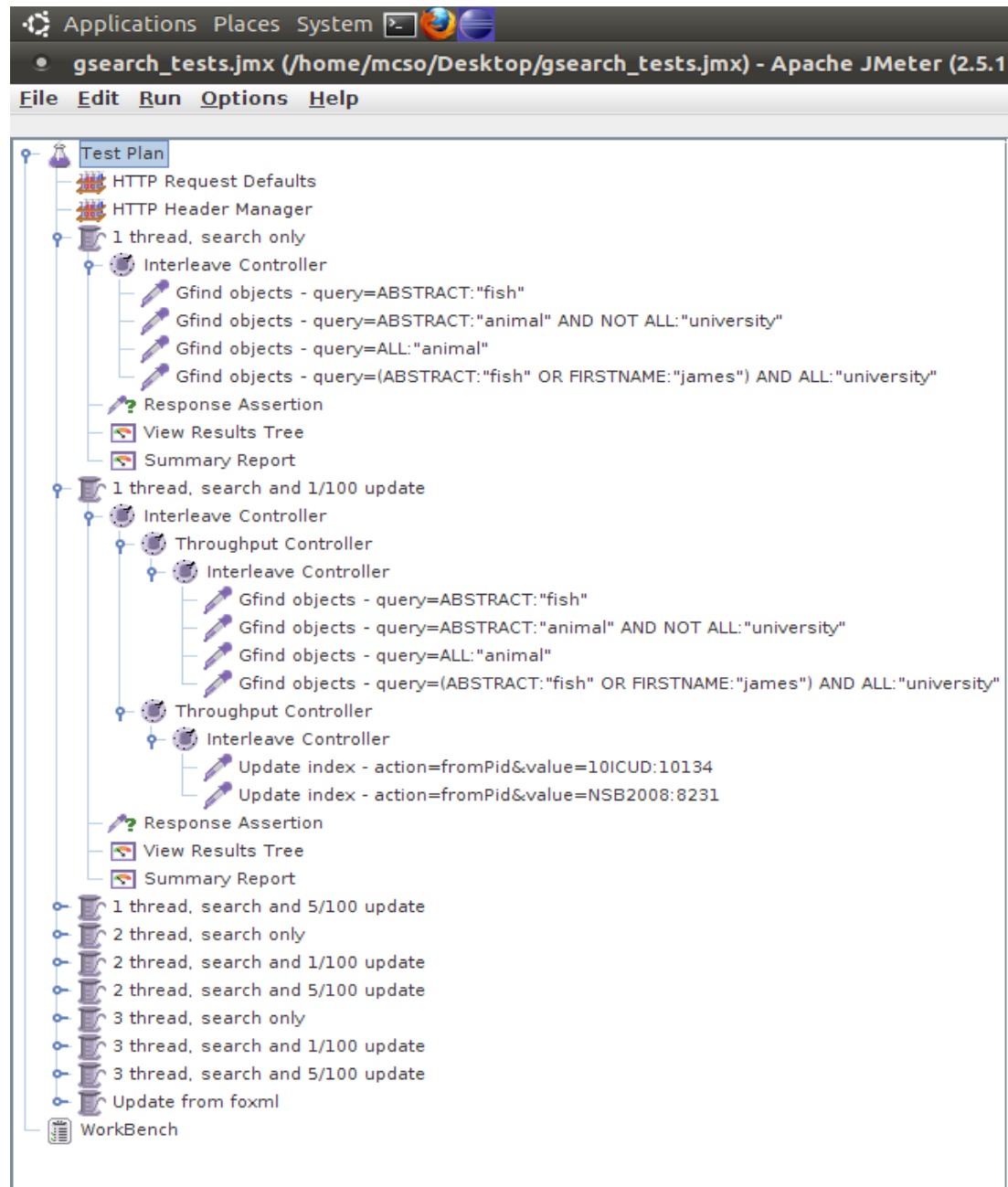
The **Apache JMeter™** desktop application is open source software, a Java application designed to load test functional behavior and measure performance. It can be used to simulate a heavy load on a server, network or object to test its strength or to analyze overall performance under different load types.

A test plan describes a series of steps JMeter will execute when run. A complete test plan will consist of one or more Thread Groups, logic controllers, sample generating controllers, listeners, timers, assertions, and configuration elements.

