NAME : KAUSHAL CHAUDHARY
NUID : 001886763

ASSIGNMENT 3

PART 1. Read the following paper (attached), and write a short summary/report.

The Google File System:

In this paper the creators present GFS, a conveyed document framework for enormous disseminated information escalated applications. GFS gives adaptation to internal failure while running on modest product equipment and it conveys high total execution to numerous customers. The creators initially portray its structure and its fundamental highlights and after that give trial results dependent on smaller scale benchmarks and genuine remaining tasks at hand. GFS shares a significant number of indistinguishable objectives from past circulated record frameworks, similar to execution and adaptability. Be that as it may, it is intended to function admirably on Google's remaining tasks at hand and condition. For instance, its essential objective is to oversee enormous multi-GB documents in a domain where equipment disappointments are normal. Other outstanding task at hand qualities that influenced its structure are the most successive activities (enormous consecutive composes and huge peruses).

One commitment of the framework is that customers never perused and compose information through the ace. They just ask the ace which chunkservers they should contact and after that store this data for quite a while. Thusly, better execution is accomplished as customer's solicitations are spread over a few chunkservers. Another commitment of the framework is its recuperation instrument. Activity logs and checkpoints are utilized to reestablish consistency when a disappointment happens. At long last, GFS bolsters a nuclear annex activity is made, with the goal that customers needn't bother with extra synchronization when attaching a record. One defect of the framework is that additional checks must be done to guarantee that the information is reliable. Each chunkserver utilizes checksumming to recognize defilement of put away information in light of the fact that the consistency model doesn't ensure indistinguishable imitations. These additional checks may diminish the presentation of the framework. Another blemish of the framework is that a chunkserver can wind up over-burden if a lump is gotten to by numerous customers. At last, it is intriguing to see an examination among GFS and other disseminated record frameworks for Google's remaining task at hand.

Mapreduce: Simplified Data Processing on Large Clusters:

The paper presents MapReduce, a programming model to take care of computational issues with huge datasets and yields that can be successfully parceled over their information. The paper acquaints us with Google's usage of MapReduce explicit to its execution condition of thousands of arranged ware frameworks. The issue handled by the paper is one of executing a programming model that robotizes to an enormous degree, the way toward apportioning an issue into sub-issues and appropriating the remaining burden on a few laborers and after that gathering the different sub-issues utilizing a client characterized capacity to produce the arrangement. What is normal to most issues that will profit by this programming model is the nearness of enormous datasets as information sources/yields or both.

The execution displayed in the paper is most appropriate for a situation where there are a large number of specialist hubs accessible with every hub being little (in calculation control) and the systems administration assets between hubs being a rare item. The usage considers these components and parts the info information into little lumps and regularly figures out how to limit the information (if not on a similar machine, the in a machine near it in the systems administration progression) that every specialist is alloted. This execution takes into account precisely 1 guide and 1 diminish stages yet as appeared by the different models, this is an enough model for some basic errands that would profit by it. The framework is worked to be profoundly issue tolerant since every datum piece is repeated up to multiple times and the ace monitors every one of the laborers that are going. At the point when specialist disappointment is recognized, the ace just doles out another laborer to the sub-issue that the bombed hub was relegated. Ace disappointments are treated as being

uncommon and as a rule brings about the whole issue being restarted in the present usage despite the fact that logging and registration like in disseminated database frameworks is a choice.

BigTable: A Distributed Storage System for Structured Data:

Bigtable is an adaptable, elite dispersed organized information stockpiling answer for both mass handling and ongoing information serving prerequisites broadly utilized by Google web ordering, Google Earth, and Google Finance.

Thoughtfully, A Bigtable is an inadequate, dispersed, relentless multi-dimensional arranged guide. It's ordered by a line key, segment key and a timestamp; each incentive in the guide is a uninterpreted cluster of bytes. Physically, it comprises of a library that is connected into each customer, an ace server and numerous tablet servers. It's based over Google File System and works in a common pool of machines that run a wide assortment of other appropriated applications. Google SSTable is utilized to store Bigtable information, which gives elite query and can be mapped into memory to discard additional circle query. Bigtable additionally depends intensely on a profoundly accessible and industrious conveyed lock administration called Chubby which uses Paxos calculation to look after consistency. Tubby customers utilizes stateful session to speak with Chubby help. Tablets are the occasions that store the organized information. Bigtable uses a three level area chain of importance which is equipped for putting away 2^32 tablet areas. So as to offload the single ace, most customer doesn't have to speak with the ace, they store the areas of tablet servers and do energetic pre-bring to get lower dormancy. One exchange off here is the unpredictability of the customer library – these library ought to be more confused than a database library of Oracle or PostgresSQL. Be that as it may, this structure nimbly avoided the need of a brought together ace server as a ton of dispersed answers for databases like MongoDB and PostgreSQL do, accordingly significantly improves the degree of versatility.

Updates of a tablet is first dedicated to a submit log that store re-try records. Latest submitted updates are put away in memtable, more established ones are put away in an arrangement of SSTables. At the point when the memtable develops into a specific size, it will be compacted into SSTable. By utilizing this method, Bigtable can right off the bat contract the memory use of the tablet server and furthermore lessen the measure of information that must be perused from the submit log during recuperation if the server passes on. A significant compaction is booked routinely to deliver SSTable that contains no erasure data or erased information.

Bigtable uses two-level reserving and sprout channels to improve read execution. Sweep Cache stores key-esteem sets returned by SSTable interface, and Block Cache reserves results came back from GFS. Sprout channels can lessen the gathering of servers that a read activity need to contact along these lines diminish the quantity of circle gets to.

The Chubby lock service for loosely-coupled distributed systems:

This paper clarifies how Chubby functions. Tubby is a disseminated lock administration planned by and utilized at Google. It gives a disseminated filesystem that is streamlined for little records and uncommon composes. Since it actualizes warning document/index locks, customers can utilize it as a lock administration, yet they can likewise utilize it as a name administration and, as per the paper, the last has turned into Chubby's essential use at Google.

A Chubby cell comprises of five reproduction servers, one of which is chosen as an ace. A given cell has territory over a subtree of the worldwide Chubby namespace. Each filename in Chubby starts with/ls/. The ace serves all customer demands, the majority of which are KeepAlive messages for sessions. Customers normally open a session and utilize these messages to keep the session alive. Customers can enlist for different occasions including record alteration occasions. They utilize a compose through reserve where the server can discredit store things for singular customers as required. On the off chance that the ace comes up short, Chubby uses a come up short over system to choose another ace and move customers over to it as quickly as time permits. Rotund incorporates a system to minimalistically move sessions over to the new ace and educate customers regarding the fizzle over with the goal that they can refute their own reserves and advise the application that occasions may have been missed.

NAME : KAUSHAL CHAUDHARY
NUID : 001886763

One blemish with this paper is that it doesn't solidly build up the adaptability of Chubby. In spite of the fact that it appears to be evident that Chubby's adaptability is sufficient for its present uses, the paper displays no observational proof that it could scale further. It shows a few strategies for scaling, some of which have been utilized underway (expanding lease spans, including Chubby cells), and others of which have not (intermediaries, apportioning). As to last strategies, it isn't obvious to me how intermediaries could deal with KeepAlive and read demands without server contribution. Further, dividing the namespace of a Chubby cell so that subtrees of the namespace would have various experts appears to be not exactly perfect as it would require application designers to physically parcel and would give no real way to the Chubby cell to deal with burden adjusting among the allotments. Given that the paper as of now subtleties a few different ways that absence of instruction about Chubby has been an issue, it appears that expecting engineers to brilliantly structure a namespace to exploit apportioning would be troublesome.

Another blemish is that the paper doesn't present especially quantitative proof. A paper like this calls for heaps of diagrams indicating execution attributes for various use situations. It additionally calls for hard numbers. Rather the main outline we get is a table giving a preview of a Chubby cell with an affirmation that the numbers are "regular" for Google. The creators additionally, abnormally, don't give us hard accessibility measurements, however they do disclose to us that they recorded 61 blackouts "over a time of half a month".

PART 2 – Programming Assignment

All hadoop commands are invoked by the bin/hadoop script. Running the hadoop script without any arguments prints the description for all commands.

