

Project Members: Ruchir Vani(#01416861), Abhijeet Thakare(#01426481)

Project Title: Inverted Indexing using Hadoop's MapReduce and MongoDB

Tools used:

- Hadoop(MapReduce)
- MongoDB

What we did?

- Installed Hadoop framework on Ubuntu.
- Installed MongoDB
- Created inverted index using Hadoop's MapReduce
- Saved inverted index in MongoDB(because NoSQL data model is suitable)

Project Description:

We uploaded input data consisting of text files and a file containing insignificant word to DFS. Using Hadoop's MapReduce created inverted index where each keyword has list of documents it was found in. This inverted index is outputted in output directory on dfs. We also saved inverted index in MongoDB. We wrote a simple java program to query MongoDB where argument is keyword and result is list of documents in which the keyword is present.

What we learned?

- Hadoop Installation(quite a lengthy process)
 - significance of name-node,data-node,job-tracker,task-tracker
- Working of MapReduce
 - flow of computation through map and reduce phases
- commands for managing Hadoop's DFS

- MongoDB's data model
 - practically used (key,value) storage, column,column family which we learned in class
- JAVA API for MongoDB

Few Screenshots:

1. HDFS

HDFS:/user/hduser - Google Chrome

localhost:50075/browseDirectory.jsp?dir=%2Fuser%2Fhduser&namenodeInfoPort=50070&nnaddr=127.0.0.1:54310

Contents of directory /user/hduser

Goto : /user/hduser go

[Go to parent directory](#)

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
input	dir				2014-12-17 14:22	rw-r--r--	hduser	supergroup
output	dir				2014-12-17 18:20	rw-r--r--	hduser	supergroup

[Go back to DFS home](#)

Local logs

[Log](#) directory

[Hadoop](#), 2014.

2. Input directory in hdfs

HDFS:/user/hduser/input - Google Chrome

localhost:50075/browseDirectory.jsp?dir=%2Fuser%2Fhduser%2Finput&namenodeInfoPort=50070&nnaddr=127.0.0.1:54310

Contents of directory /user/hduser/input

Goto : /user/hduser/input go

[Go to parent directory](#)

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
doc1.txt	file	165 B	1	128 MB	2014-12-17 14:22	rw-r--r--	hduser	supergroup
doc2.txt	file	177 B	1	128 MB	2014-12-17 14:22	rw-r--r--	hduser	supergroup
insignificant_words.txt	file	48 B	1	128 MB	2014-12-17 14:22	rw-r--r--	hduser	supergroup

[Go back to DFS home](#)

Local logs

[Log](#) directory

[Hadoop](#), 2014.

3. Output directory in hdfs

HDFS:/user/hduser/output - Google Chrome

HDFS:/user/hduser/ x

localhost:50075/browseDirectory.jsp?dir=%2Fuser%2Fhduser%2Foutput&namenodeInfoPort=50070&nnaddr=127.0.0.1:54310

Contents of directory /user/hduser/output

Goto : /user/hduser/output go

[Go to parent directory](#)

Name	Type	Size	Replication	Block Size	Modification Time	Permission	Owner	Group
SUCCESS	file	0 B	1	128 MB	2014-12-17 18:20	rw-r--r--	hduser	supergroup
part-00000	file	572 B	1	128 MB	2014-12-17 18:20	rw-r--r--	hduser	supergroup

[Go back to DFS home](#)

Local logs

[Log directory](#)

[Hadoop](#), 2014.

4. Inverted index

HDFS:/user/hduser/output/part-00000 - Google Chrome

HDFS:/user/hduser/ x

localhost:50075/browseBlock.jsp?blockId=1073741929&blockSize=572&genstamp=1105&filename=%2Fuser%2Fhduser%2Foutput%2Fpart-00000&d

File: /user/hduser/output/part-00000

Goto : /user/hduser/output go

[Go back to dir listing](#)

[Advanced view/download options](#)

```
abhiheet      doc2.txt
badminton,    doc1.txt
computer      doc1.txt, doc2.txt
cricket doc1.txt
cricket,      doc2.txt
from doc2.txt, doc1.txt
graduate      doc1.txt, doc2.txt
gujrat, doc1.txt
hello doc1.txt, doc2.txt
hobbies doc2.txt, doc1.txt
india doc1.txt, doc2.txt
lowell doc2.txt, doc1.txt
maharashtra, doc2.txt
my doc2.txt, doc1.txt
of doc1.txt, doc2.txt
photography doc2.txt
playing doc1.txt
ruchir doc1.txt
science doc1.txt, doc2.txt
student doc2.txt, doc1.txt
studying      doc1.txt, doc2.txt
swimming,     doc2.txt
tennis, doc2.txt, doc1.txt
umass doc1.txt, doc2.txt
world, doc2.txt, doc1.txt
```