We have built the following test plan consisting of queries and updates, running in one, two, or three threads. Three threads simulate heavy load. One set of test cases consists of queries only, two other sets have queries interleaved with 1% or 5% updates.

The test plan has been saved in xml format and is available for download, see References.

The test results have been saved as spreadsheets and are available for download, see References.



Test Dimensions – Explanation of measurement table columns

Measurements have been extracted from the spreadsheets from JMeter and presented in tables in the next section. The table columns are explained in the following.

Operation – defined in JMeter Test Plan

- gfindObjects – query I = ABSTRACT:"fish"
- gfindObjects – query II = ABSTRACT:"animal" AND NOT ALL:"university"
- gfindObjects – query III = (ABSTRACT:"fish" OR FIRSTNAME:"james") AND ALL:"university"
- updateIndex – one object without pdf = action=fromPid&value=NSB2008:8231
- updateIndex – one object with pdf = action=fromPid&value=10ICUD:10134

Thread count – defined in JMeter Test Plan

- 1 – one thread at a time, operations cannot overlap
- 2 – two threads at a time, some delay due to overlap may occur
- 3 – three threads at a time, simulates heavy traffic

Update percent – defined in JMeter Test Plan

- 0% - no updates mixed with queries
- 1% - updates occur once for every 99 queries
- 5% - updates occur 5 times for every 95 queries

Xslt processor – configured in fedoragsearch.properties

- Xalan – the original option, has Xpath 1.0 functions
- Saxon – the new choice, has Xpath 2.0 functions

Interface

- gsearch – operation sent to GSearch server
- solr – operation sent to Solr server

Indexer

- lucene – configured in index.properties to use lucene plugin
- solr – configured in index.properties to use solr plugin

Object count – two sizes of Fedora repositories, around 75% of the objects have a pdf datastream of 2-8 pages

- 12020 – Ten copies of the original 1202 objects
- 120200 – Hundred copies of the original 1202 objects

ms – average time in milliseconds, over appr. 250 repetitions of queries, and over 5 to 25 repetitions of updates.

Test results

Table 1 Simplest query - 1 thread –impact from Interface: solr best

<i>Operation</i>	<i>Thread count</i>	<i>Update percent</i>	<i>Xslt processor</i>	<i>Interface</i>	<i>Indexer</i>	<i>Object count</i>	<i>ms</i>
gfindObjects - query l	1	0%	xalan	solr	solr	120200	30
gfindObjects - query l	1	1%	xalan	solr	solr	12020	30
gfindObjects - query l	1	1%	xalan	solr	solr	120200	30
gfindObjects - query l	1	5%	xalan	solr	solr	12020	30
gfindObjects - query l	1	5%	xalan	solr	solr	120200	30
gfindObjects - query l	1	0%	xalan	solr	solr	12020	31
gfindObjects - query l	1	5%	saxon	gsearch	lucene	12020	32
gfindObjects - query l	1	1%	saxon	gsearch	lucene	12020	33
gfindObjects - query l	1	5%	xalan	gsearch	lucene	12020	38
gfindObjects - query l	1	1%	xalan	gsearch	lucene	12020	39
gfindObjects - query l	1	5%	xalan	gsearch	solr	120200	49
gfindObjects - query l	1	1%	xalan	gsearch	solr	120200	50
gfindObjects - query l	1	0%	saxon	gsearch	lucene	12020	51
gfindObjects - query l	1	0%	xalan	gsearch	lucene	12020	56
gfindObjects - query l	1	1%	xalan	gsearch	solr	12020	58
gfindObjects - query l	1	0%	xalan	gsearch	solr	120200	70
gfindObjects - query l	1	0%	xalan	gsearch	solr	12020	76
gfindObjects - query l	1	5%	xalan	gsearch	solr	12020	114

