Solr: Peak Performance

Mark.Miller @
LucidImagination.com

The Agenda

- Quick Solr Background
- Performance Factors
- Performance Monitoring
- Benchmarking
- Quotes and Pictures!

- A wise man can see more from the bottom of a well than a fool can from a mountain top . (Unknown)

- **Wise:** having the power of discerning and judging properly as to what is true or right; possessing discernment, judgment, or discretion.

- **Wise:** possessed of or characterized by scholarly knowledge or learning; learned;

- Who Knows Solr? Lucene?
- Who Knows of the Committers?
- Read the User lists
- Read Lucene in Action
- Read the Solr Wiki
- Read the free Lucid Solr Ref Guide
- Read the Solr books

"The capacity of human beings to bore one another seems to be vastly greater than that of any other animal."

H. L. Mencken
US editor (1880 - 1956)

- Que non boring inspirational image:

- Inspiration

- a. Stimulation of the mind or emotions to a high level of feeling or activity.

- b. The condition of being so stimulated.

Yes - believe it or not, you where all just inspired.

"When monitoring or debugging performance, things are often not as they first appear."

- Me?? Now?

Performance Factors

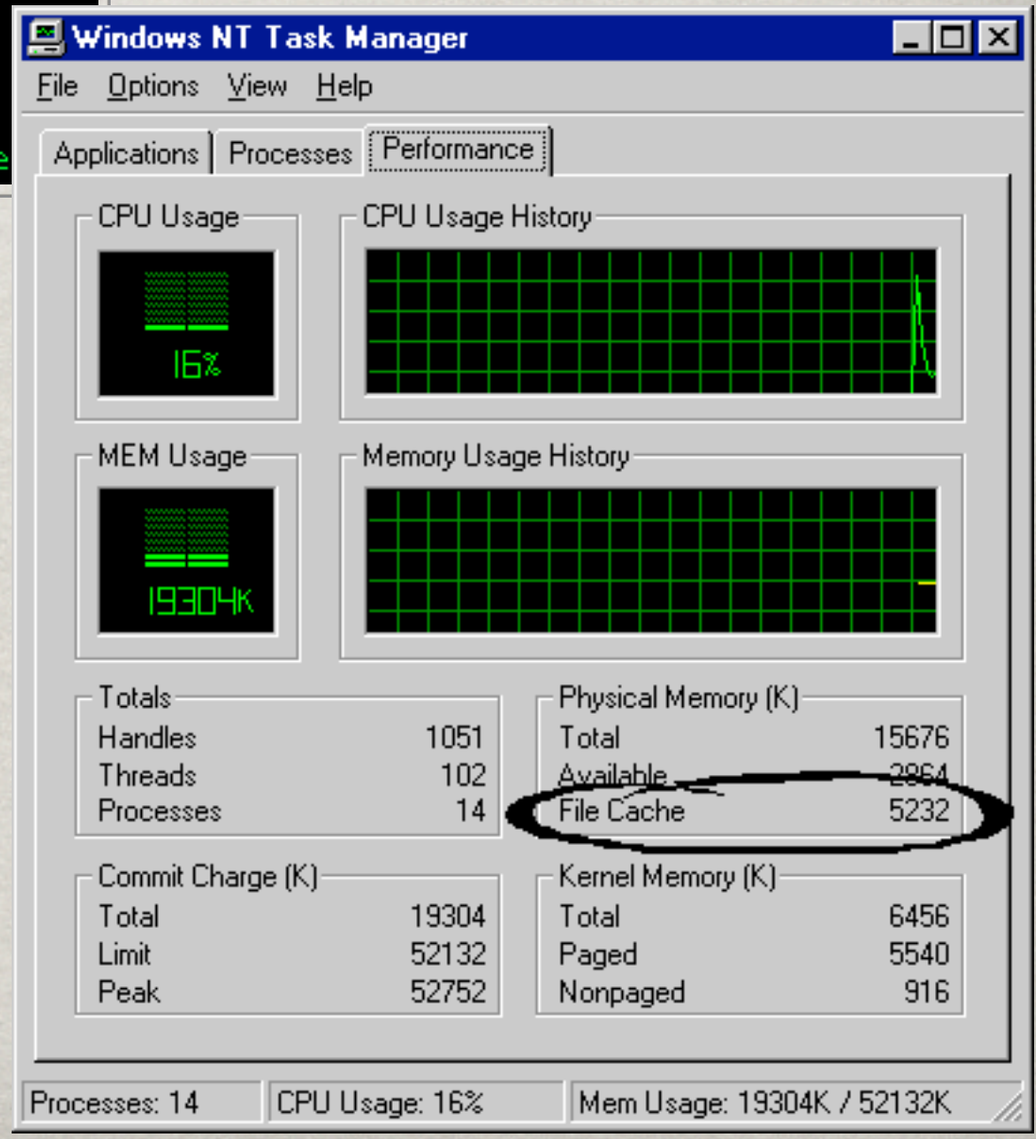
- “Give a man a fish and you feed him for a day; teach him how to fish… and you’ve opened up a whole ‘nother can of worms.” -Warm Southern Breeze

We are here to crack open a can of something.

?

```
top - 09:24:23 up 3 min,  1 user,  load average: 1.26, 0.88, 0.37
Tasks: 168 total,   1 running, 167 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.1%us,  0.2%sy,  0.0%ni, 99.8%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   8185600k total,   349500k used,  7836100k free,   115772k buffers
Swap:  8193144k total,        0k used,  8193144k free,   115680k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 1261 root      20   0 57940  17m 7120 S    0  0.2   0:00.62 Xorg
 1531 gdm       20   0 85884  13m  10m S    0  0.2   0:00.13 gdm-simple-gree
 1493 gdm       20   0 77332  10m 7972 S    0  0.1   0:00.16 gnome-settings-
 1528 gdm       20   0 73200 9404 7412 S    0  0.1   0:00.05 metacity
 1530 gdm       20   0 31568 7540 6172 S    0  0.1   0:00.04 gnome-power-man
 1468 gdm       20   0 39368 6268 5176 S    0  0.1   0:00.04 gnome-session
 1557 gdm        9 -11 99500 4700 3576 S    0  0.1   0:00.09 pulseaudio
 1068 root      20   0 19128 4132 3468 S    0  0.1   0:00.09 NetworkManager
 1486 gdm       20   0  9088 4024 2340 S    0  0.0   0:00.06 gconfd-2
 1563 root      20   0  6748 3932 2960 S    0  0.0   0:00.03 polkitd
 1170 root      20   0 21376 3444 2856 S    0  0.0   0:00.00 gdm-simple-slav
 1536 root      20   0  8460 3332 2732 S    0  0.0   0:00.03 upowerd
 1678 gdm       20   0 23752 3300 2652 S    0  0.0   0:00.00 gconf-helper
 1686 root      20   0  8952 3252 2592 S    0  0.0   0:00.08 sshd
 1754 root      20   0  6556 3228 1524 S    0  0.0   0:00.16 bash
  962 root      20   0 19536 3140 2680 S    0  0.0   0:00.01 gdm-binary
 1098 root      20   0 21184 3024 2396 S    0  0.0   0:00.02 console-kit-dae
```

wa -> iowait: Amount of time the CPU has been waiting for I/O to complete.