Usage: hadoop [--config confdir] [--loglevel loglevel] [COMMAND] [GENERIC_OPTIONS] [COMMAND_OPTIONS]

Execute each hadoop command once, and place the screenshots into a word file. If a command cannot be executed for any reason (such as, a distributed environment is needed), you may write the definition of the command, and skip execution.

http://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-common/FileSystemShell.html

- cat

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -cat file:///home/
kchaudhary/Downloads/GitRepos.txt
https://github.com/pawarad
https://github.com/keiraqz/artmosphere
https://github.com/ranga11
https://github.com/PreetikaKuls/Insight-MapMyCab
https://github.com/jgors/anywazekchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin
$
```

- checksum

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -checksum file:///
home/kchaudhary/Downloads/GitRepos.txt
file:///home/kchaudhary/Downloads/GitRepos.txt  NONE
```

- chgrp

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -chgrp -R hdfs /te
stdir/test1
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -ls /testdir
Found 2 items
drwxr-xr-x   - kchaudhary hdfs                0 2019-10-10 16:40 /testdir/test1
-rwxrwxrwx   1 hdfs       supergroup          0 2019-10-10 16:05 /testdir/testfil
e
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$
```

- chmod

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -chmod -R 777 /tes
tdir
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -ls /
Found 2 items
drwxrwxrwx   - kchaudhary supergroup          0 2019-10-10 16:05 /testdir
drwxr-xr-x   - kchaudhary supergroup          0 2019-10-10 16:03 /testdir1
```

- chown

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -chown -R hdfs /te
stdir
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -ls /
Found 2 items
drwxrwxrwx   - hdfs       supergroup          0 2019-10-10 16:05 /testdir
drwxr-xr-x   - kchaudhary supergroup          0 2019-10-10 16:03 /testdir1
```

- copyToLocal

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -copyToLocal /test
dir  file:///home/kchaudhary/Downloads
```

- count

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -count /testdir
           1            1                  0 /testdir
```

- df

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -df
Filesystem             Size  Used  Available  Use%
hdfs://localhost:9000     0     0          0  NaN%
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$
```

- du

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -du /
0  0  /testdir
0  0  /testdir1
```

- find

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -find / -name test
1 -print
/testdir/test1
```

- getfacl

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -getfacl /testdir/
test1
# file: /testdir/test1
# owner: kchaudhary
# group: supergroup
user::rwx
group::r-x
other::r-x
```

- getfattr

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -getfattr -d /test
dir/test1
# file: /testdir/test1
```

- getmerge

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -getmerge -nl /tes
tdir/test1 file://home/kchaudhary/Downloads/test2
getmerge: Mkdirs failed to create file:/kchaudhary/Downloads (exists=false, cwd=f
ile:/usr/local/bin/hadoop-3.2.1/bin)
```

- help

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -help
Usage: hadoop fs [generic options]
        [-appendToFile <localsrc> ... <dst>]
        [-cat [-ignoreCrc] <src> ...]
        [-checksum <src> ...]
        [-chgrp [-R] GROUP PATH...]
        [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
        [-chown [-R] [OWNER][:[GROUP]] PATH...]
        [-copyFromLocal [-f] [-p] [-l] [-d] [-t <thread count>] <localsrc> ... <d
st>]
        [-copyToLocal [-f] [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]
        [-count [-q] [-h] [-v] [-t [<storage type>]] [-u] [-x] [-e] <path> ...]
        [-cp [-f] [-p | -p[topax]] [-d] <src> ... <dst>]
        [-createSnapshot <snapshotDir> [<snapshotName>]]
        [-deleteSnapshot <snapshotDir> <snapshotName>]
        [-df [-h] [<path> ...]]
        [-du [-s] [-h] [-v] [-x] <path> ...]
        [-expunge [-immediate]]
        [-find <path> ... <expression> ...]
        [-get [-f] [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]
```

- ls

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -ls /
Found 1 items
drwxr-xr-x   - kchaudhary supergroup          0 2019-10-07 15:41 /testdir
```

- lsr

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -lsr /
lsr: DEPRECATED: Please use 'ls -R' instead.
drwxrwxrwx   - hdfs       supergroup          0 2019-10-10 16:48 /testdir
drwxr-xr-x   - kchaudhary hdfs                0 2019-10-10 16:40 /testdir/test1
-rwxrwxrwx   1 hdfs       supergroup          0 2019-10-10 16:05 /testdir/testfil
e
drwxr-xr-x   - kchaudhary supergroup          0 2019-10-10 16:29 /testdir1
```

- mkdir

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -ls /
Found 2 items
drwxr-xr-x   - kchaudhary supergroup          0 2019-10-07 15:41 /testdir
drwxr-xr-x   - kchaudhary supergroup          0 2019-10-10 16:03 /testdir1
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$
```

- moveFromLocal

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -moveFromLocal fil
e:///home/kchaudhary/Downloads/test /testdir
moveFromLocal: `/home/kchaudhary/Downloads/test': No such file or directory
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -moveFromLocal fil
e:///home/kchaudhary/Downloads/test1 /testdir
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -ls /testdir
Found 2 items
drwxr-xr-x   - kchaudhary supergroup          0 2019-10-10 16:40 /testdir/test1
-rwxrwxrwx   1 hdfs       supergroup          0 2019-10-10 16:05 /testdir/testfil
e
```

- rm

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -rm -f /testdir/te
st1
rm: `/testdir/test1': Is a directory
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$
```

- rmdir

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -rmdir /testdir/te
st1
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -ls /testdir
Found 1 items
-rwxrwxrwx   1 hdfs supergroup          0 2019-10-10 16:05 /testdir/testfile
```

- stat

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -stat /testdir
2019-10-10 23:05:04
```

- tail

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -tail file:///home
/kchaudhary/Downloads/GitRepos.txt
https://github.com/pawarad
https://github.com/keiraqz/artmosphere
https://github.com/ranga11
https://github.com/PreetikaKuls/Insight-MapMyCab
https://github.com/jgors/anywazekchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin
$
```

- text

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -text file:///home
/kchaudhary/Downloads/GitRepos.txt
https://github.com/pawarad
https://github.com/keiraqz/artmosphere
https://github.com/ranga11
https://github.com/PreetikaKuls/Insight-MapMyCab
https://github.com/jgors/anywazekchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin
$
```

- touchz