Table 2 Impact of Tread count, impact of Interface, no impact of Object count

<i>Operation</i>	<i>Thread count</i>	<i>Update percent</i>	<i>Xslt processor</i>	<i>Interface</i>	<i>Indexer</i>	<i>Object count</i>	<i>ms</i>
gfindObjects - query I	1	0%	xalan	gsearch	solr	12020	76
gfindObjects - query I	1	0%	xalan	gsearch	lucene	12020	56
gfindObjects - query I	2	0%	xalan	gsearch	solr	12020	113
gfindObjects - query I	2	0%	xalan	gsearch	lucene	12020	79
gfindObjects - query I	3	0%	xalan	gsearch	solr	12020	167
gfindObjects - query I	3	0%	xalan	gsearch	lucene	12020	159
gfindObjects - query I	1	0%	xalan	gsearch	solr	120200	70
gfindObjects - query I	2	0%	xalan	gsearch	solr	120200	101
gfindObjects - query I	3	0%	xalan	gsearch	solr	120200	150
gfindObjects - query I	1	0%	xalan	solr	solr	12020	31
gfindObjects - query I	2	0%	xalan	solr	solr	12020	49
gfindObjects - query I	3	0%	xalan	solr	solr	12020	70
gfindObjects - query I	1	0%	xalan	solr	solr	120200	30
gfindObjects - query I	2	0%	xalan	solr	solr	120200	53
gfindObjects - query I	3	0%	xalan	solr	solr	120200	70

Table 3 Strange reverse impact of Operation, impact of Interface, no impact of Object count

<i>Operation</i>	<i>Thread count</i>	<i>Update percent</i>	<i>Xslt processor</i>	<i>Interface</i>	<i>Indexer</i>	<i>Object count</i>	<i>ms</i>
gfindObjects - query I	3	5%	xalan	gsearch	solr	12020	169
gfindObjects - query I	3	5%	xalan	gsearch	lucene	12020	158
gfindObjects - query II	3	5%	xalan	gsearch	lucene	12020	143
gfindObjects - query II	3	5%	xalan	gsearch	solr	12020	35
gfindObjects - query III	3	5%	xalan	gsearch	lucene	12020	138
gfindObjects - query III	3	5%	xalan	gsearch	solr	12020	38
gfindObjects - query I	3	5%	xalan	gsearch	solr	120200	154
gfindObjects - query II	3	5%	xalan	gsearch	solr	120200	32
gfindObjects - query III	3	5%	xalan	gsearch	solr	120200	34
gfindObjects - query I	3	5%	xalan	solr	solr	12020	73
gfindObjects - query II	3	5%	xalan	solr	solr	12020	20
gfindObjects - query III	3	5%	xalan	solr	solr	12020	15
gfindObjects - query I	3	5%	xalan	solr	solr	120200	70
gfindObjects - query II	3	5%	xalan	solr	solr	120200	21
gfindObjects - query III	3	5%	xalan	solr	solr	120200	16

Table 4 Impact of Xslt processor: saxon best at queries, xalan best at updates, except for Thread count 2 and 3

<i>Operation</i>	<i>Thread count</i>	<i>Update percent</i>	<i>Xslt processor</i>	<i>Interface</i>	<i>Indexer</i>	<i>Object count</i>	<i>ms</i>
gfindObjects - query I	3	5%	xalan	gsearch	lucene	12020	158
gfindObjects - query I	3	5%	saxon	gsearch	lucene	12020	115
gfindObjects - query II	3	5%	xalan	gsearch	lucene	12020	143
gfindObjects - query II	3	5%	saxon	gsearch	lucene	12020	106
gfindObjects - query III	3	5%	xalan	gsearch	lucene	12020	138
gfindObjects - query III	3	5%	saxon	gsearch	lucene	12020	107
updateIndex - no pdf	1	1%	xalan	gsearch	lucene	12020	103
updateIndex - no pdf	1	1%	saxon	gsearch	lucene	12020	154
updateIndex - no pdf	1	5%	xalan	gsearch	lucene	12020	57
updateIndex - no pdf	1	5%	saxon	gsearch	lucene	12020	68
updateIndex - no pdf	2	1%	xalan	gsearch	lucene	12020	50
updateIndex - no pdf	2	1%	saxon	gsearch	lucene	12020	58
updateIndex - no pdf	2	5%	xalan	gsearch	lucene	12020	56
updateIndex - no pdf	2	5%	saxon	gsearch	lucene	12020	142
updateIndex - no pdf	3	1%	xalan	gsearch	lucene	12020	332
updateIndex - no pdf	3	1%	saxon	gsearch	lucene	12020	331
updateIndex - no pdf	3	5%	xalan	gsearch	lucene	12020	306
updateIndex - no pdf	3	5%	saxon	gsearch	lucene	12020	225
updateIndex - with pdf	1	1%	xalan	gsearch	lucene	12020	487
updateIndex - with pdf	1	1%	saxon	gsearch	lucene	12020	1256
updateIndex - with pdf	1	5%	xalan	gsearch	lucene	12020	230
updateIndex - with pdf	1	5%	saxon	gsearch	lucene	12020	277
updateIndex - with pdf	2	1%	xalan	gsearch	lucene	12020	610
updateIndex - with pdf	2	1%	saxon	gsearch	lucene	12020	664
updateIndex - with pdf	2	5%	xalan	gsearch	lucene	12020	546
updateIndex - with pdf	2	5%	saxon	gsearch	lucene	12020	483
updateIndex - with pdf	3	1%	xalan	gsearch	lucene	12020	776
updateIndex - with pdf	3	1%	saxon	gsearch	lucene	12020	682
updateIndex - with pdf	3	5%	xalan	gsearch	lucene	12020	689
updateIndex - with pdf	3	5%	saxon	gsearch	lucene	12020	856