- The FileSystem Cache

- Field Caches - uninverting the index. The index goes from term -> doc - now we want doc -> term!

- Sorting, faceting, many fields, oh my.

Lucene 3 and before...

Non Locale String FieldCache uses two arrays

[Dog,Horse,Lion,Zebra]	[0,2,1,3,2,1,1,1,3,3,2,1,4,2,1,2]
4 unique terms	16 documents

Things are changing...shared prefix, packed ints, byte arrays, oh my again.

- Solr Caches
- Query Result Cache, Document Cache, Filter Cache

CACHE	
name:	queryResultCache
class:	org.apache.solr.search.LRUCache
version:	1.0
description:	LRU Cache(maxSize=512, initialSize=512)
stats:	lookups : 26463 hits : 26462 hitratio : 0.99 inserts : 2 evictions : 0 size : 2 warmupTime : 0 cumulative_lookups : 26463 cumulative_hits : 26462 cumulative_hitratio : 0.99 cumulative_inserts : 1 cumulative_evictions : 0

- JVM Settings, Garbage Collection

http://www.lucidimagination.com/Community/Hear-from-the-Experts/Articles/Scaling-Lucene-and-Solr

http://www.lucidimagination.com/blog/2011/03/27/garbage-collection-bootcamp-1-0/

Performance Monitoring

[Drink] provokes the desire, but it takes away the performance.

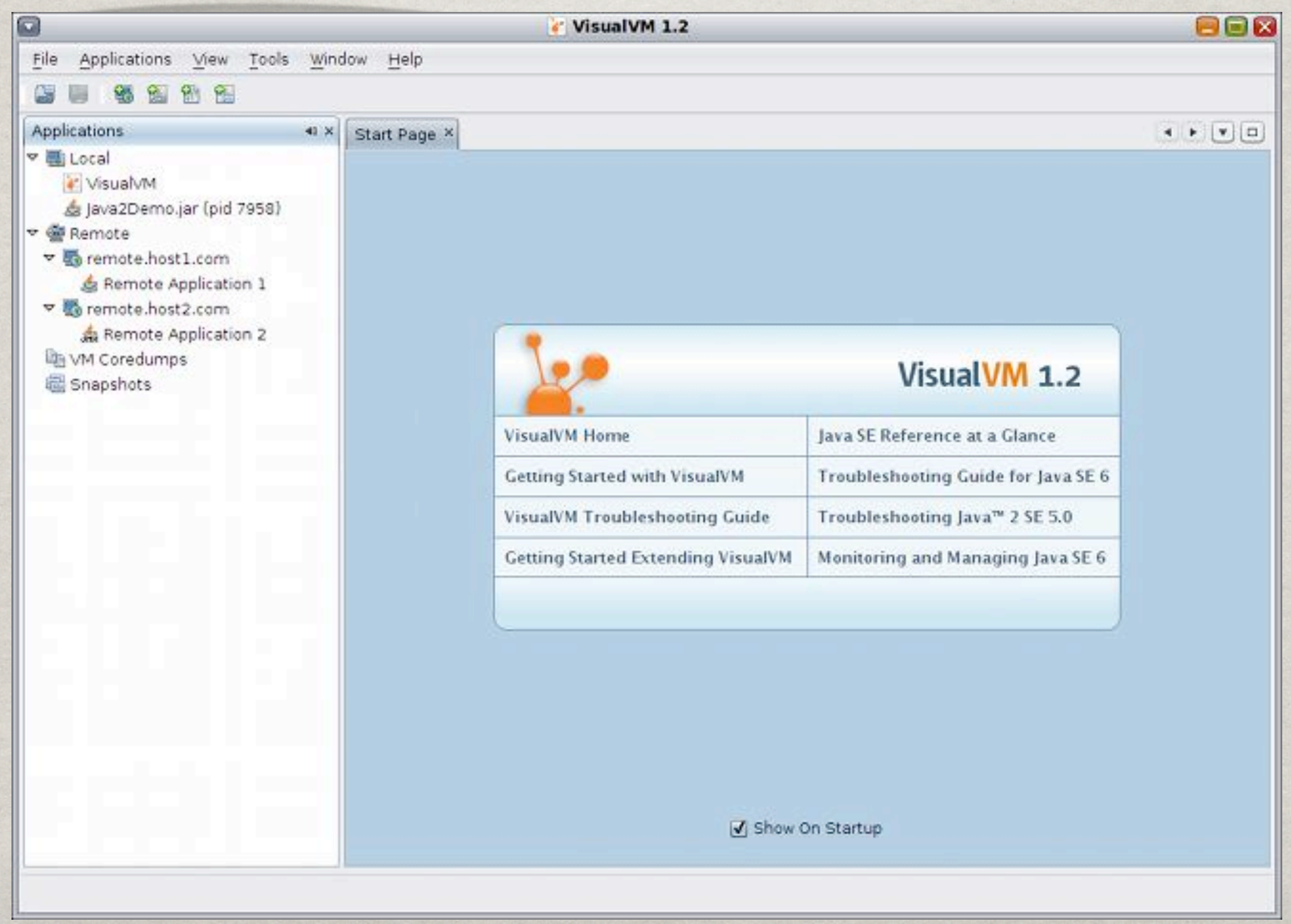
William Shakespeare (1564 - 1616), Macbeth, Act II, sc. 3

"Bullshit!"

- What the $#!#$%@ is happening to my Field Caches!?

- A lot of times, getting the best performance first means figuring out what is not performing. There are a lot of moving parts here!

JConsole's Cooler Older Brother



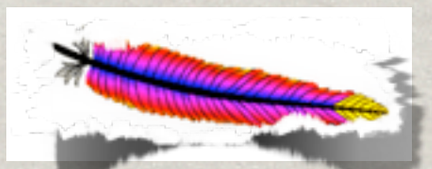
Developer Tools! WTF!?

“For germans showers in the US are simply too complicated.”

- Uwe Schindler

- VisualVM is easy! Solr **Users** should be able to use it too!! German Solr Users too!!

