UCL

A Performance Analysis of Distributed Indexing using Terrier

Amaury Couste <a.couste@ucl.ac.uk>
Jakub Kozłowski <j.kozlowski@ucl.ac.uk>
William Martin <w.martin@ucl.ac.uk>

Indexing

Indexing

- Used by search engines.

- Facilitates fast, accurate information retrieval.

Indexing

- Traditionally done on a single machine.
- Easy to implement.

But...

- Datasets can be very large.
- A single machine takes too long - over 1 day for ~25 million documents*.

* From 'Comparing Distributed Indexing: To MapReduce or Not?' by McCreadie et al. (2009).

Solution?

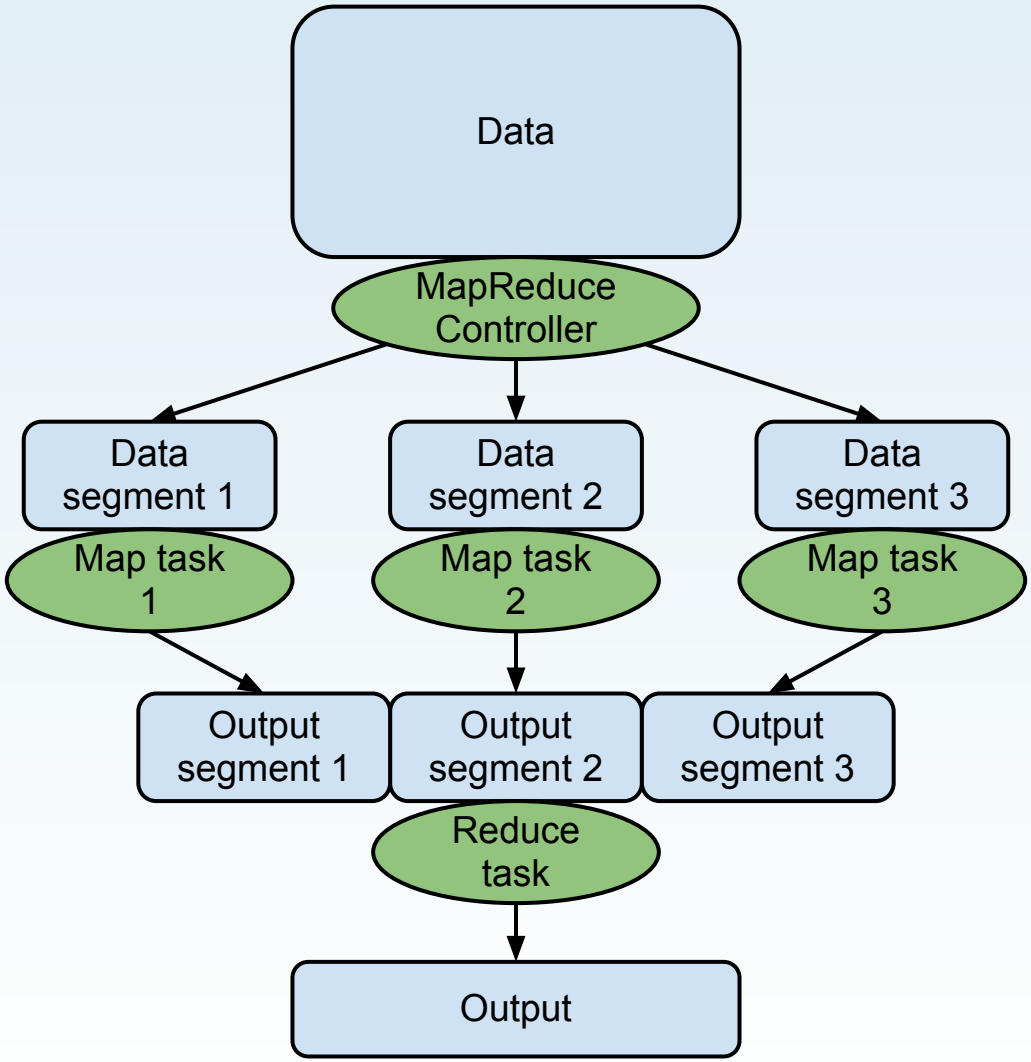
- Use multiple machines:
  - Distribute the work.
  - Distribute the dataset.