Table 5 Impact of Object count and Indexer cannot be judged because of insufficient measurements

<i>Operation</i>	<i>Thread count</i>	<i>Update percent</i>	<i>Xslt processor</i>	<i>Interface</i>	<i>Indexer</i>	<i>Object count</i>	<i>min</i>
updateIndex fromFoxmlFiles			xalan	gsearch	lucene	12020	~30
updateIndex fromFoxmlFiles			xalan	gsearch	solr	120200	~340

Conclusions

Impacts and non-impacts have shown up as follows for the seven test dimensions:

Operation

Measured times go up to 168 ms, when there are three simultaneous threads, Table 3, and down to 15 ms, when queries are sent directly to the Solr server

There is a strange reverse impact from gfindObjects queries, where the simple query takes longer than more complex ones. This needs more measurements.

Table 4 shows that updates of one object without pdf take between 50 ms and 332 ms. For one object with pdf, updates take between 230 and 1256 ms.

Table 5 shows two measurements for updateIndex fromFoxmlFiles. One on 12020 objects uses around 150 ms per object, the other on 120200 objects uses around 170 ms per object. Around 75% of the objects have a pdf datastream of 4-8 pages.

Thread count

Table 2 shows a considerable impact from the number of simultaneous threads used by JMeter to send query operations, from 56 ms at one thread to 167 ms at three threads on the Gsearch server, and from 30 ms to 70 ms on the Solr server. Three threads simulate a heavy traffic on a web server, and few more threads can put it out of action.

Update percent

Variations of the update percent are used to simulate various real life cases. Our measurements have not shown particular impacts from the variations employed.

Xslt processor

Table 4 indicates that saxon is 30-40% better on queries, while xalan is 20-50% better on updates under light traffic. A few measurements are more extreme, so more decisive measurements should be carried out.

Interface

Queries sent to the Solr server take less than half the time than the same queries sent to the GSearch server, Table 1 and Table 2.

Indexer

The choice of indexer plugin done in index.properties may have an impact on updateIndex operations, but our measurements have not been sufficient to reveal it. Besides, the choice affects the further configuration choices that the administrator may make. For the lucene plugin they are in index.properties, for the solr plugin they are in the conf directory, especially in schema.xml and solrconfig.xml.

Object count

The two sizes of Fedora repositories for the measurements have not shown higher times for queries on the larger repository, and only linear increase for updates. More measurements are needed on this dimension.

References

Apache JMeter (<http://jmeter.apache.org/>)

Test plan as generated by JMeter:

<https://github.com/fcrepo/gsearch/tree/master/FedoraGenericSearch/src/performance/>

Measurements in Excel spreadsheet:

<https://github.com/fcrepo/gsearch/tree/master/FedoraGenericSearch/src/performance/>

Measurements in Open Office spreadsheet:

<https://github.com/fcrepo/gsearch/tree/master/FedoraGenericSearch/src/performance/>