At worst, do a little inspection, deepen your knowledge, and *then* come to the Solr user mailing list with what you have found.



org.mortbay.start.Main (pid 7007)

Overview | Monitor | Threads | Sampler | Profiler | MBeans | Visual GC | Tracer

org.mortbay.start.Main (pid 7007)

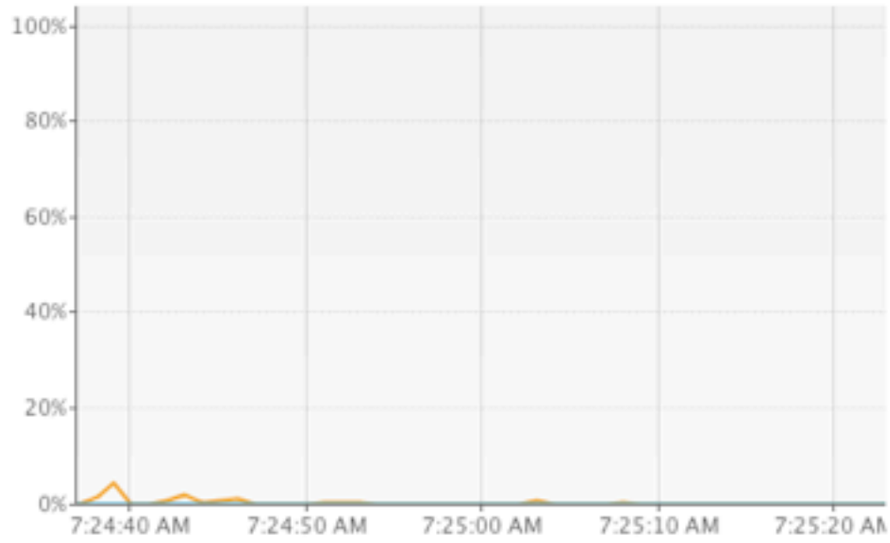
Monitor — CPU ☑ Memory ☑ Classes ☑ Threads ☑

Uptime: 12 hrs 23 min 16 sec

Perform GC | Heap Dump

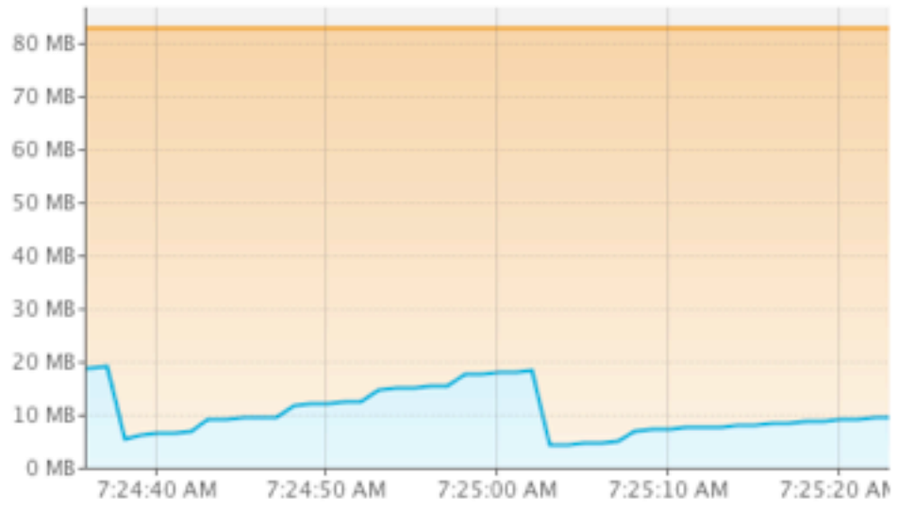
CPU

CPU usage: 0.4% **GC activity:** 0.0%



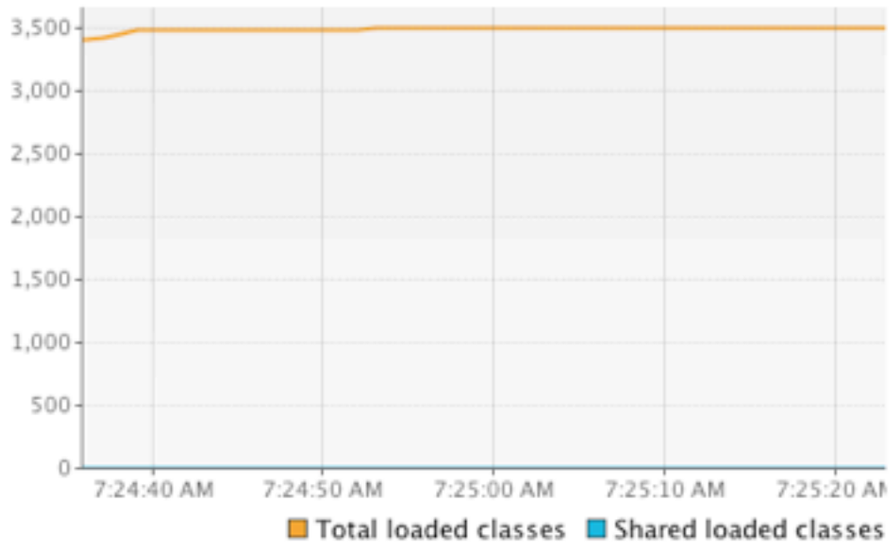
Heap | PermGen

Size: 87,162,880 B **Used:** 10,324,600 B
Max: 536,870,912 B



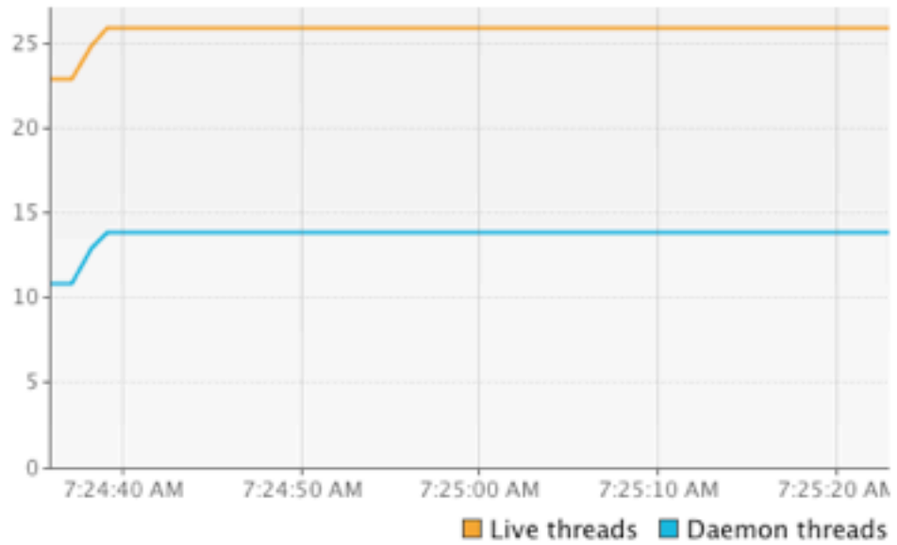
Classes

Total loaded: 3,501 **Shared loaded:** 0
Total unloaded: 5 **Shared unloaded:** 0



Threads

Live: 26 **Daemon:** 14
Live peak: 26 **Total started:** 28



org.mortbay.start.Main (pid 7007)

Overview | Monitor | Threads | Sampler | Profiler | MBeans | Visual GC | Tracer

org.mortbay.start.Main (pid 7007)