[Download this file](#)

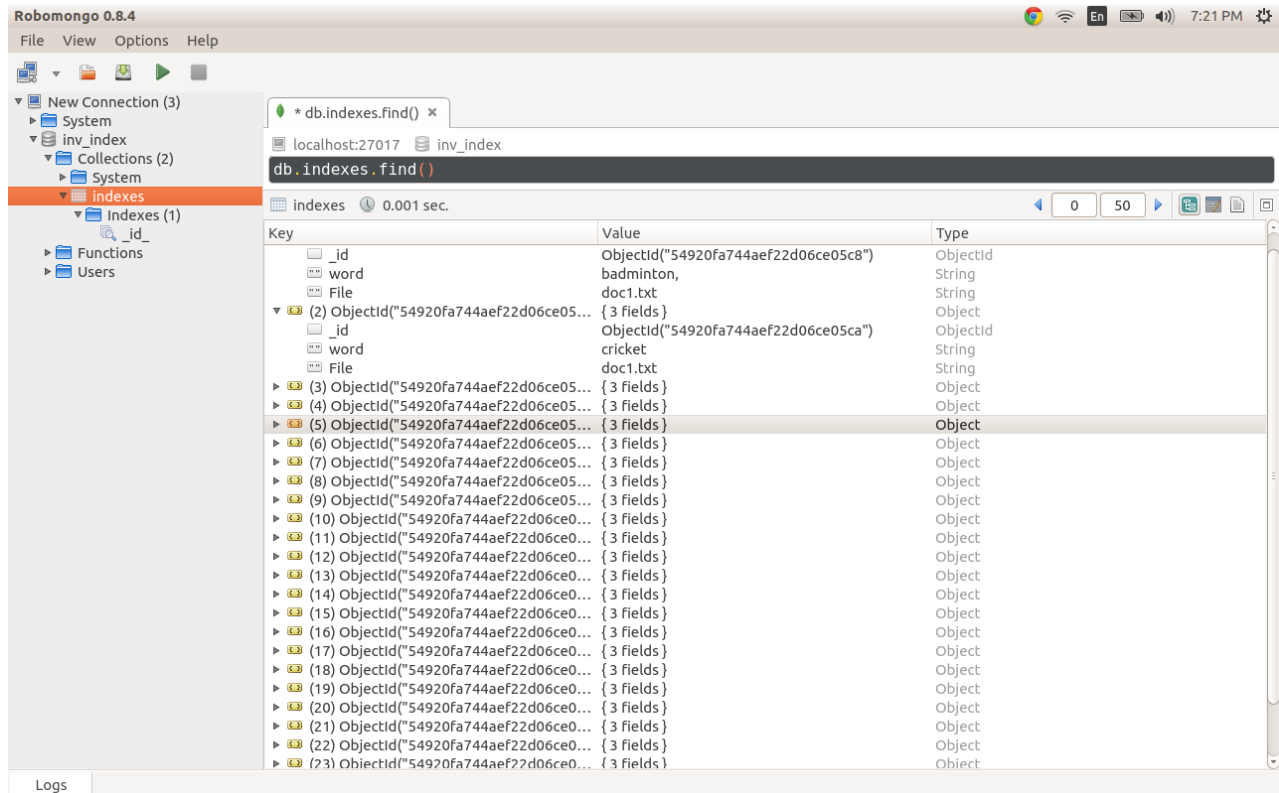
[Tail this file](#)

Chunk size to view (in bytes, up to file's DFS block size): 32768 Refresh

Total number of blocks: 1

1073741929: [127.0.0.1:50010](#) [View Block Info](#)

5. Inverted index stored in mongoDB



Robomongo 0.8.4

File View Options Help

New Connection (3)

- System
- inv_index
 - Collections (2)
 - System
 - indexes
 - Indexes (1)
 - _id
 - Functions
 - Users