```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -touchz /testdir/t
estfile
```

- usage

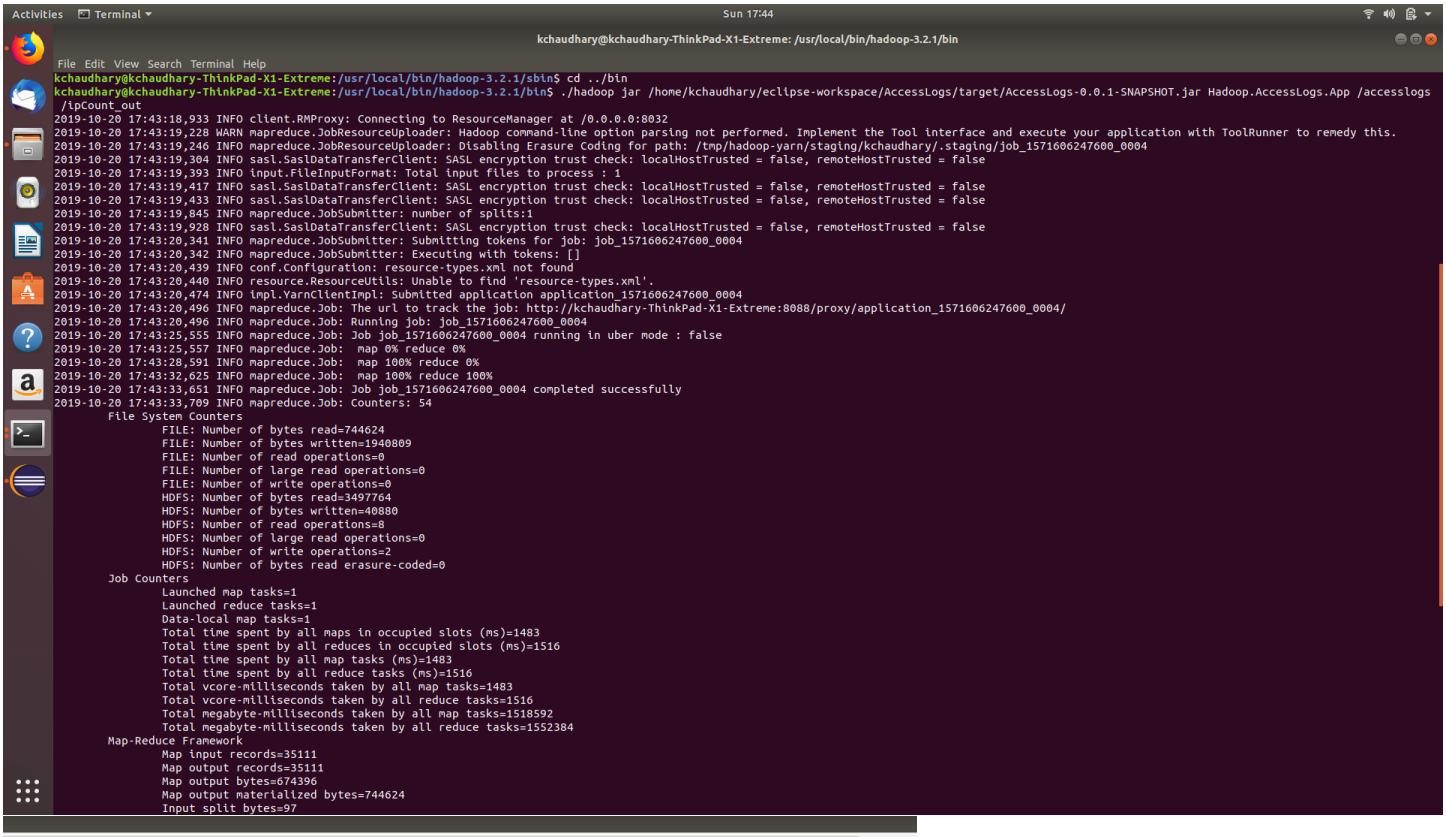
```
kchaudhary@ubuntu:/usr/local/bin/hadoop-3.2.1/bin$ ./hadoop fs -usage
Usage: hadoop fs [generic options]
        [-appendToFile <localsrc> ... <dst>]
        [-cat [-ignoreCrc] <src> ...]
        [-checksum <src> ...]
        [-chgrp [-R] GROUP PATH...]
        [-chmod [-R] <MODE[,MODE]... | OCTALMODE> PATH...]
        [-chown [-R] [OWNER][:[GROUP]] PATH...]
        [-copyFromLocal [-f] [-p] [-l] [-d] [-t <thread count>] <localsrc> ...
st>]
        [-copyToLocal [-f] [-p] [-ignoreCrc] [-crc] <src> ... <localdst>]
        [-count [-q] [-h] [-v] [-t [<storage type>]] [-u] [-x] [-e] <path> ...]
        [-cp [-f] [-p | -p[topax]] [-d] <src> ... <dst>]
        [-createSnapshot <snapshotDir> [<snapshotName>]]
        [-deleteSnapshot <snapshotDir> <snapshotName>]
        [-df [-h] [<path> ...]]
```

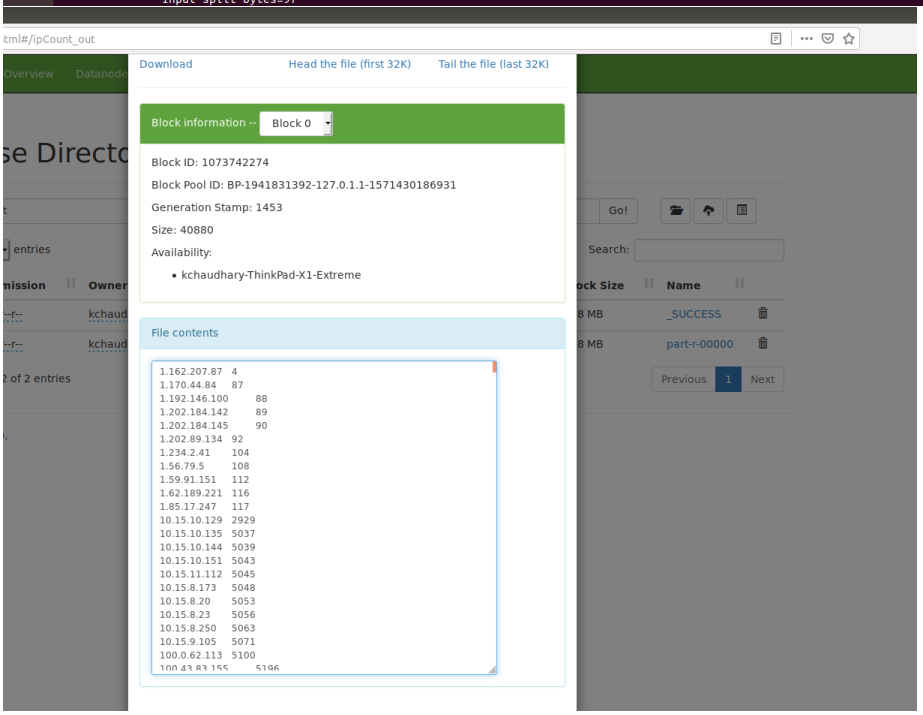
NAME : KAUSHAL CHAUDHARY
NUID : 001886763

PART 3 – Programming Assignment

Copy the attached 'access.log' file into HDFS under /logs directory.

Using the access.log file stored in HDFS, implement MapReduce in Hadoop to find the number of times each IP accessed the website.



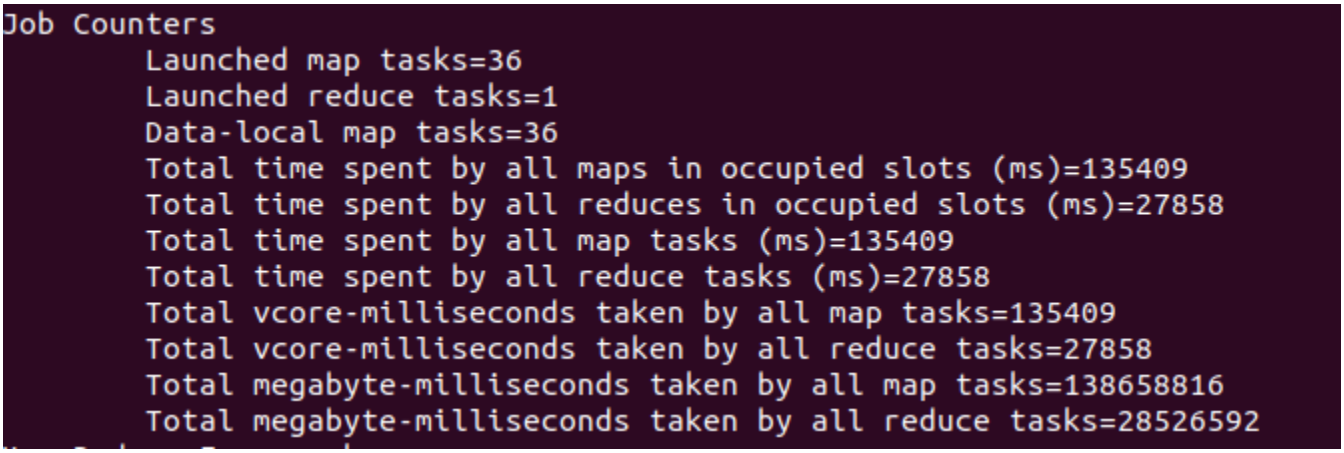