Profiler — Settings

Profile: CPU | Memory | Stop

Status: profiling running (1,484 methods instrumented)

Profiling results

Snapshot

Hot Spots - Method	Self time [%]	Self time	Invocations
org.mortbay.io.ByteArrayBuffer.**readFrom** (java.io.InputStream, int)		4580 ms (59.2%)	3
org.mortbay.jetty.bio.SocketConnector$Connection.**run** ()		2949 ms (38.1%)	5
org.mortbay.jetty.webapp.WebAppClassLoader.**loadClass** (String, boolean)		156 ms (2%)	11
org.apache.solr.core.Config.**getNode** (String, boolean)		3.10 ms (0%)	2
org.apache.solr.common.util.FastWriter.**flush** ()		2.93 ms (0%)	2
org.slf4j.impl.JDK14LoggerAdapter.**log**(String, java.util.logging.Level, String, T...		1.66 ms (0%)	4
org.mortbay.util.Scanner.**scanFiles** ()		1.34 ms (0%)	11
org.apache.solr.response.XMLResponseWriter.**write**(java.io.Writer, org.apache....		0.997 ms (0%)	2
org.slf4j.impl.JDK14LoggerAdapter.**fillCallerData**(String, java.util.logging.LogR...		0.993 ms (0%)	4
org.mortbay.jetty.HttpParser.**parseNext** ()		0.756 ms (0%)	3
org.apache.solr.servlet.SolrDispatchFilter.**writeResponse**(org.apache.solr.resp...		0.712 ms (0%)	2
org.mortbay.io.ByteArrayBuffer.**get** ()		0.685 ms (0%)	903
org.apache.solr.servlet.SolrDispatchFilter.**doFilter**(javax.servlet.ServletRequest...		0.601 ms (0%)	2
org.apache.solr.core.SolrCore.**execute**(org.apache.solr.request.SolrRequestHa...		0.460 ms (0%)	2
org.mortbay.util.Scanner.**reportDifferences** (java.util.Map, java.util.Map)		0.397 ms (0%)	11
org.apache.solr.common.params.DefaultSolrParams.**get** (String)		0.365 ms (0%)	60
org.apache.lucene.search.MatchAllDocsQuery.**createWeight**(org.apache.lucen...		0.361 ms (0%)	1
org.apache.solr.handler.component.SearchHandler.**handleRequestBody**(org.a...		0.344 ms (0%)	2
org.apache.solr.search.SolrQueryParser.**checkAllowLeadingWildcards** ()		0.320 ms (0%)	2
org.mortbay.jetty.servlet.ServletHandler.**handle**(String, javax.servlet.http.Http...		0.316 ms (0%)	2
org.mortbay.jetty.handler.ContextHandler.**handle**(String, javax.servlet.http.Htt...		0.314 ms (0%)	2
org.mortbay.jetty.HttpConnection.**handleRequest** ()		0.313 ms (0%)	2
org.apache.lucene.analysis.synonym.SynonymFilter.**incrementToken** ()		0.302 ms (0%)	2
org.mortbay.jetty.Request.**getCookies** ()		0.297 ms (0%)	2
org.apache.solr.handler.component.QueryComponent.**prepare**(org.apache.solr...		0.287 ms (0%)	2
org.mortbay.jetty.Response.**setContentType** (String)		0.278 ms (0%)	2
org.apache.solr.search.ReturnFields.**<init>**(String[], org.apache.solr.request.So...		0.265 ms (0%)	2
org.apache.solr.common.util.FastWriter.**write** (int)		0.264 ms (0%)	151

[Method Name Filter]

org.mortbay.start.Main (pid 7007)

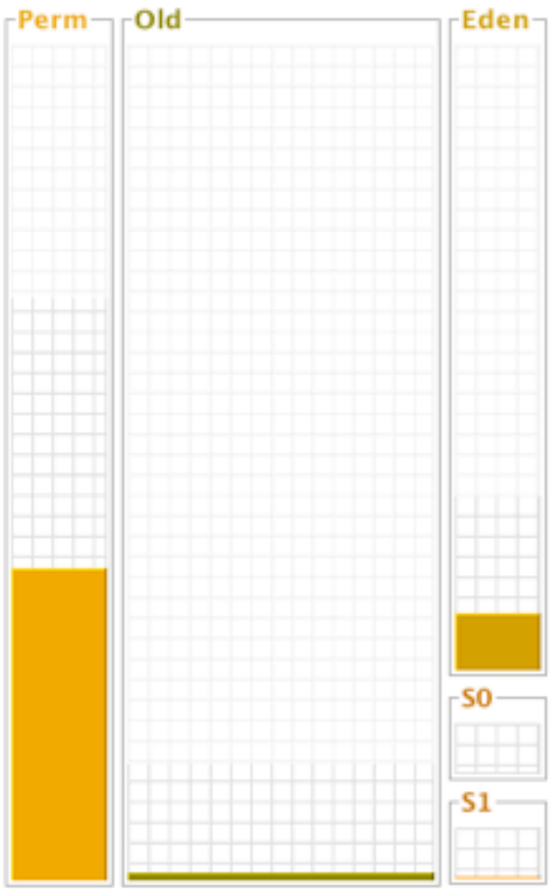
Overview | Monitor | Threads | Sampler | Profiler | MBeans | Visual GC | Tracer

org.mortbay.start.Main (pid 7007)