* db.indexes.find() x

localhost:27017 inv_index

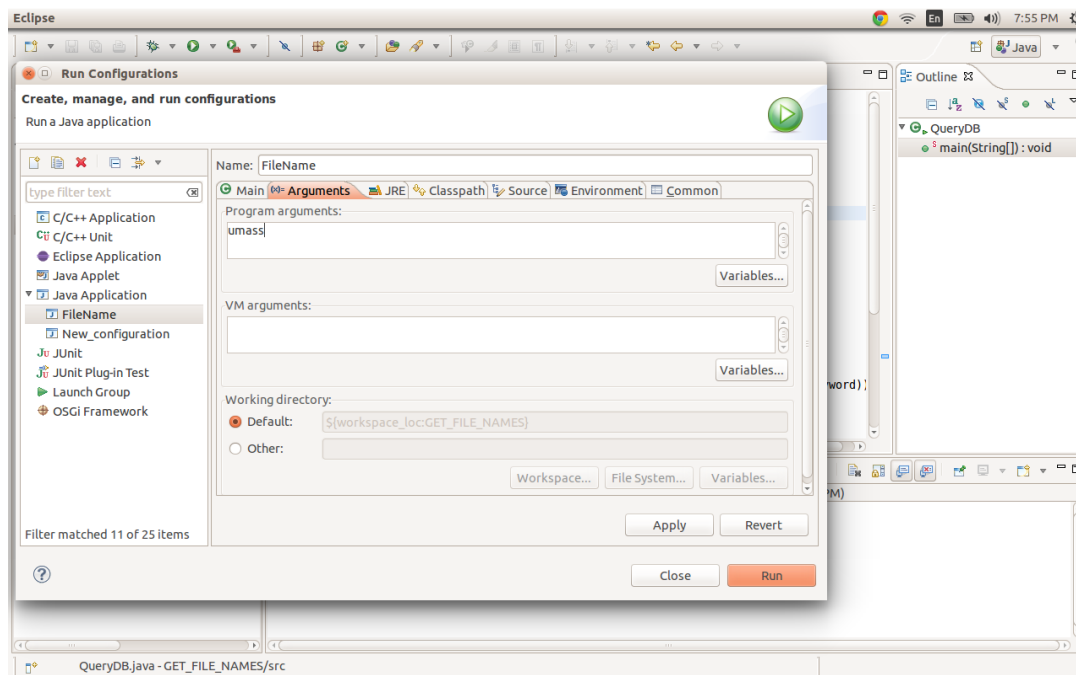
db.indexes.find()

indexes 0.001 sec.

Key	Value	Type
_id	ObjectId("54920fa744aef22d06ce05c8")	ObjectId
word	badminton,	String
File	doc1.txt	String
(2) ObjectId("54920fa744aef22d06ce05c8")	{ 3 fields }	Object
_id	ObjectId("54920fa744aef22d06ce05ca")	ObjectId
word	cricket	String
File	doc1.txt	String
(3) ObjectId("54920fa744aef22d06ce05c8")	{ 3 fields }	Object
(4) ObjectId("54920fa744aef22d06ce05c8")	{ 3 fields }	Object
(5) ObjectId("54920fa744aef22d06ce05c8")	{ 3 fields }	Object
(6) ObjectId("54920fa744aef22d06ce05c8")	{ 3 fields }	Object
(7) ObjectId("54920fa744aef22d06ce05c8")	{ 3 fields }	Object
(8) ObjectId("54920fa744aef22d06ce05c8")	{ 3 fields }	Object
(9) ObjectId("54920fa744aef22d06ce05c8")	{ 3 fields }	Object
(10) ObjectId("54920fa744aef22d06ce05c8")	{ 3 fields }	Object
(11) ObjectId("54920fa744aef22d06ce05c8")	{ 3 fields }	Object
(12) ObjectId("54920fa744aef22d06ce05c8")	{ 3 fields }	Object
(13) ObjectId("54920fa744aef22d06ce05c8")	{ 3 fields }	Object
(14) ObjectId("54920fa744aef22d06ce05c8")	{ 3 fields }	Object
(15) ObjectId("54920fa744aef22d06ce05c8")	{ 3 fields }	Object
(16) ObjectId("54920fa744aef22d06ce05c8")	{ 3 fields }	Object
(17) ObjectId("54920fa744aef22d06ce05c8")	{ 3 fields }	Object
(18) ObjectId("54920fa744aef22d06ce05c8")	{ 3 fields }	Object
(19) ObjectId("54920fa744aef22d06ce05c8")	{ 3 fields }	Object
(20) ObjectId("54920fa744aef22d06ce05c8")	{ 3 fields }	Object
(21) ObjectId("54920fa744aef22d06ce05c8")	{ 3 fields }	Object
(22) ObjectId("54920fa744aef22d06ce05c8")	{ 3 fields }	Object
(23) ObjectId("54920fa744aef22d06ce05c8")	{ 3 fields }	Object

Logs

6. Query DB by keyword



Eclipse

Run Configurations

Create, manage, and run configurations

Run a Java application

Name: FileName

Main Arguments

Program arguments: umass

Variables...