NAME : KAUSHAL CHAUDHARY
NUID : 001886763

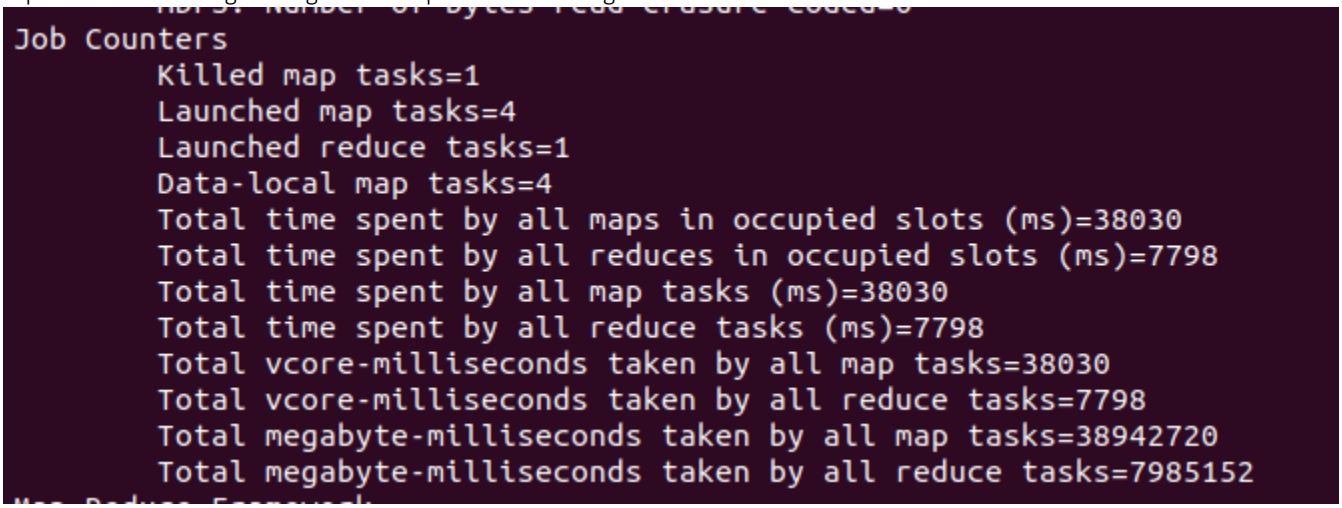
PART 4 – Programming Assignment

Download and Copy all the files (http://msis.neu.edu/nyse/) (DailyPrices_A to DailyPrices_Z) to a folder in HDFS.

PART 4.1 – Write a MapReduce to find the Max price of stock_price_high for each stock. Capture the running time programmatically (or manually using a wristwatch or smartphone).



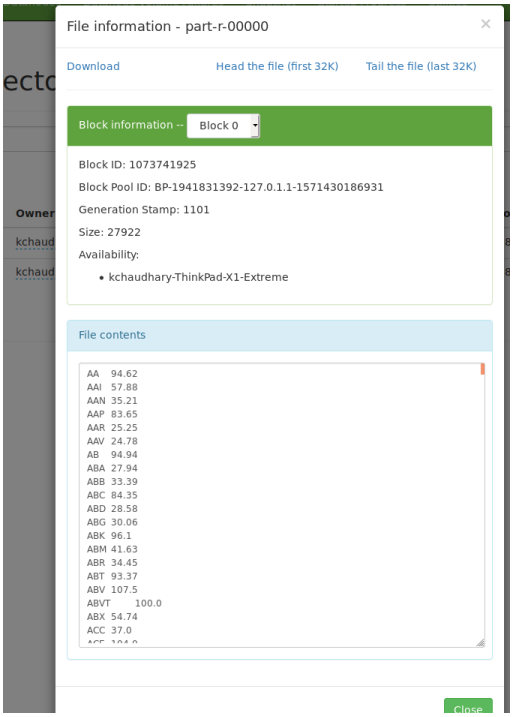
PART 4.2 – Write a Java Program to implement PutMerge as discussed in the class to merge the NYSE files in a single file on HDFS. Now, repeat 4.1 on the single merged-file. Capture the running time.



Did MapReduce on a single file run faster than running MapReduce on a bunch of files?

Answer : The two screenshots above shows that the MapReduce performed well after PutMerge. Since there were multiple files present in the HDFS before putMerge, the mappers were taking time to read each file therefore the processing time was much larger than the later one. Therefore, the MapReduce program performed better after the PutMerge method.

NAME : KAUSHAL CHAUDHARY
NUID : 001886763



NAME : KAUSHAL CHAUDHARY
NUID : 001886763

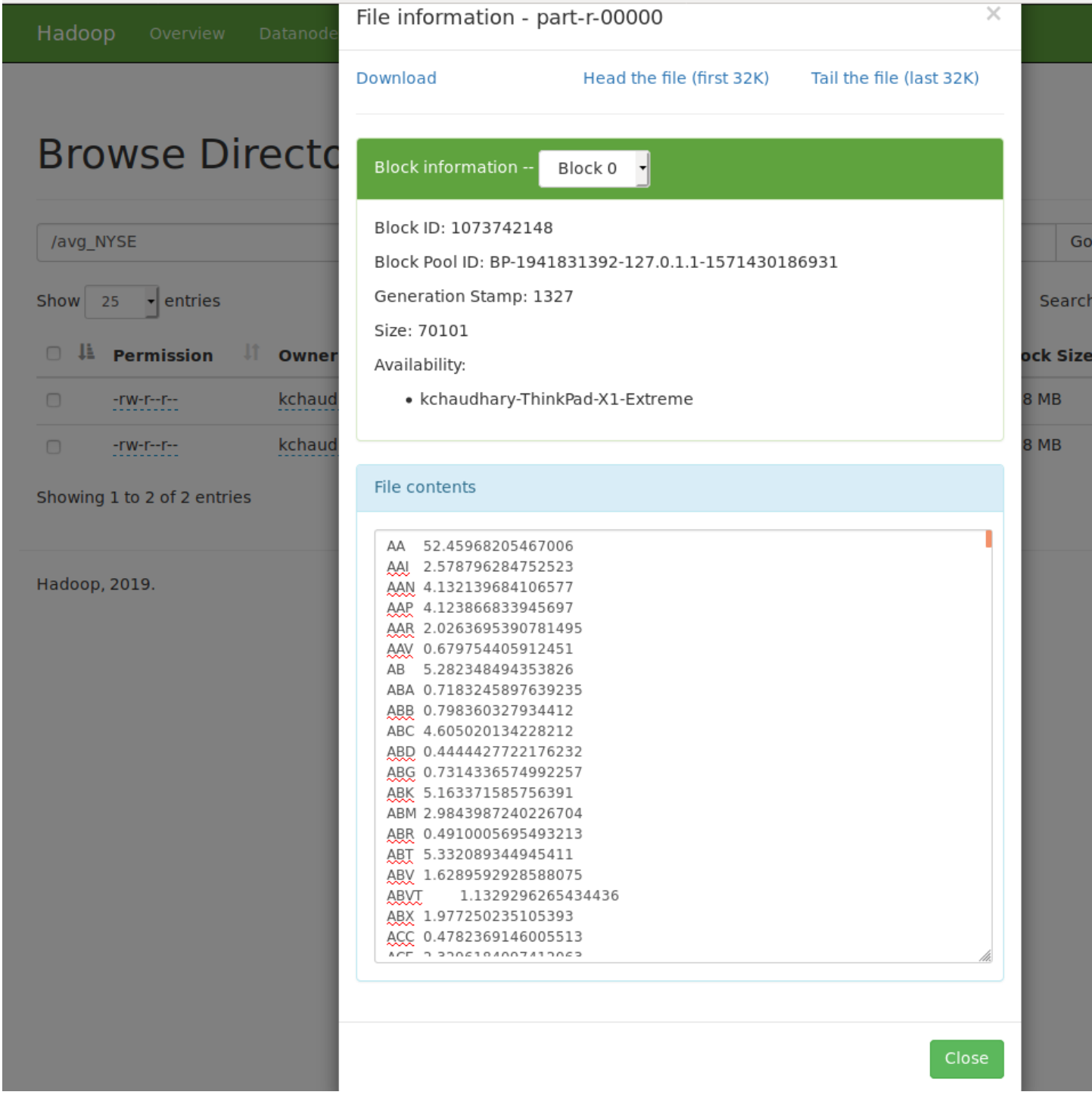
PART 5 – Programming Assignment