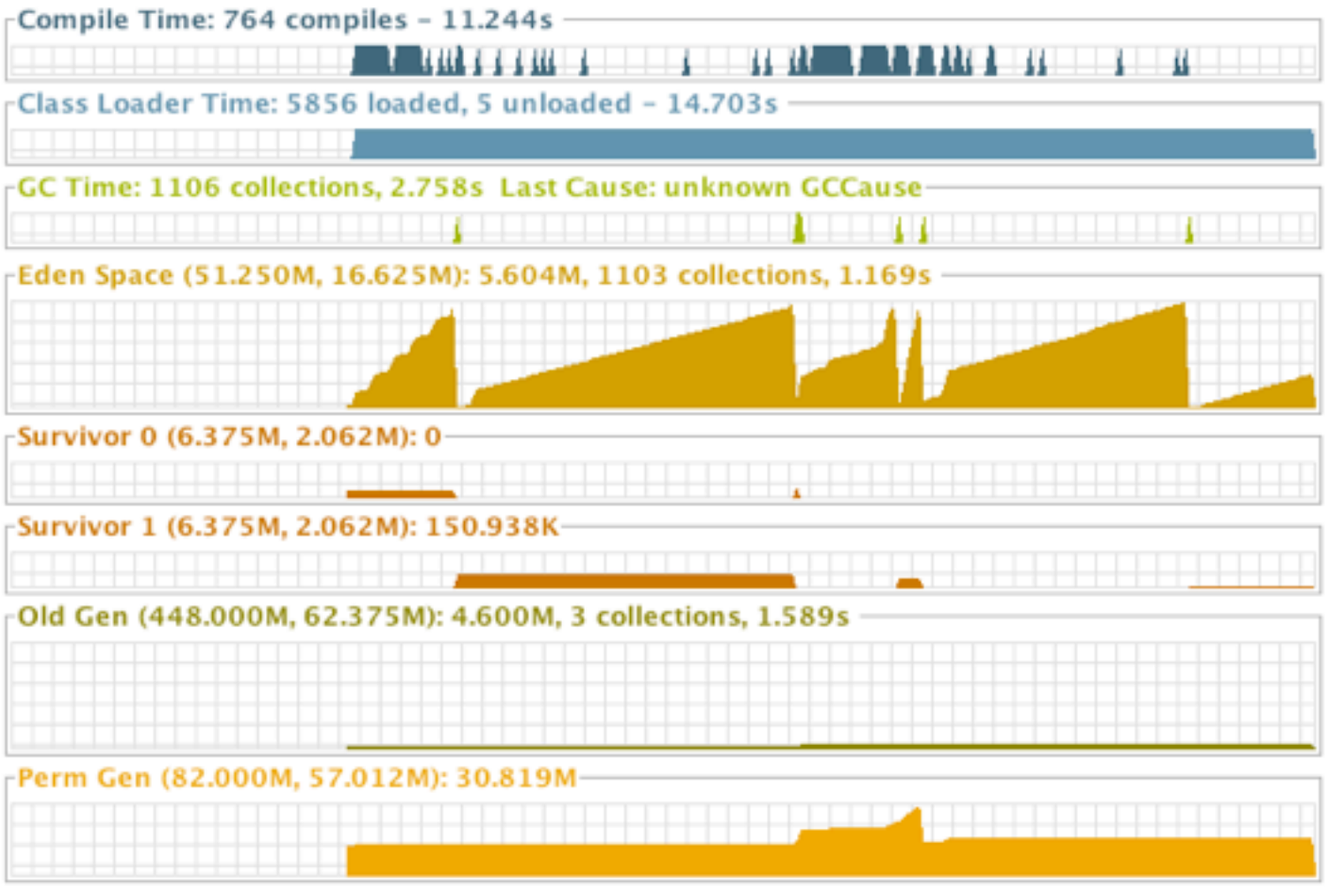
Visual GC — Spaces ☑ Graphs ☑ Histogram ☑

Refresh rate: Auto msec.

Spaces



Graphs

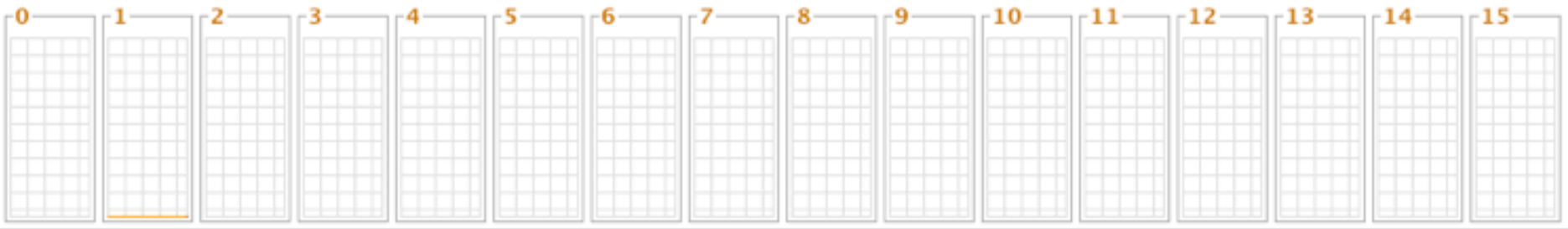


Histogram

Parameters

Tenuring Threshold: 4 Max Tenuring Threshold: 4 Desired Survivor Size: 1081344 Current Survivor Size: 2162688

Histogram



org.mortbay.start.Main (pid 7007)

Overview | Monitor | Threads | Sampler | Profiler | MBeans | Visual GC | Tracer

org.mortbay.start.Main (pid 7007)

Tracer — Probes ☑ Timeline ☑ Details ☑

Start | Stop

Probes

- JIT Compiler
  - ☐ JIT Compiler — Monitors the JIT compiler activity
- Garbage Collectors
- Java I/O
  - ☑ File Descriptors — Measures number of available and used file
  - ☐ Java Files Utilization — Measures read/write rates of Java Files.
  - ☐ NIO Utilization — Measures read/write rates of NIO.
- Jvmstat counters 'java.ci'
- Jvmstat counters 'java.cls'
- Jvmstat counters 'java.threads'

Timeline

Probes | 7:41:00 AM | 7:41:10 AM | 7:41:20 AM

File Descriptors — 10,240 — Available — Used — 0

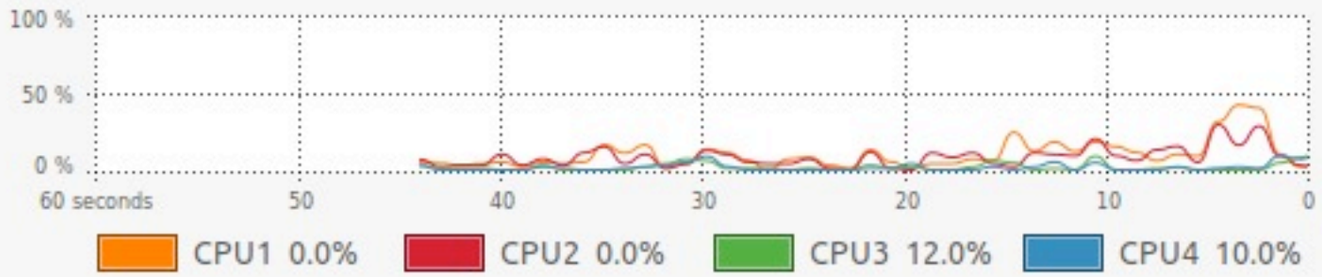
Details

Mark	Time [ms]	Available	Used
☐	7:41:07.459 AM	10,240	113
☐	7:41:08.459 AM	10,240	113
☐	7:41:09.459 AM	10,240	113
☐	7:41:10.459 AM	10,240	113
☐	7:41:11.460 AM	10,240	113
☐	7:41:12.459 AM	10,240	113
☐	7:41:13.459 AM	10,240	113
☐	7:41:14.459 AM	10,240	113
☐	7:41:15.459 AM	10,240	113
☐	7:41:16.460 AM	10,240	113
☐	7:41:17.459 AM	10,240	113
☐	7:41:18.460 AM	10,240	113
☐	7:41:19.459 AM	10,240	113
☐	7:41:20.459 AM	10,240	113
☐	7:41:21.459 AM	10,240	113
☐	7:41:22.459 AM	10,240	113