MapReduce Framework

- Introduced by Google in 2004*.
- Split work into a number of **Map** and **Reduce** tasks.
- **Map** tasks do the work.
- **Reduce** tasks aggregate the results.

* 'MapReduce: simplified data processing on large clusters' by Dean & Ghemawat (2004).

MapReduce Framework



Hadoop

- Java implementation of MapReduce framework.
- Makes it fast and easy to bring distributed work to applications.
- Hadoop Distributed File System (HDFS).
- Can we use it for indexing?

Indexers

Apache Lucene

- High-performance text search engine.
- Simple, stable API.
- Excellent documentation.
- Commercially friendly open source licence.

But...

- Designed to work on a single machine.

Katta

- Scalable distributed retrieval solution.
- Serves large, replicated indexes as shards.
- Can serve very large Lucene indexes sharded over many servers.
- Experimental solution, currently in development.

Apache Solr

- Open-source enterprise search platform built on top of Apache Lucene.
- Provides a high-availability, distributed retrieval solution.
- Uses replication and sharding.
- Provides language-independent REST-like HTTP/XML and JSON APIs.

Terrier

- Uses HDFS.
- Custom indexer makes use of distributed file-system.
- Decompresses files on the fly.
- Reads many different file formats (ClueWeb09B).

Terrier - Single Machine

- 4-core 2.4 Ghz Xeon processor.
- 4 GB RAM.
- 2 x 400 GB hard drives.
- Indexes 425 GB in over 1 day.

From 'Comparing Distributed Indexing: To MapReduce or Not?' by McCreadie et al. (2009).

Terrier - Distributed

- 4 identical machines.
- 3.5 times faster.

But...

- Query time not measured.

From 'Comparing Distributed Indexing: To MapReduce or Not?' by McCreadie et al. (2009).

Experiment

ClueWeb09

- Modern standard for information retrieval.
- Very large corpus to evaluate text retrieval methods.
- ~1 billion text files.
- 10 languages.
- Sample queries.

Questions

- What benefits can distributed solutions offer?
- Are they faster?
- Can we quickly index a large dataset?
- Can we quickly query a large dataset?

Data

- Subset of ClueWeb09B.
- 15 GB compressed.
- > 100 GB uncompressed.

Hardware

Totals...

- 4 physical machines.
- 18 CPU cores.
- 19 GB of RAM.
- 7 TB hard drive storage.