Write one MapReduce program using each of the classes that extend FileInputFormat<k,v>

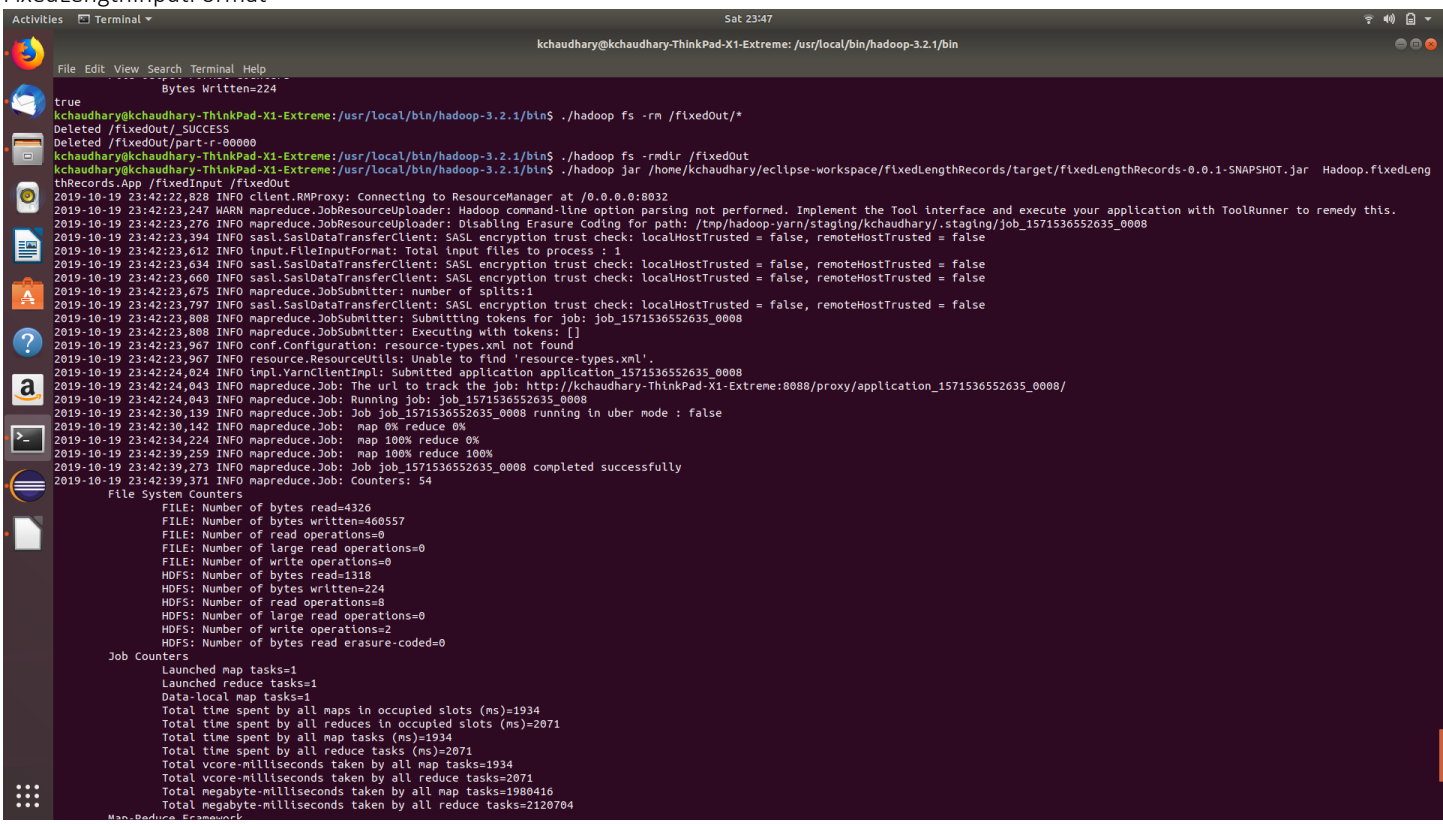
1. CombineFileInputForm

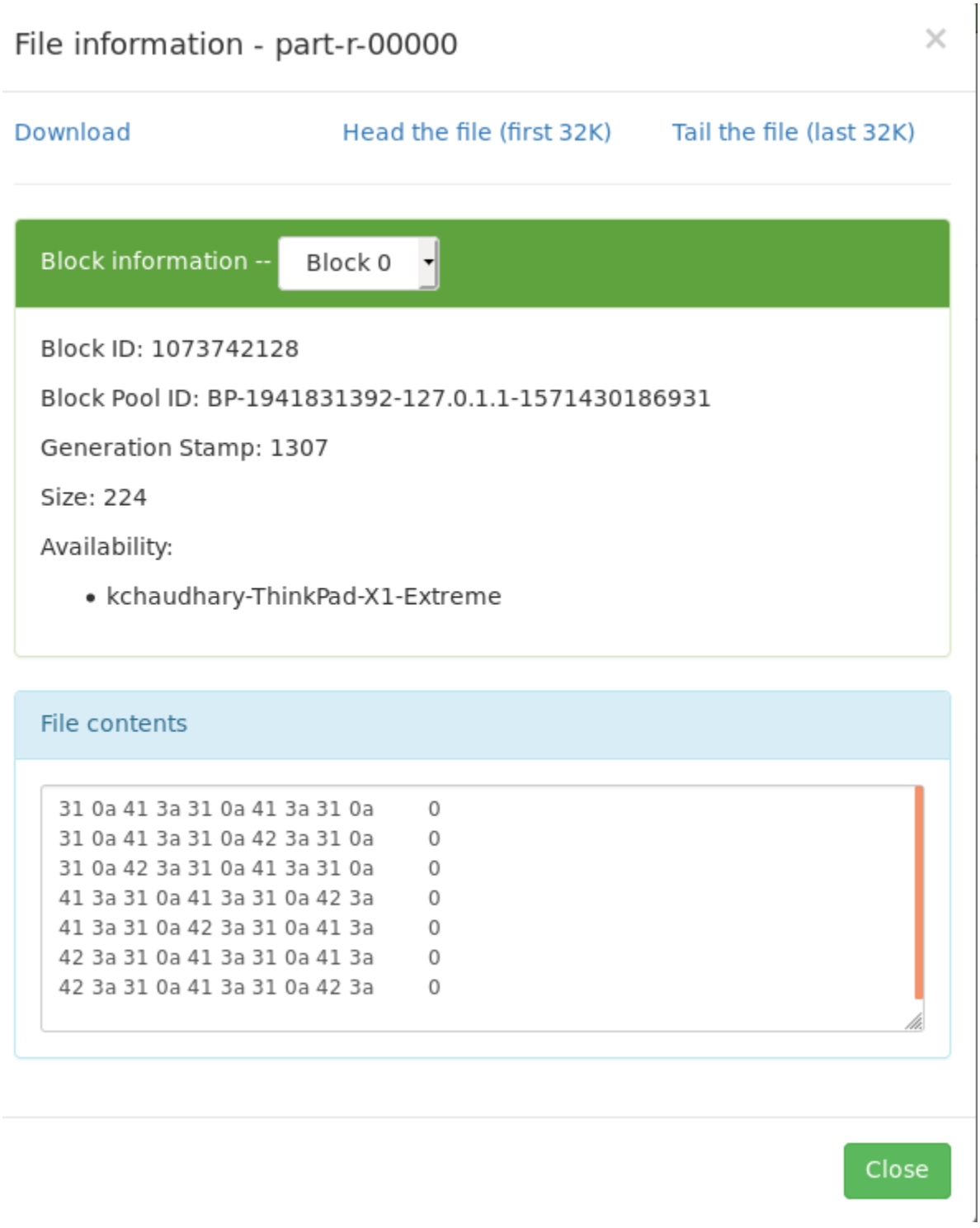
```
File System Counters
        FILE: Number of bytes read=256346724
        FILE: Number of bytes written=384971340
        FILE: Number of read operations=0
        FILE: Number of large read operations=0
        FILE: Number of write operations=0
        HDFS: Number of bytes read=511089414
        HDFS: Number of bytes written=70101
        HDFS: Number of read operations=43
        HDFS: Number of large read operations=0
        HDFS: Number of write operations=2
        HDFS: Number of bytes read erasure-coded=0
Job Counters
        Launched map tasks=1
        Launched reduce tasks=1
        Other local map tasks=1
        Total time spent by all maps in occupied slots (ms)=16831
        Total time spent by all reduces in occupied slots (ms)=6689
        Total time spent by all map tasks (ms)=16831
        Total time spent by all reduce tasks (ms)=6689
        Total vcore-milliseconds taken by all map tasks=16831
        Total vcore-milliseconds taken by all reduce tasks=6689
        Total megabyte-milliseconds taken by all map tasks=17234944
        Total megabyte-milliseconds taken by all reduce tasks=6849536