System Monitor

System | Processes | Resources | File Systems

CPU History

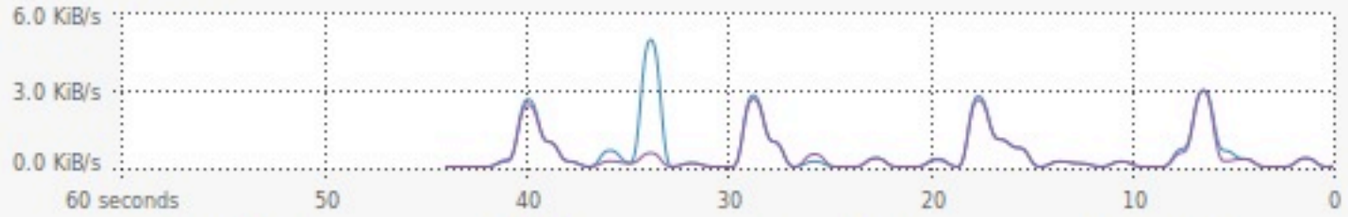


Memory and Swap History

Memory
1.9 GiB (24.2 %) of 7.8 GiB

Swap
0 bytes (0.0 %) of 13.2 GiB

Network History



Receiving	0 bytes/s	Sending	73 bytes/s
Total Received	1.1 GiB	Total Sent	221.0 MiB

Top

```
top - 09:24:23 up 3 min,  1 user,  load average: 1.26, 0.88, 0.37
Tasks: 168 total,   1 running, 167 sleeping,   0 stopped,   0 zombie
Cpu(s):  0.1%us,  0.2%sy,  0.0%ni, 99.8%id,  0.0%wa,  0.0%hi,  0.0%si,  0.0%st
Mem:   8185600k total,   349500k used,  7836100k free,   115772k buffers
Swap:  8193144k total,        0k used,  8193144k free,   115680k cached

  PID USER      PR  NI  VIRT  RES  SHR S %CPU %MEM    TIME+  COMMAND
 1261 root      20   0 57940  17m 7120 S    0  0.2   0:00.62 Xorg
 1531 gdm       20   0 85884  13m  10m S    0  0.2   0:00.13 gdm-simple-gree
 1493 gdm       20   0 77332  10m 7972 S    0  0.1   0:00.16 gnome-settings-
 1528 gdm       20   0 73200 9404 7412 S    0  0.1   0:00.05 metacity
 1530 gdm       20   0 31568 7540 6172 S    0  0.1   0:00.04 gnome-power-man
 1468 gdm       20   0 39368 6268 5176 S    0  0.1   0:00.04 gnome-session
 1557 gdm        9 -11 99500 4700 3576 S    0  0.1   0:00.09 pulseaudio
 1068 root      20   0 19128 4132 3468 S    0  0.1   0:00.09 NetworkManager
 1486 gdm       20   0  9088 4024 2340 S    0  0.0   0:00.06 gconfd-2
 1563 root      20   0  6748 3932 2960 S    0  0.0   0:00.03 polkitd
 1170 root      20   0 21376 3444 2856 S    0  0.0   0:00.00 gdm-simple-slav
 1536 root      20   0  8460 3332 2732 S    0  0.0   0:00.03 upowerd
 1678 gdm       20   0 23752 3300 2652 S    0  0.0   0:00.00 gconf-helper
 1686 root      20   0  8952 3252 2592 S    0  0.0   0:00.08 sshd
 1754 root      20   0  6556 3228 1524 S    0  0.0   0:00.16 bash
  962 root      20   0 19536 3140 2680 S    0  0.0   0:00.01 gdm-binary
 1098 root      20   0 21184 3024 2396 S    0  0.0   0:00.02 console-kit-dae
```

- Which part of the query is slow?
- Get the time taken in each component for a request.
- Add debug=timing to your query parameters.

- A perf hunting anecdote - what are some of the generic steps to tracking down perf issues?
- Is there disk activity? How is the heap? Timing of various Solr components? What does the Profiler say? Put it all together. Or take your evidence to the Solr User list.

Benchmarking

My 2nd attempt at benchmarking how long I could forget my coffee before meltdown. It did not smell nice.

- Lucene has some nice standard benchmarking tools that many of the developers use.

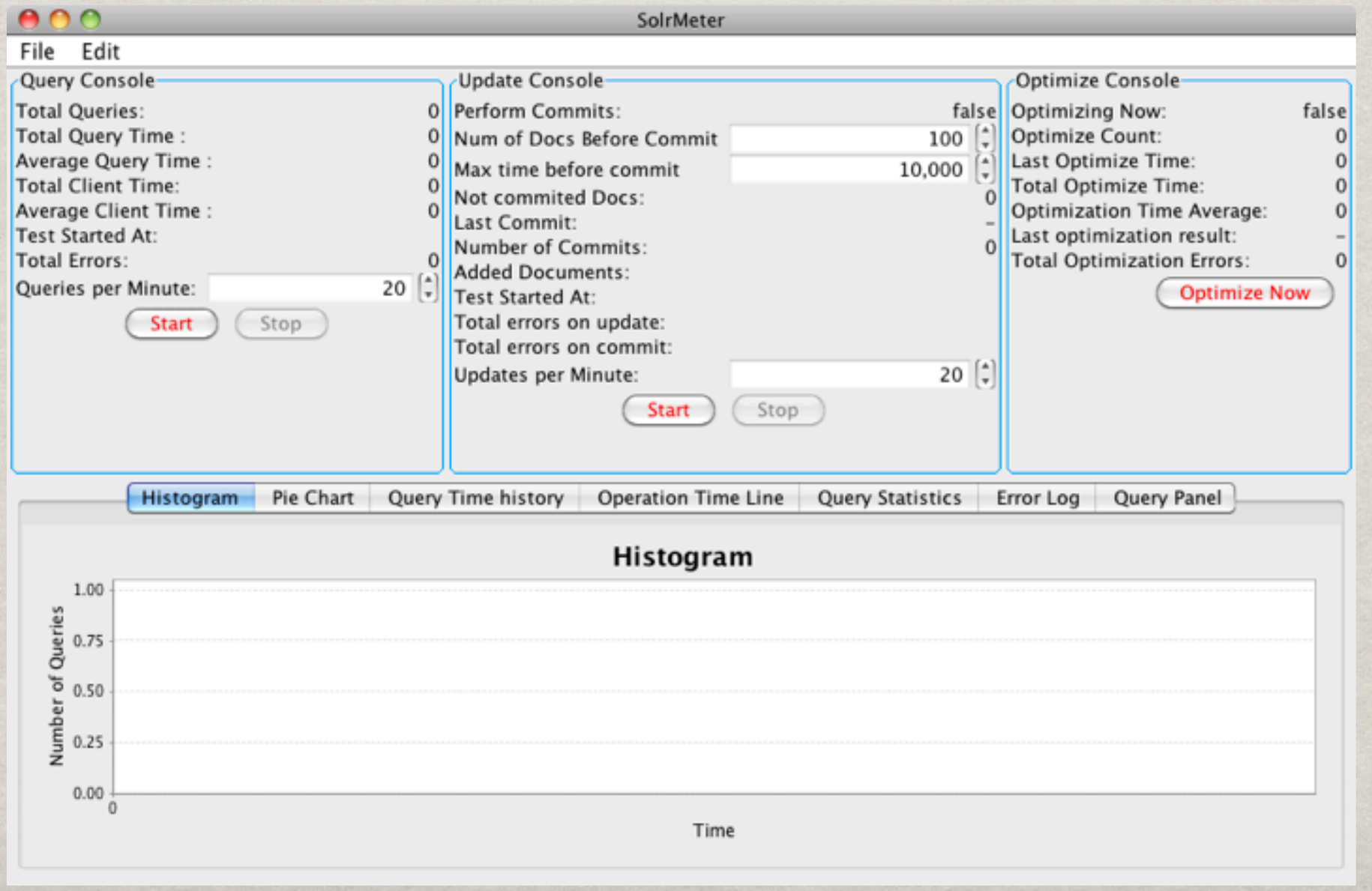
  The benchmark contrib/module.
  Python scripts that drive the benchmark module in an Apache Extra's project.