VM arguments:

Variables...

Working directory:

Default: \${workspace_loc:GET_FILE_NAMES}

Other:

Workspace... File System... Variables...

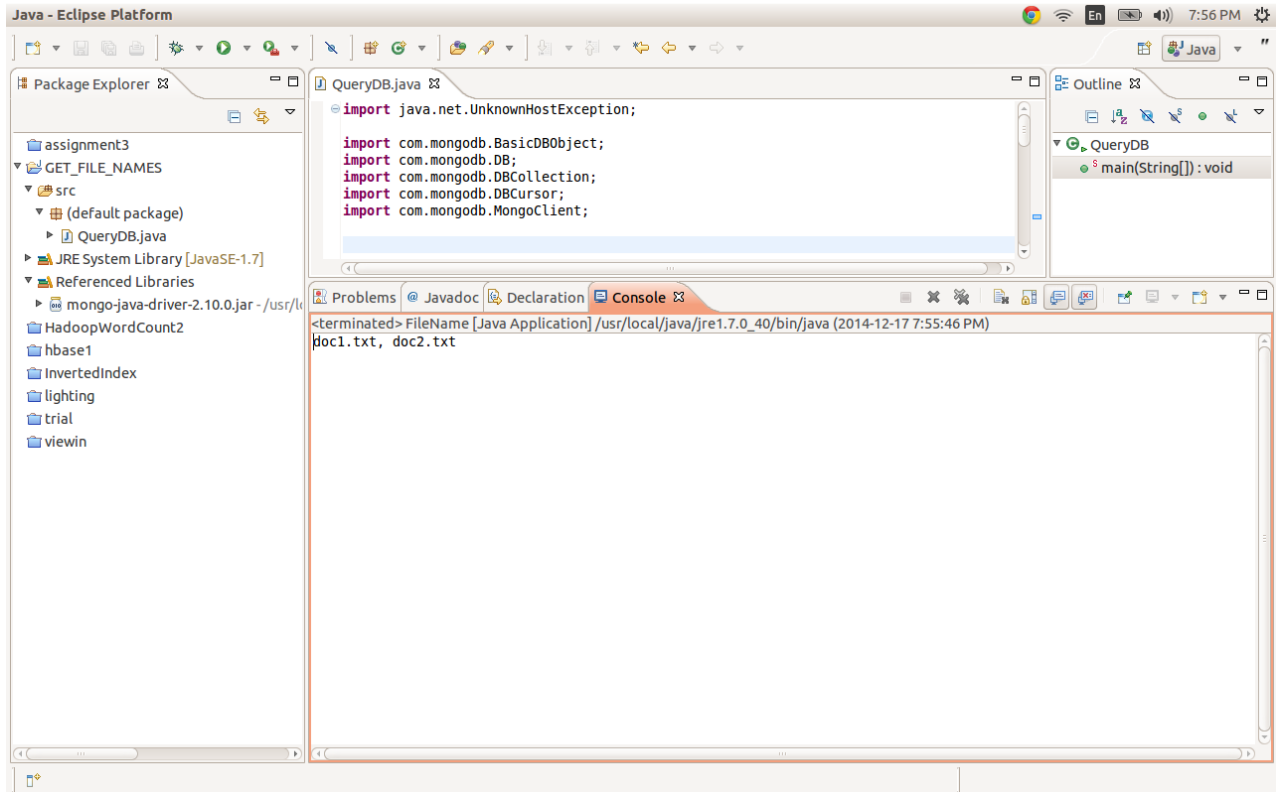
Apply Revert

Filter matched 11 of 25 items

Close Run

QueryDB.java - GET_FILE_NAMES/src

7. Result of above query



Problems faced -

- **Integration of HBase with Hadoop** - HBase worked fine from its shell and we could do CRUD operations there. But, could not get it running in Hadoop environment because of dependencies were not been resolved. We are still wondering and trying to figure out the reason because it worked fine with MongoDB.