Map-Reduce Framework
        Map input records=9211067
        Map output records=9211031
        Map output bytes=109751285
        Map output materialized bytes=128173353
        Input split bytes=2487
        Combine input records=0
        Combine output records=0
        Reduce input groups=2853
        Reduce shuffle bytes=128173353
        Reduce input records=9211031
        Reduce output records=2853
        Spilled Records=27633093
        Shuffled Maps =1
        Failed Shuffles=0
        Merged Map outputs=1
        GC time elapsed (ms)=211
        CPU time spent (ms)=26690
        Physical memory (bytes) snapshot=892407808
        Virtual memory (bytes) snapshot=5339172864
        Total committed heap usage (bytes)=911736832
        Peak Map Physical memory (bytes)=504942592
        Peak Map Virtual memory (bytes)=2664935424
        Peak Reduce Physical memory (bytes)=387465216
        Peak Reduce Virtual memory (bytes)=2674237440
```

at

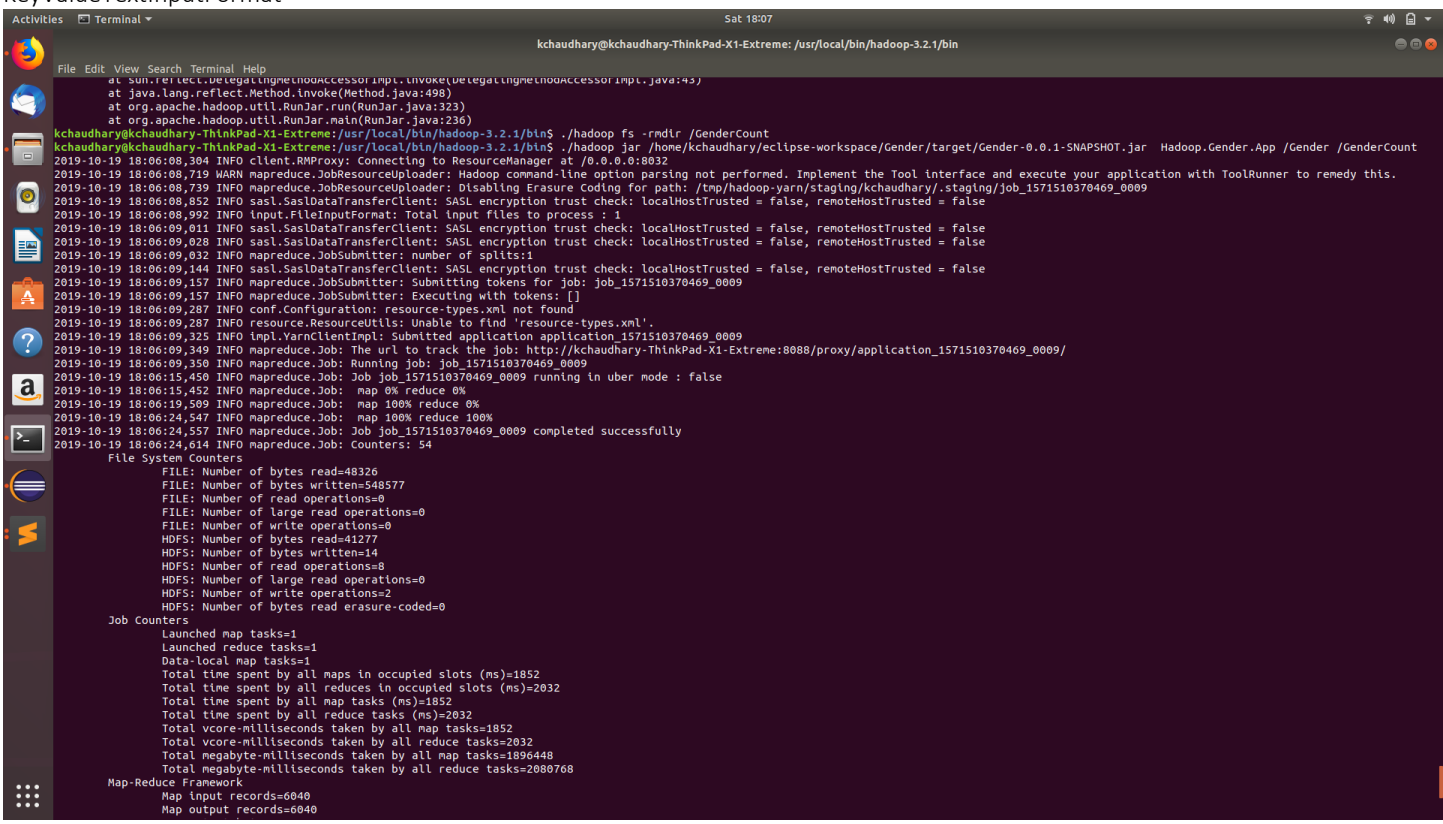

Hadoop
Overview
Datanode
Browse Directo
/avg_NYSE
Show 25 entries
Permission
Owner
-rw-r--r--
kchaud
-rw-r--r--
kchaud
Showing 1 to 2 of 2 entries
Hadoop, 2019.
File information - part-r-00000
Download
Head the file (first 32K)
Tail the file (last 32K)
Block information -- Block 0
Block ID: 1073742148