- Solr developers have taken a much more ad-hoc approach thus far.

- What's common?
- Custom One Off or Couple Off Benchmark Code

0.2

- SolrMeter: http://code.google.com/p/solrmeter/



lwe-load.jmx (/Users/markrmiller/Workspaces/lucid/touchstone/jmeter/lwe-load.jmx) - Apache JMeter (2.4 r961953)

File Edit Run Options Help

0 / 0

- Test Plan
  - User Defined Variables
  - AndHighHigh
    - CSV Data Set Config
    - CSV Data Set Config
    - HTTP Request
      - View Results Tree
    - Summary Report
  - OrHighHigh
    - CSV Data Set Config
    - CSV Data Set Config
    - HTTP Request
      - View Results Tree
    - Summary Report
  - Fuzzy1
    - CSV Data Set Config
    - CSV Data Set Config
    - HTTP Request
      - View Results Tree
    - Summary Report
  - HighFreqExactPhrase
    - CSV Data Set Config
    - CSV Data Set Config
    - HTTP Request
      - View Results Tree
    - Summary Report

HTTP Request

Name: HTTP Request

Comments:

Web Server

Server Name or IP: ${server} Port Number: ${port}

Timeouts (milliseconds)

Connect: Response:

HTTP Request

Protocol (default http): Method: GET Content encoding:

Path: /solr/collection1/select?json.nl=map&wt=ruby&facet.field=author_display&facet.field=data_source_name&

☐ Redirect Automatically ☑ Follow Redirects ☑ Use KeepAlive ☐ Use multipart/form-data for HTTP POST

Send Parameters With the Request:

Name:	Value	Encode?	Include Equals?
role	DEFAULT	☑	☑
q	${query}	☑	☑
sort	${sort}	☑	☑

Add Delete

Send Files With the Request:

File Path:	Parameter Name:	MIME Type:

- A new Solr benchmarking option

- Written by me, for you, as part of prep for this talk!! Your $$$ got you some code!

- But it's not complete, or yet committed...there are a few things to be worked out...

- But you can start playing with it now if you are adventurous!

- How to easily generate documents for Solr indexing benchmarks?
- Tactic? First, like many times before, like many others before, started ripping code from Lucene benchmark contrib.

- The benchmark module is already generic.
- It already has an expressive algorithm language.
- It already has great document generation and query source code.
- Many developers have added lots to it over the years.
- It just doesn't work with Solr (yet).

Benchmark Docs?

http://lucene.apache.org/java/3_0_2/api/contrib-benchmark/index.html

For newer versions, now that benchmark is a module rather than contrib, you might have to look around a bit.

I've done some work to address this.

Starting with...

- New Benchmark Tasks:
- SolrAddDocTask
- SolrClearIndexTask
- SolrCommitTask
- StartSolrServerTask
- StopSolrServerTask

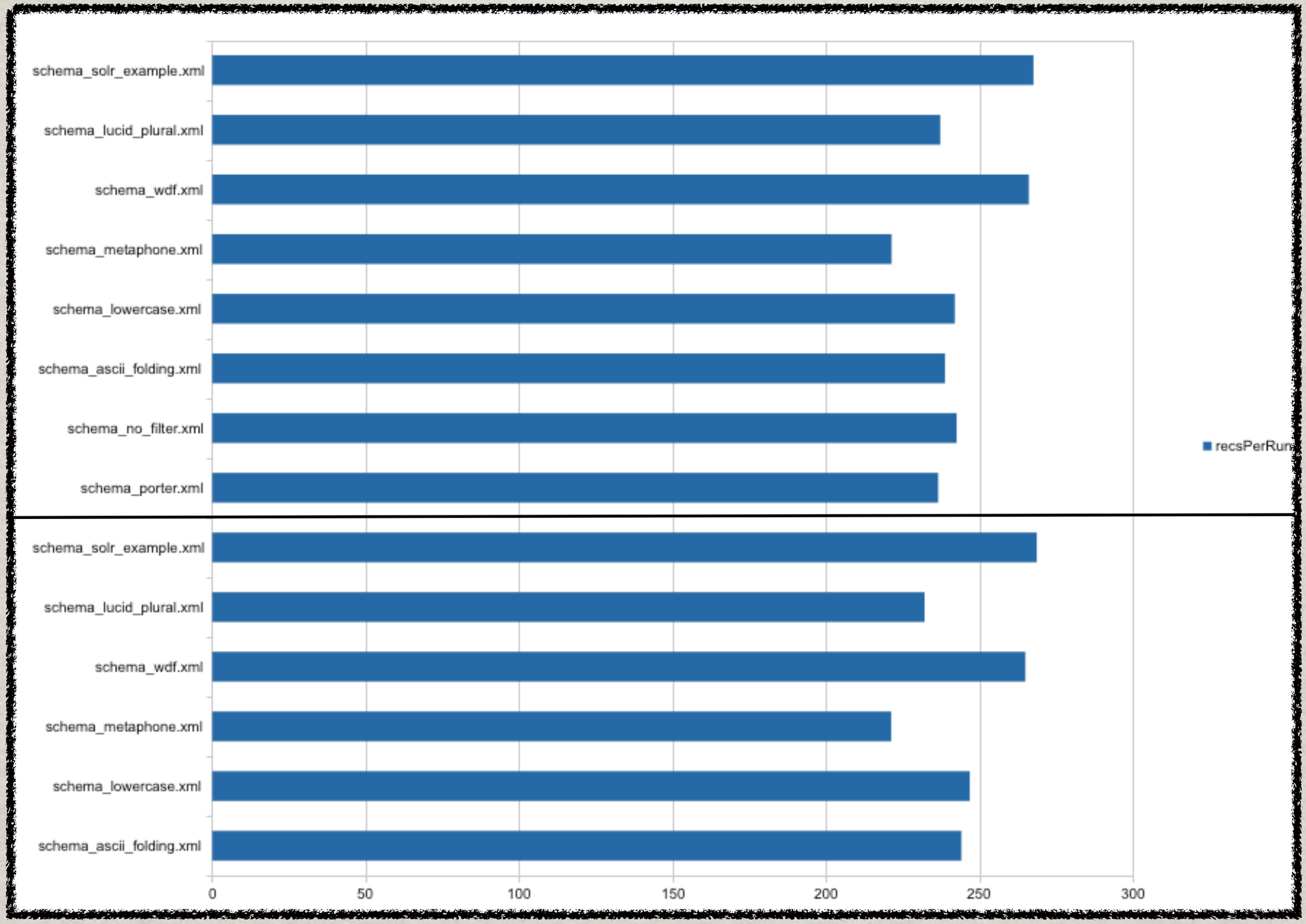
```
log.step=5000

docs.file=/Users/markrmiller/wikidumps/enwiki.out.txt
content.source=org.apache.lucene.benchmark.byTask.feeds.LineDocSource

solr.server=org.apache.solr.client.solrj.impl.StreamingUpdateSolrServer
#solr.server=org.apache.solr.client.solrj.impl.CommonsHttpSolrServer
solr.url=http://192.168.1.200:8983/solr

solr.field.mappings=doctitle>title,docid>id,docname>text,docdate>text,docdatenum>text,
doctimesecnum>text,body>text


solr.streaming.server.queue.size=10
solr.streaming.server.threadcount=5

query.maker=org.apache.lucene.benchmark.byTask.feeds.ReutersQueryMaker

# task at this depth or less would print when they start
task.max.depth.log=2

log.queries=false
#
------------------------------------------------------------------------------
```