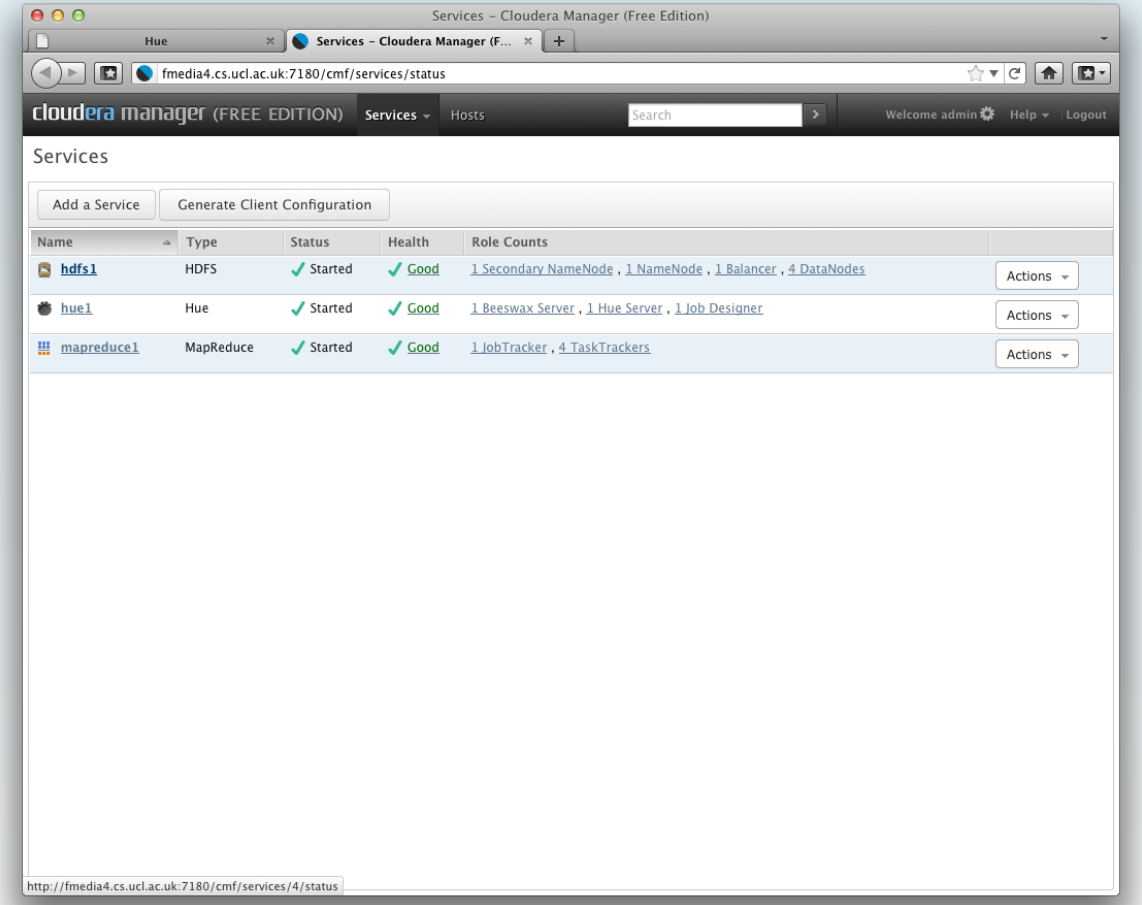
Software

- hadoop v0.20
- terrier v3.5
- cloudera
- hue

Cloudera

- Automatic setup of nodes.
- Simple web-based interface.
- Remotely configure nodes.
- Start and stop services.
- Monitor performance in real-time.
- CDH - custom distribution of Hadoop and related software, packaged with Cloudera Manager.