Block Pool ID: BP-1941831392-127.0.1.1-1571430186931
Generation Stamp: 1327
Size: 70101
Availability:
kchaudhary-ThinkPad-X1-Extreme
File contents
AA 52.45968205467006
AAI 2.578796284752523
AAN 4.132139684106577
AAP 4.123866833945697
AAR 2.0263695390781495
AAV 0.679754405912451
AB 5.282348494353826
ABA 0.7183245897639235
ABB 0.798360327934412
ABC 4.605020134228212
ABD 0.4444427722176232
ABG 0.7314336574992257
ABK 5.163371585756391
ABM 2.9843987240226704
ABR 0.4910005695493213
ABT 5.332089344945411
ABV 1.6289592928588075
ABVT 1.1329296265434436
ABX 1.977250235105393
ACC 0.4782369146005513
Close

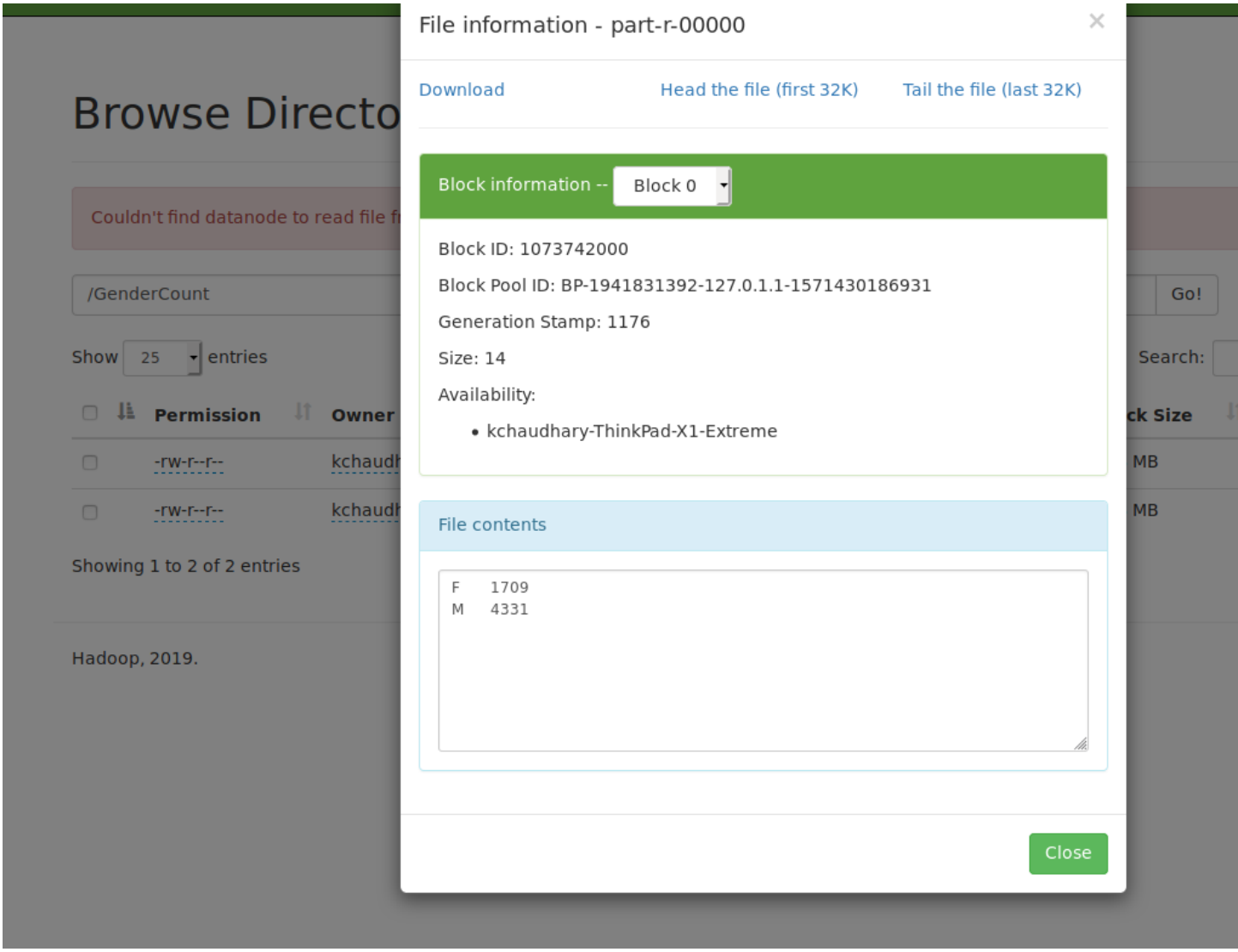
2. FixedLengthInputFormat



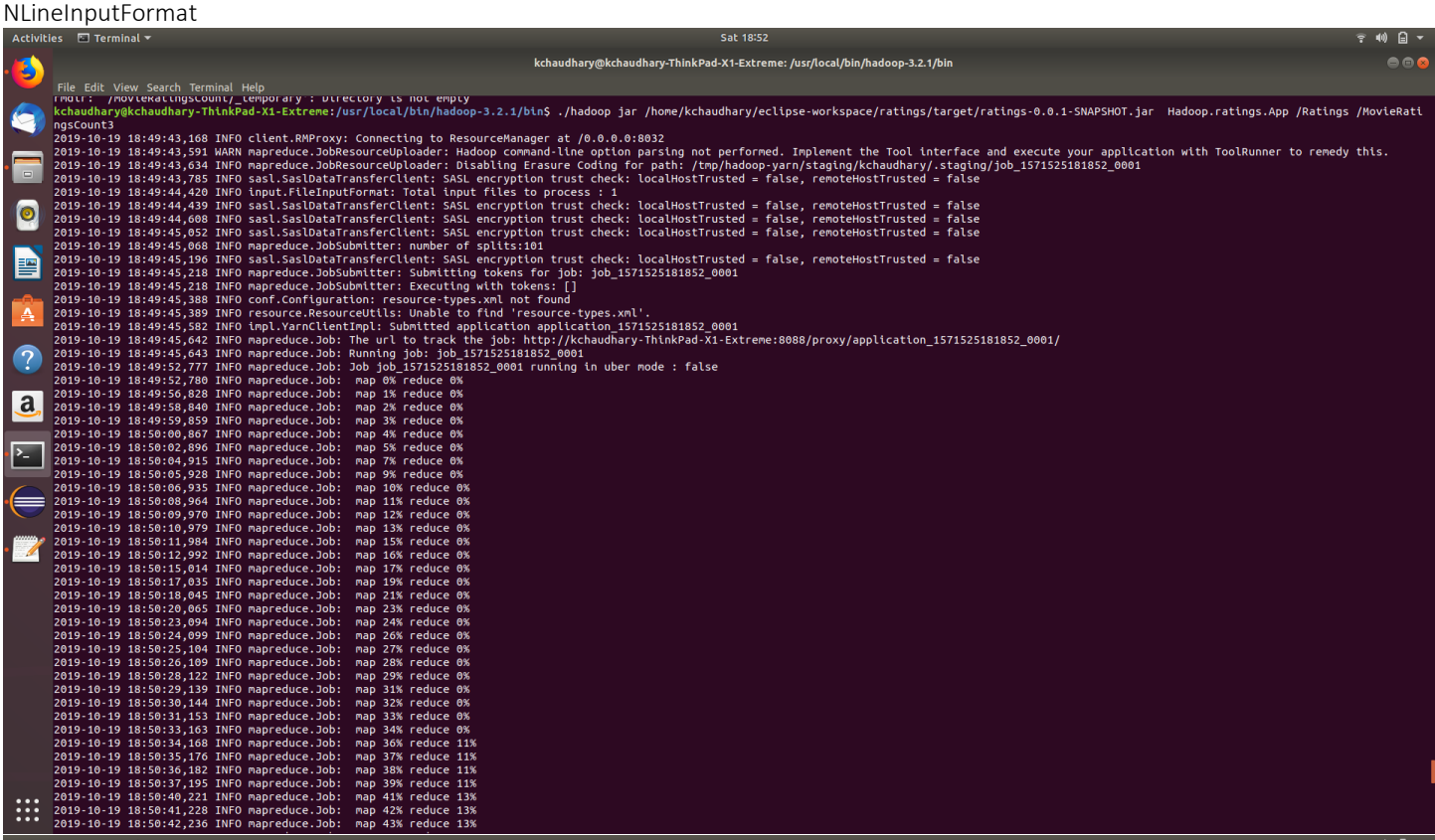

File information - part-r-00000
Download
Head the file (first 32K)
Tail the file (last 32K)
Block information -- Block 0
Block ID: 1073742128
Block Pool ID: BP-1941831392-127.0.1.1-1571430186931
Generation Stamp: 1307
Size: 224
Availability:
kchaudhary-ThinkPad-X1-Extreme
File contents
31 0a 41 3a 31 0a 41 3a 31 0a 0
31 0a 41 3a 31 0a 42 3a 31 0a 0
31 0a 42 3a 31 0a 41 3a 31 0a 0
41 3a 31 0a 41 3a 31 0a 42 3a 0
41 3a 31 0a 42 3a 31 0a 41 3a 0
42 3a 31 0a 41 3a 31 0a 41 3a 0
42 3a 31 0a 41 3a 31 0a 42 3a 0
Close

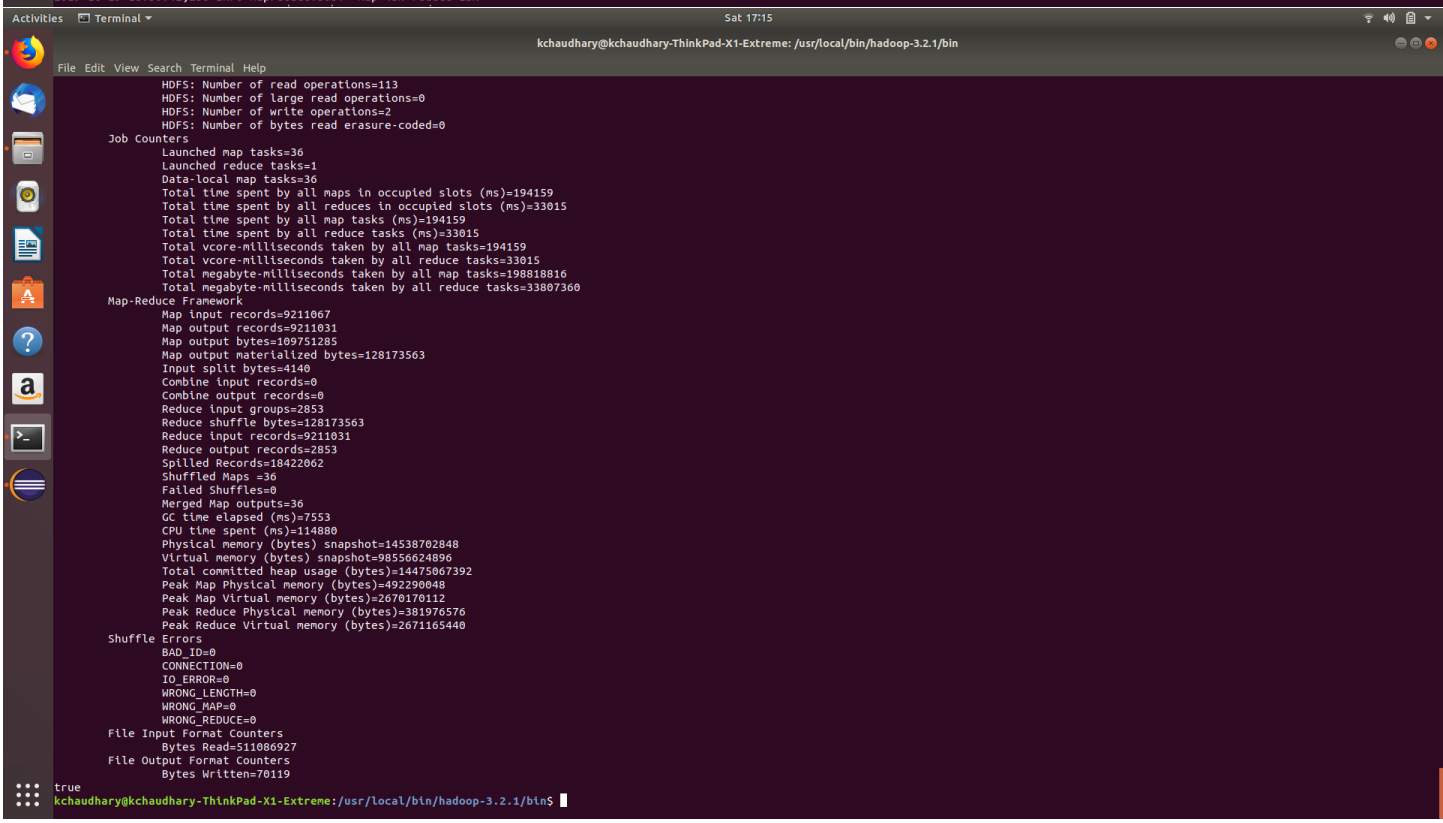
3. KeyValueTextInputFormat