Easy to vary settings per round:

```
solr.server=solrserver_impl:org.apache.solr.client.solrj.im
pl.StreamingUpdateSolrServer:solr.server=org.apache.solr.cl
ient.solrj.impl.CommonsHttpSolrServer
```

```
#
-----------------------------------------------------------------

{ "Rounds"

    SolrClearIndex
    ResetSystemErase

    [ "Populate"
        { "MAddDocs" SolrAddDoc > : 20000
        { BlockSolrServerTillFinish >
    ] : 3

     SolrCommit

    NewRound

} : 4

RepSumByName
RepSumByPrefRound MAddDocs
RepSumByPrefRound Populate
```

```
    [java] ------------> Report sum by Prefix (Populate) and Round (14 about 14 out of 156)
    [java] Operation   round schema                    runCnt   recsPerRun        rec/s  elapsedSec    avgUsedMem    avgTotalMem
    [java] Populate        0 schema_ascii_folding.xml      1        15001       244.20       61.43   109,249,064    264,110,080
    [java] Populate -  -   1 schema_lowercase.xml -  -  -  - 1 -  -   15001 -  -  246.91 -  -  60.76 -  26,536,264 -  264,830,976
    [java] Populate        2 schema_metaphone.xml          1        15001       221.33       67.78   131,362,496    264,503,296
    [java] Populate -  -   3 schema_wdf.xml -  -  -  -  -  -  - 1 -  -   15001 -  -  265.04 -  -  56.60 -  84,795,792 -  264,503,296
    [java] Populate        4 schema_lucid_plural.xml       1        15001       232.18       64.61    93,306,864    264,306,688
    [java] Populate -  -   5 schema_solr_example.xml -  -  - 1 -  -   15001 -  -  268.76 -  -  55.81 -  75,910,520 -  264,437,760
    [java] Populate        6 schema_porter.xml             1        15001       236.62       63.40   182,119,104    264,044,544
    [java] Populate -  -   7 schema_no_filter.xml -  -  -  - 1 -  -   15001 -  -  242.64 -  -  61.83 - 113,457,016 -  264,568,832
    [java] Populate        8 schema_ascii_folding.xml      1        15001       238.86       62.80   115,160,256    264,765,440
    [java] Populate -  -   9 schema_lowercase.xml -  -  -  - 1 -  -   15001 -  -  242.05 -  -  61.97 -  26,807,736 -  264,503,296
    [java] Populate       10 schema_metaphone.xml          1        15001       221.45       67.74    31,253,776    264,241,152
    [java] Populate -  -  11 schema_wdf.xml -  -  -  -  -  -  - 1 -  -   15001 -  -  266.17 -  -  56.36 - 170,388,704 -  264,503,296
    [java] Populate       12 schema_lucid_plural.xml       1        15001       237.31       63.21   174,202,544    264,503,296
    [java] Populate -  -  13 schema_solr_example.xml -  -  - 1 -  -   15001 -  -  267.68 -  -  56.04 -  36,058,616 -  264,634,368
    [java] ####################
    [java] ###  D O N E !!! ###
    [java] ####################
BUILD SUCCESSFUL
Total time: 18 minutes 43 seconds
```

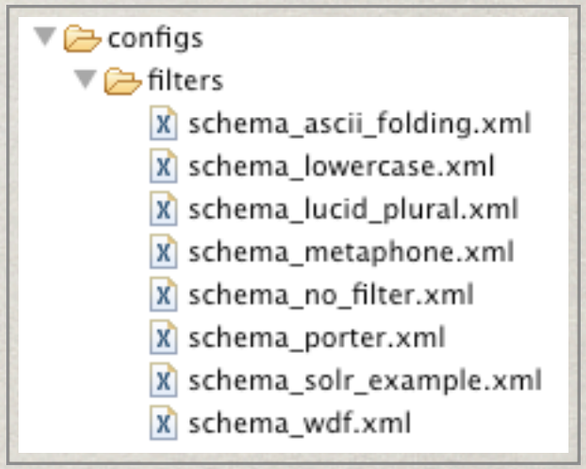

schema_solr_example.xml
schema_lucid_plural.xml
schema_wdf.xml
schema_metaphone.xml
schema_lowercase.xml
schema_ascii_folding.xml
schema_no_filter.xml
schema_porter.xml
recsPerRun
schema_solr_example.xml
schema_lucid_plural.xml
schema_wdf.xml
schema_metaphone.xml
schema_lowercase.xml
schema_ascii_folding.xml
0
50
100
150
200
250
300

- What other cool things?

- Local or Remote - local allows starting a new JVM during the algorithm run - remote lets you hit any remote or locally running Solr instance.

- Select Solr config files per round.

- Select config per round allows you to easily setup a wide variety of configurations for testing - add in a little version control and now you are cooking with fire.



solr.configs.home=/configs/filters
solr.schema=schema:schema_ascii_folding.xml:schema_lowercase.xml:schema_metaphone.xml

APACHE LUCENE EUROCON

17-18 October 2011 | Training
19-20 October 2011 | Conference

CALL FOR PARTICIPATION NOW OPEN:

http://2011.lucene-eurocon.org

PRESENTED BY:

lucid IMAGINATION

ALL PROCEEDS BENEFIT THE APACHE SOFTWARE FOUNDATION