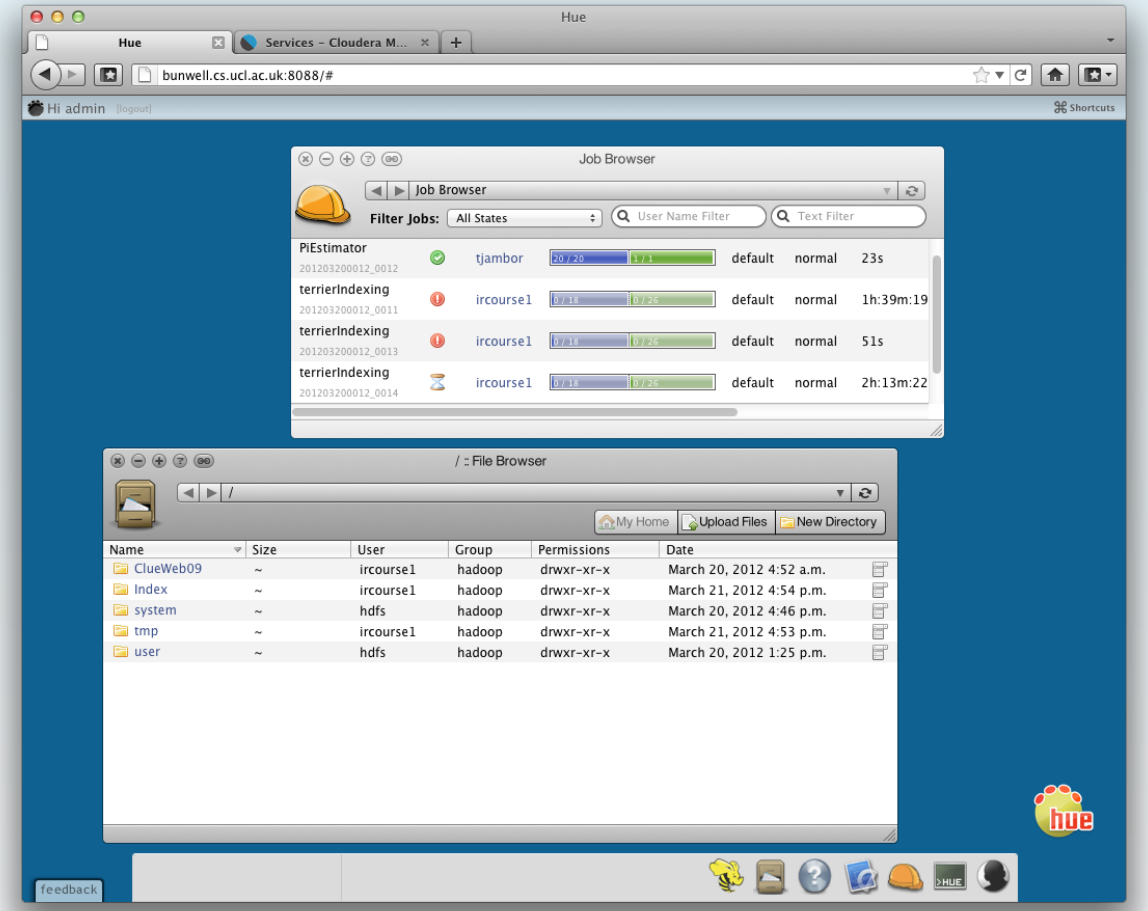
Cloudera


Services – Cloudera Manager (Free Edition)
Hue
Services – Cloudera Manager (F...
fmedia4.cs.ucl.ac.uk:7180/cmf/services/status
cloudera manager (FREE EDITION)
Services
Hosts
Search
Welcome admin
Help
Logout
Services
Add a Service
Generate Client Configuration
Name
Type
Status
Health
Role Counts
hdfs1
HDFS
Started
Good
1 Secondary NameNode , 1 NameNode , 1 Balancer , 4 DataNodes
Actions
hue1
Hue
Started
Good
1 Beeswax Server , 1 Hue Server , 1 Job Designer
Actions
mapreduce1
MapReduce
Started
Good
1 JobTracker , 4 TaskTrackers
Actions
http://fmedia4.cs.ucl.ac.uk:7180/cmf/services/4/status

Hue

- Used to monitor jobs.
- Included in CDH.
- Simple web interface.
- Includes File Browser and Job Browser.
- Monitor the progress of jobs.
- Viewing logs of tasks.
- Indispensable for debugging.

Hue


Hue
Hue
Services - Cloudera M...
bunwell.cs.ucl.ac.uk:8088/#
Hi admin
Shortcuts
Job Browser
Job Browser
Filter Jobs:
All States
User Name Filter
Text Filter
PiEstimator
tjambor
default
normal
23s
terrierIndexing
ircourse1
default
normal
1h:39m:19
terrierIndexing
ircourse1
default
normal
51s
terrierIndexing
ircourse1
default
normal
2h:13m:22
/ :: File Browser
My Home
Upload Files
New Directory
Name
Size
User
Group
Permissions
Date
ClueWeb09
Index
system
tmp
user
hadoop
drwxr-xr-x
March 20, 2012 4:52 a.m.
March 21, 2012 4:54 p.m.
March 20, 2012 4:46 p.m.
March 21, 2012 4:53 p.m.
March 20, 2012 1:25 p.m.
feedback
hue

Hypotheses

- Indexing performance scales linearly with the number of concurrent mappers.

- Querying performance scales linearly with the number of concurrent mappers.