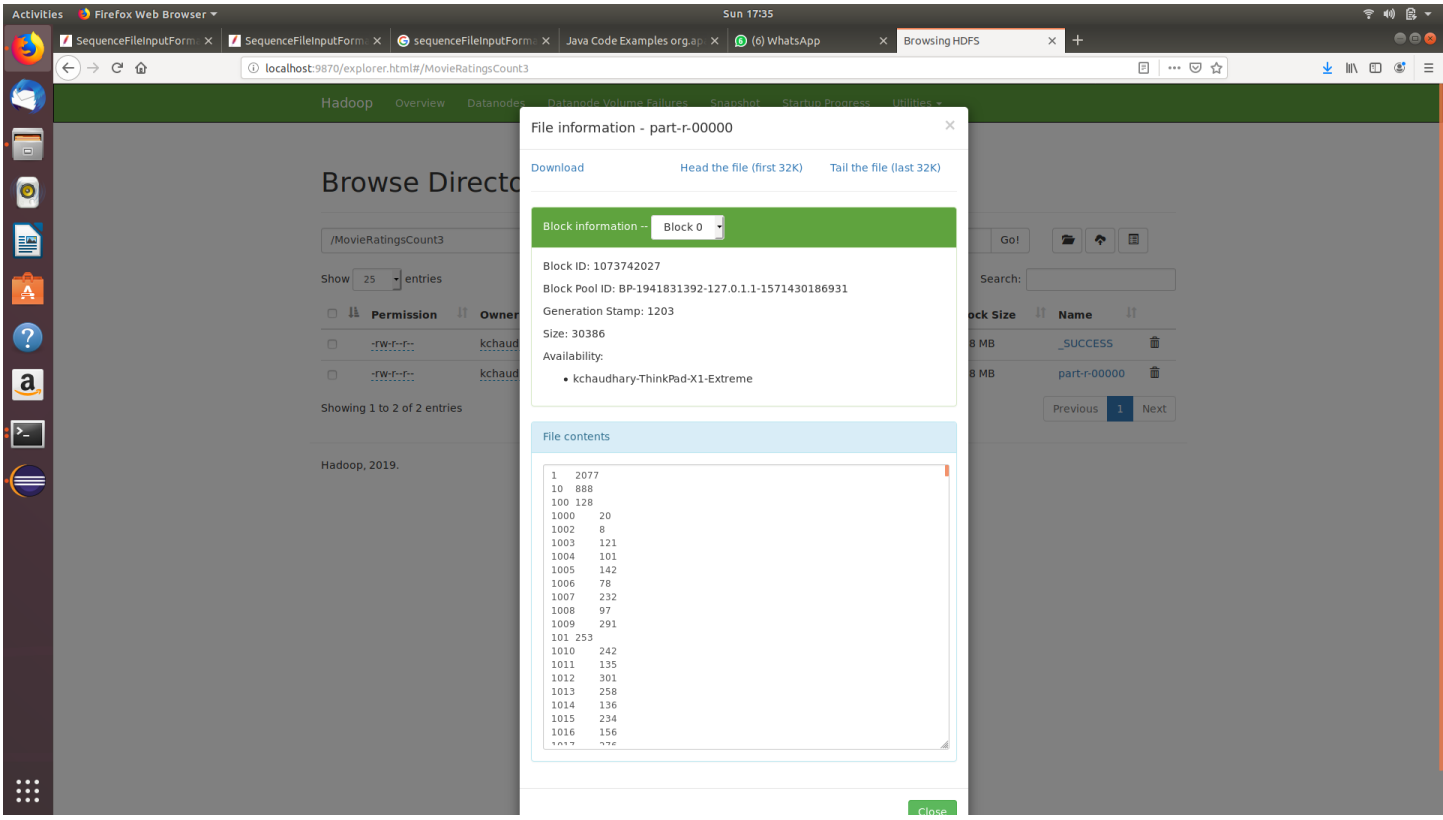

Browse Directo
Couldn't find datanode to read file fr
/GenderCount
Go!
Show 25 entries
Search:
Permission
Owner
-rw-r--r--
kchaudh
-rw-r--r--
kchaudh
MB
MB
Showing 1 to 2 of 2 entries
Hadoop, 2019.
File information - part-r-00000
Download
Head the file (first 32K)
Tail the file (last 32K)
Block information -- Block 0
Block ID: 1073742000
Block Pool ID: BP-1941831392-127.0.1.1-1571430186931
Generation Stamp: 1176
Size: 14
Availability:
kchaudhary-ThinkPad-X1-Extreme
File contents
F 1709
M 4331
Close

4. NLineInputFormat

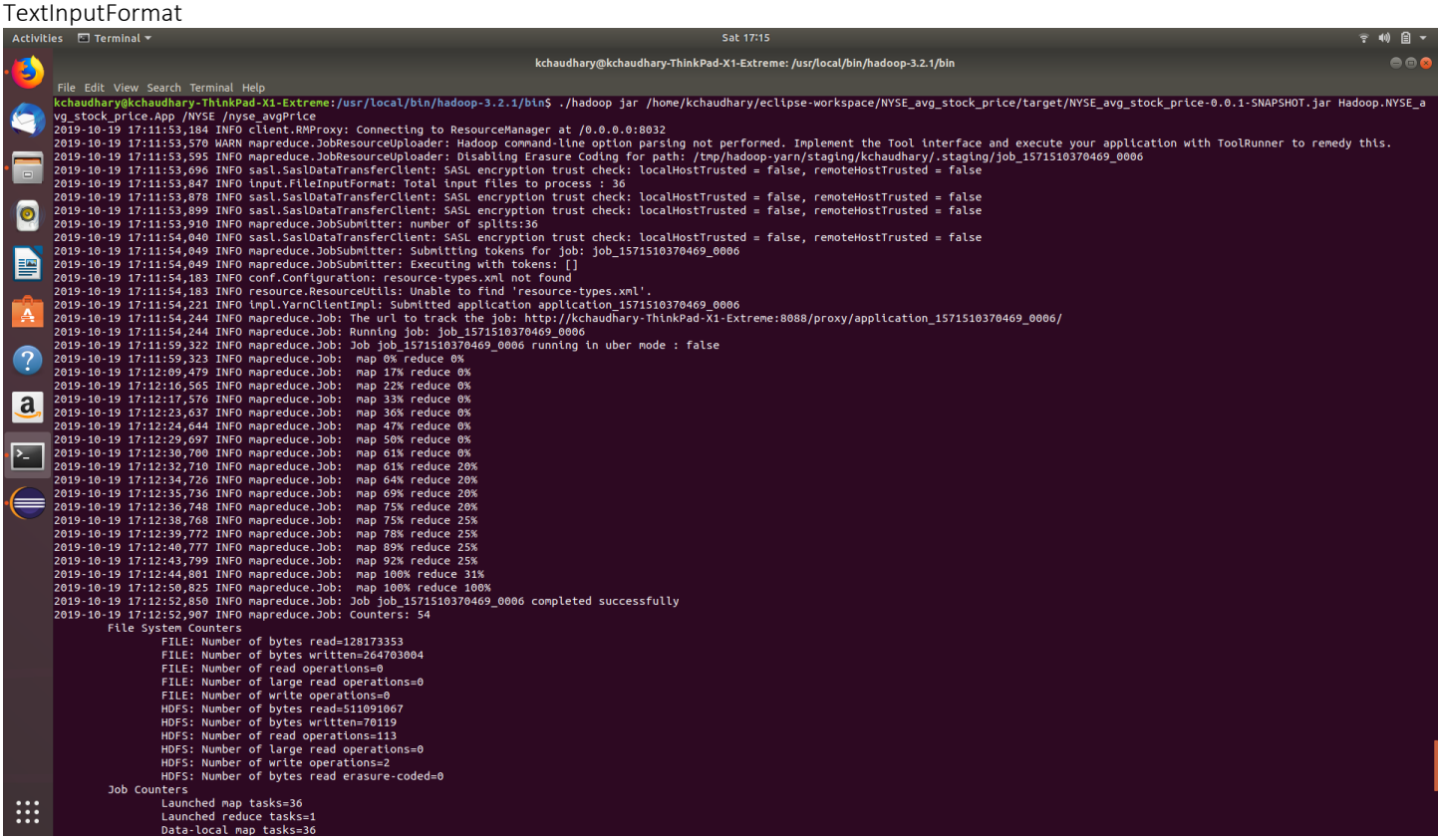




NAME : KAUSHAL CHAUDHARY
NUID : 001886763



5. TextInputFormat



File information - part-r-00000

Download Head the file (first 32K) Tail the file (last 32K)

Block information -- Block 0

Block ID: 1073741899

Block Pool ID: BP-1941831392-127.0.1.1-1571430186931

Generation Stamp: 1075

Size: 27922

Availability:

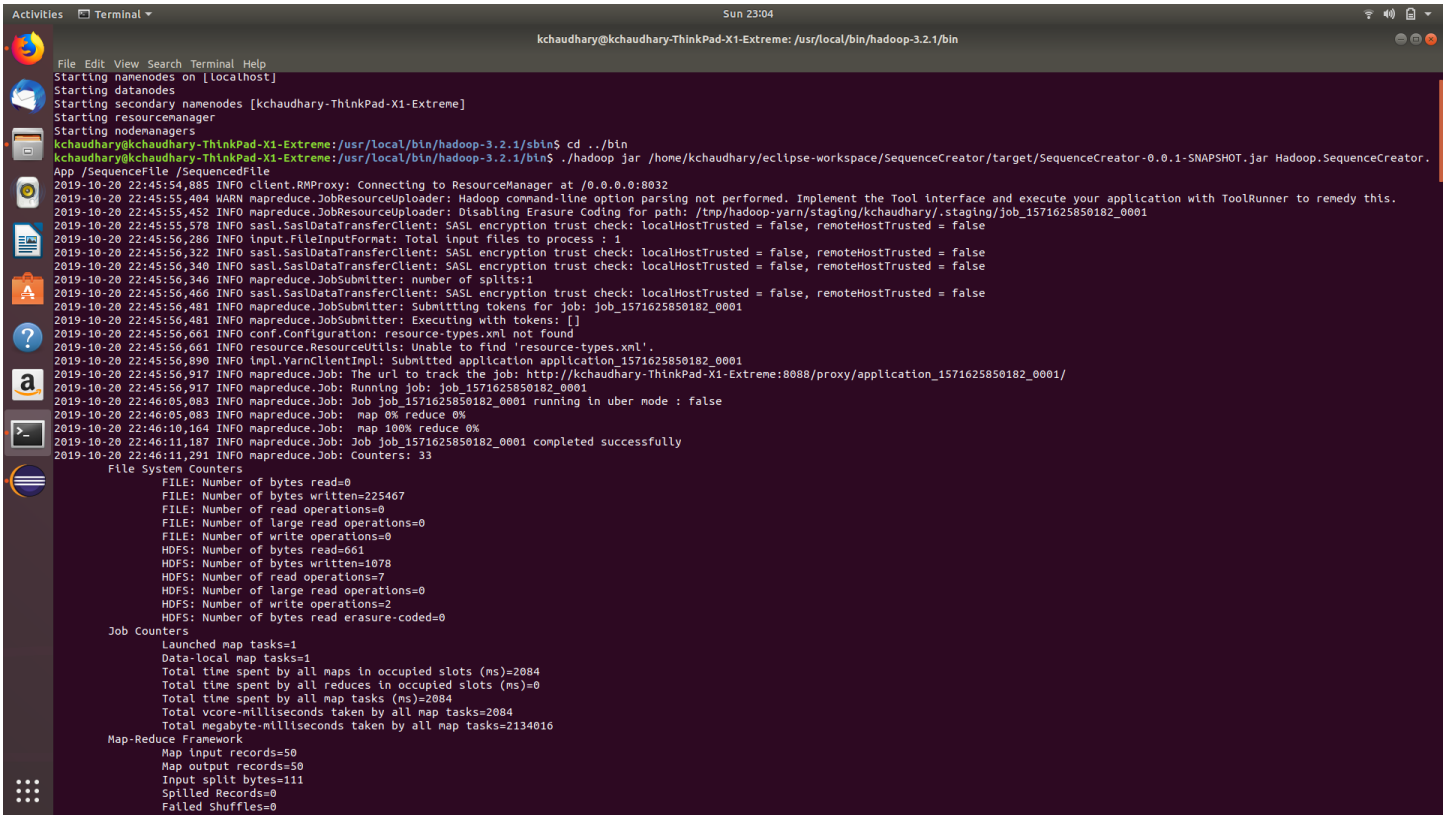
- kchaudhary-ThinkPad-X1-Extreme

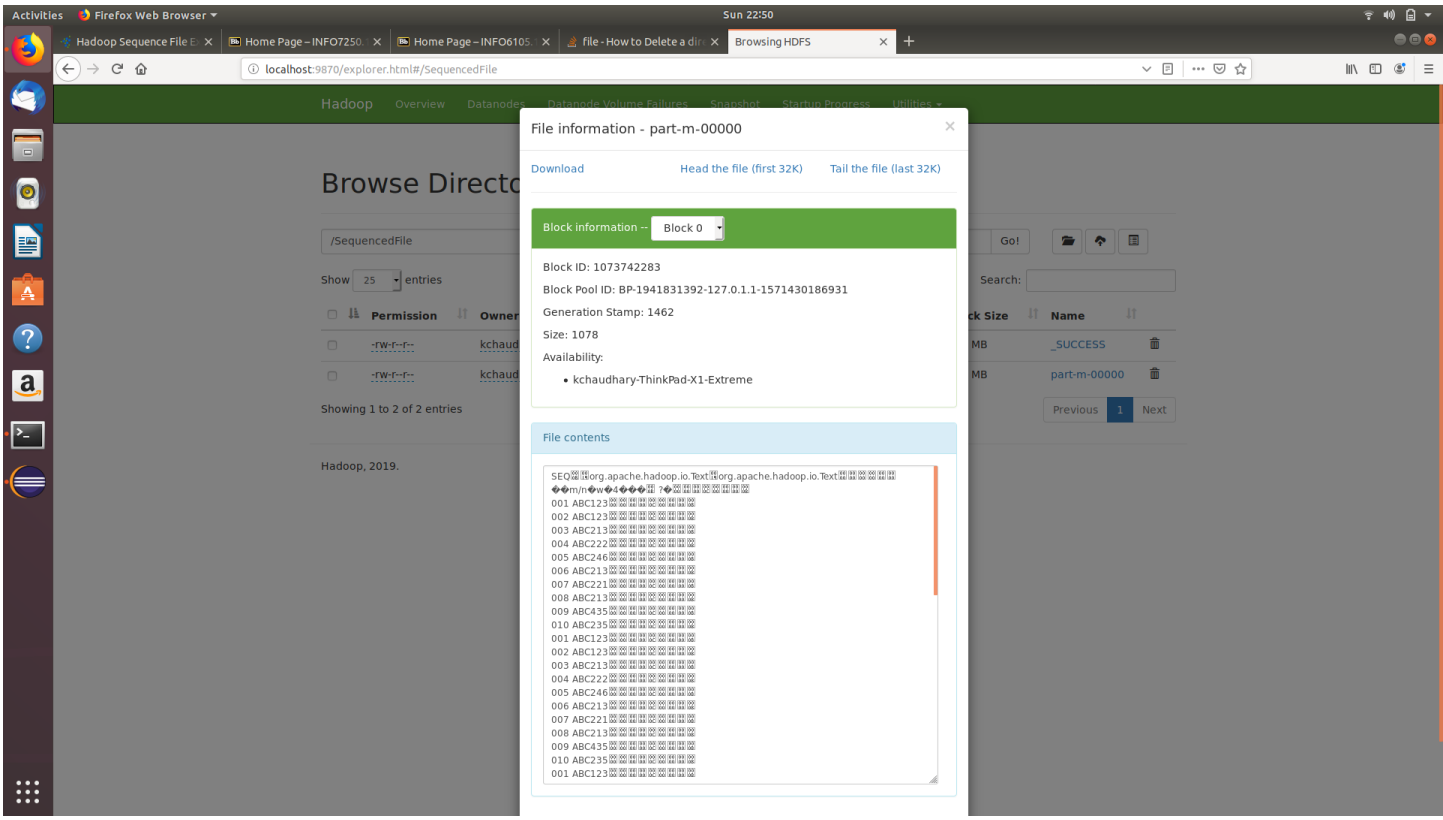
File contents

```
AA   94.62
AAI  57.88
AAN  35.21
AAP  83.65
AAR  25.25
AAV  24.78
AB   94.94
ARA  27.94
```

6. SequenceFileInputFormat

Create a Sequence File





NAME : KAUSHAL CHAUDHARY
NUID : 001886763

Running SequenceFileFormatReaderJob