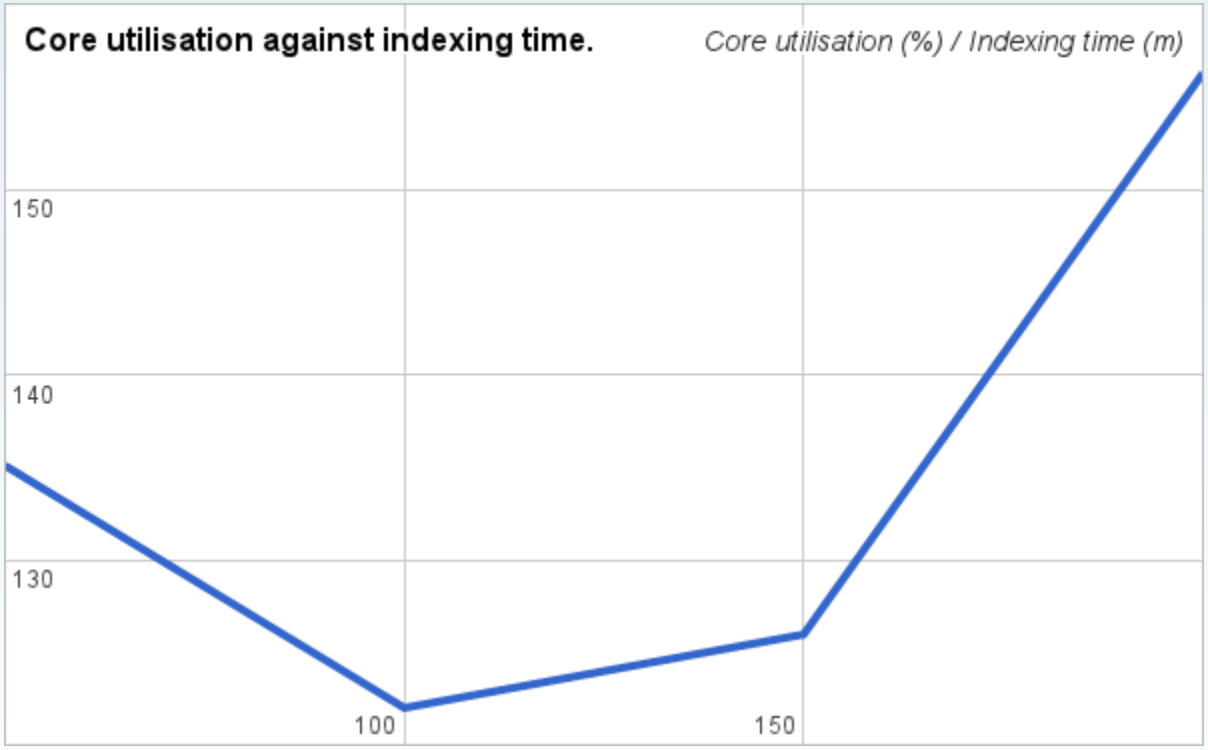
Results

Results - Concurrent Mappers

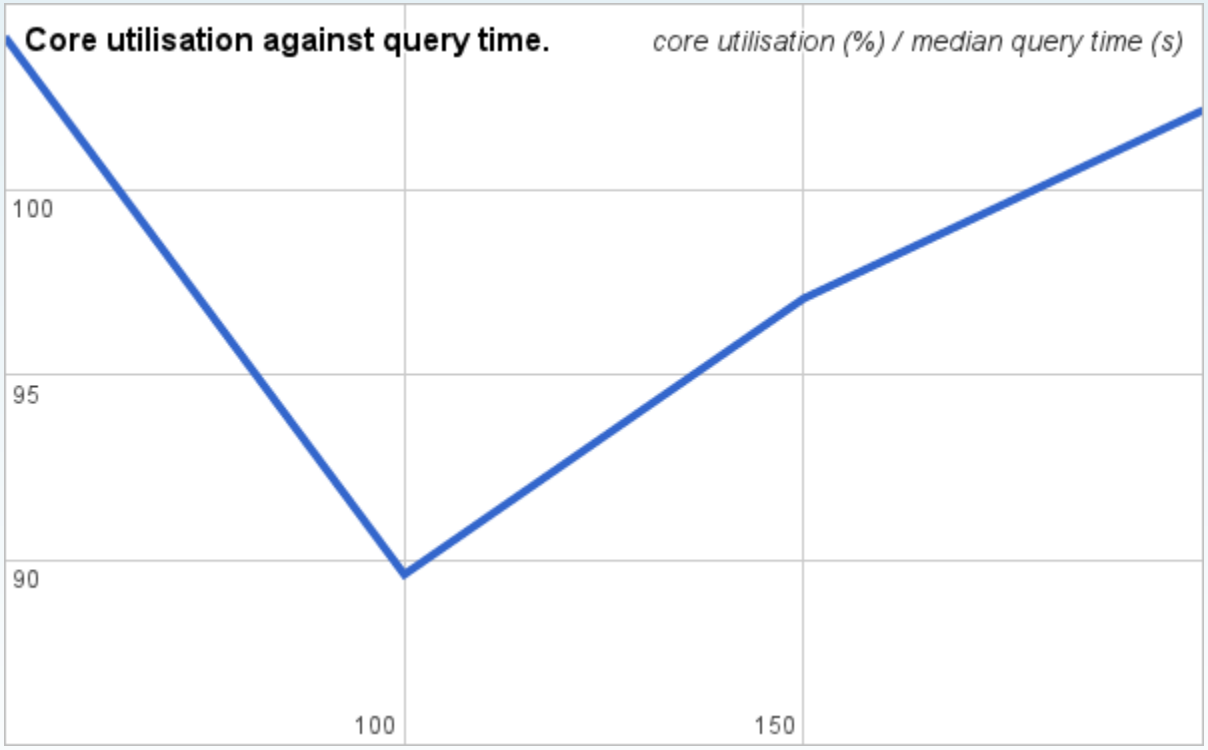
- We fixed the configuration (heap space, data partitioning) to remain constant.

- Adjusted the number of concurrent mappers in order to test the scaling of performance with hardware utilisation.

Results - Concurrent Mappers



Results - Concurrent Mappers

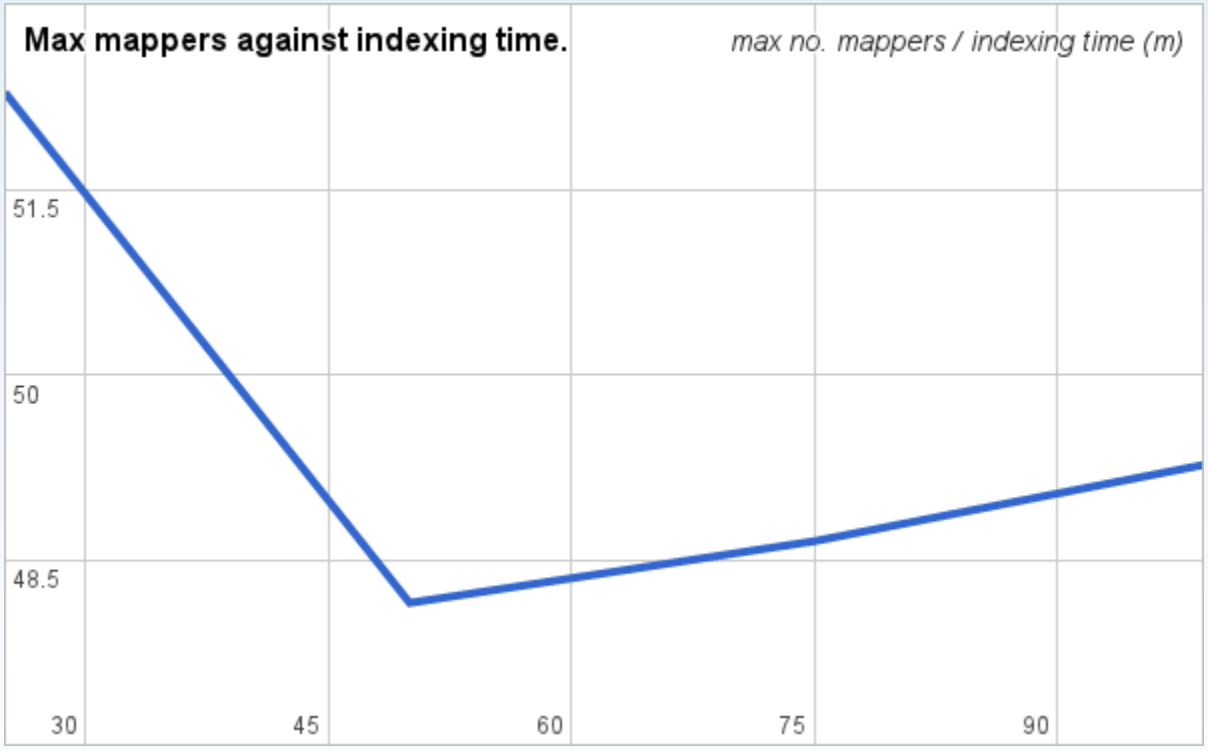


Results - Maximum Mappers

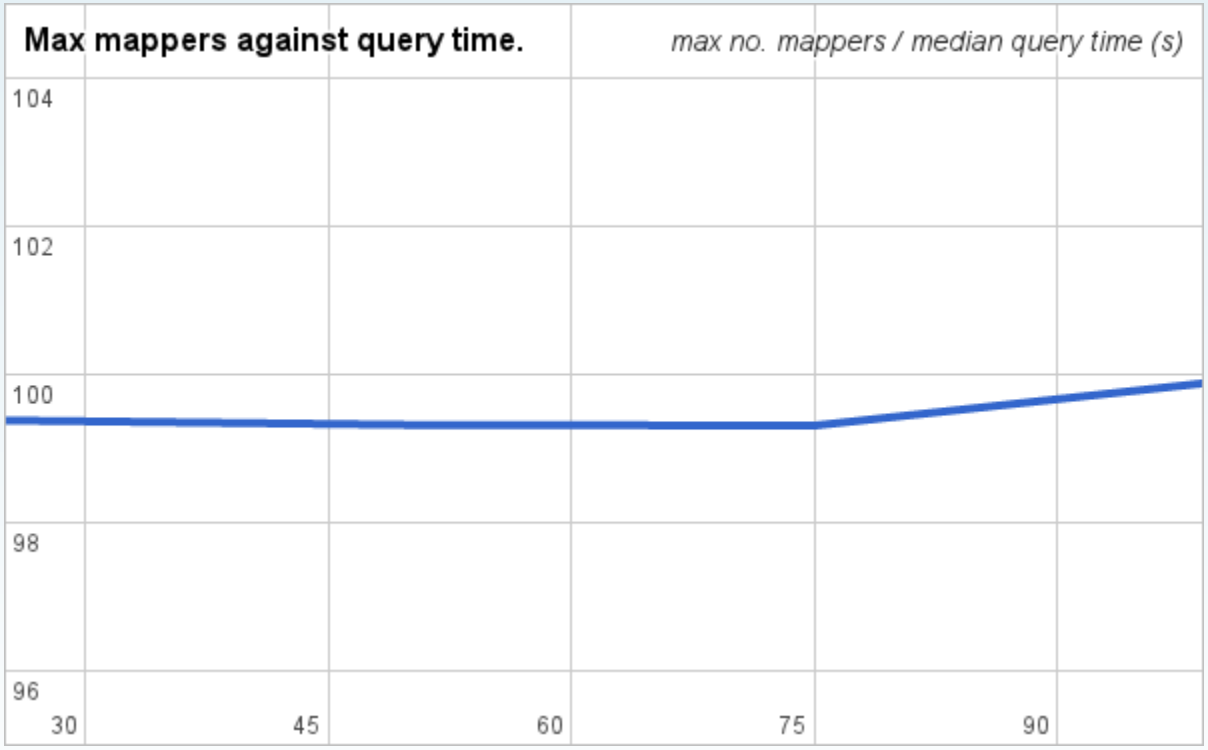
- We set the optimal configuration (heap space, data partitioning).

- Adjusted maximum mappers to test how well terrier utilises its resources.

Results - Maximum Mappers



Results - Maximum Mappers



Issues

Issues

- We encountered a number of issues with terrier.
- Major - single files causing entire experiment to fail.
- Minor - configuration nightmares.

Problem 1

- Job setup code copies jar file dependencies from the CLASSPATH to HDFS.
- Doesn't check whether the files exist on the local hard drive.
- Some are needed - not all.

Problem 2

- To control the partitioning of data into map tasks, we had to add a new configuration parameter to the Terrier configuration architecture.

- Hadoop feature not implemented by Terrier.

Problem 3

- Hadoop logs mapped to /var.
- Jobs fail if log can't be written (low space).
- Moving the logs to the same partition as HDFS on all nodes fixed the issue.

Problem 4

- Jobs always fail on the same files - even a single node cluster!
- Not related to size of file - some were small.
- We had to remove these specific files to continue.

Conclusions

Conclusions

- Performance scales with concurrent mappers up to 100% core utilisation.
- Homogeneous nodes are preferable.

- The best performing configuration is found through trial and error.
- Out of box configuration?

Summary

- Distributed indexing is faster but takes too long to configure.
- Tradeoff - time saved by distribution vs time taken to configure and debug.
- Terrier needs more work.

Questions?