Experiment 5: Hadoop Word Count and Join Approach

Namansh Singh Maurya
22MIA1034

Aim:

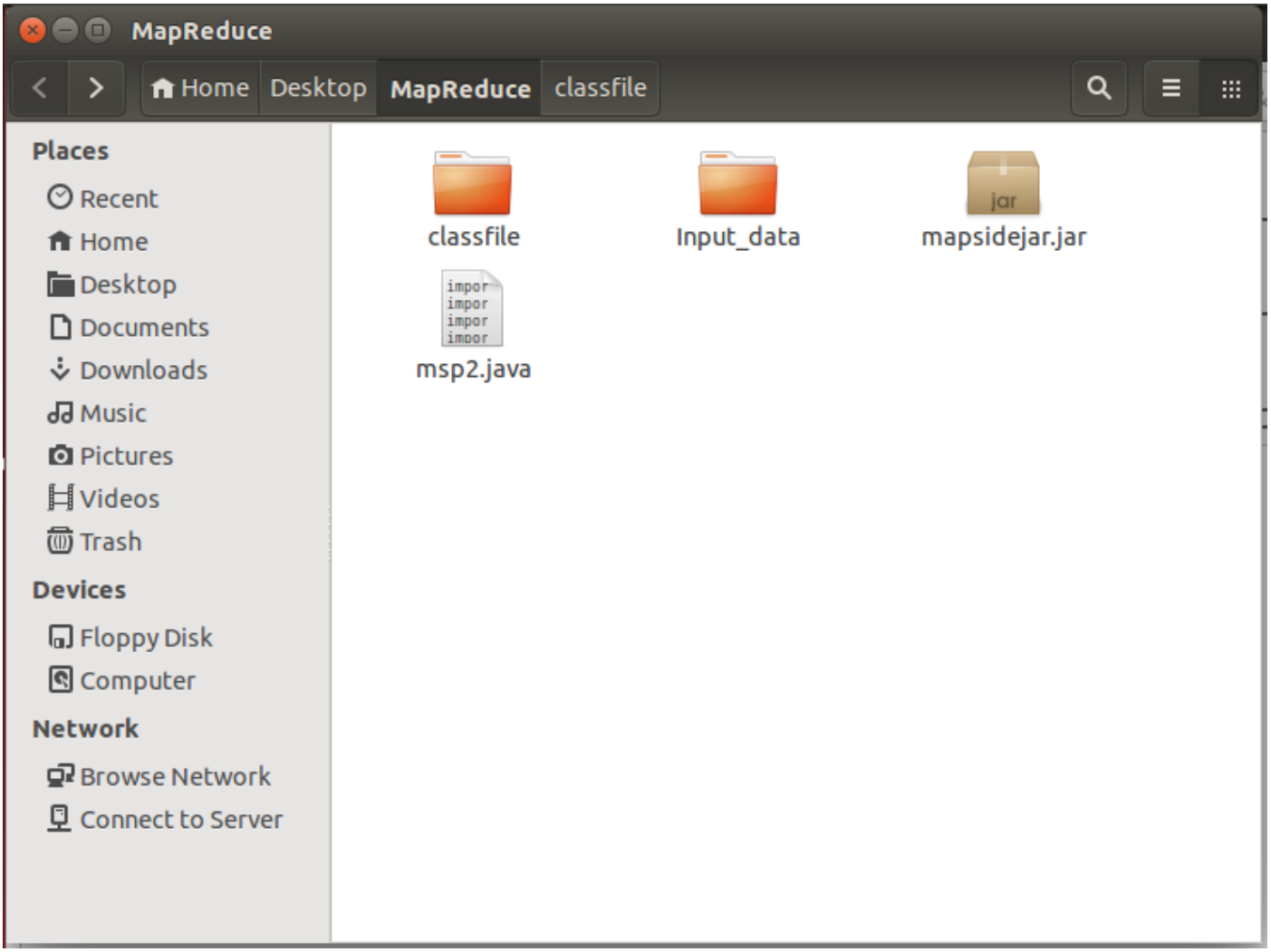
Implement the Word Count program and the Join approach using Hadoop MapReduce.

Algorithm/Procedure:

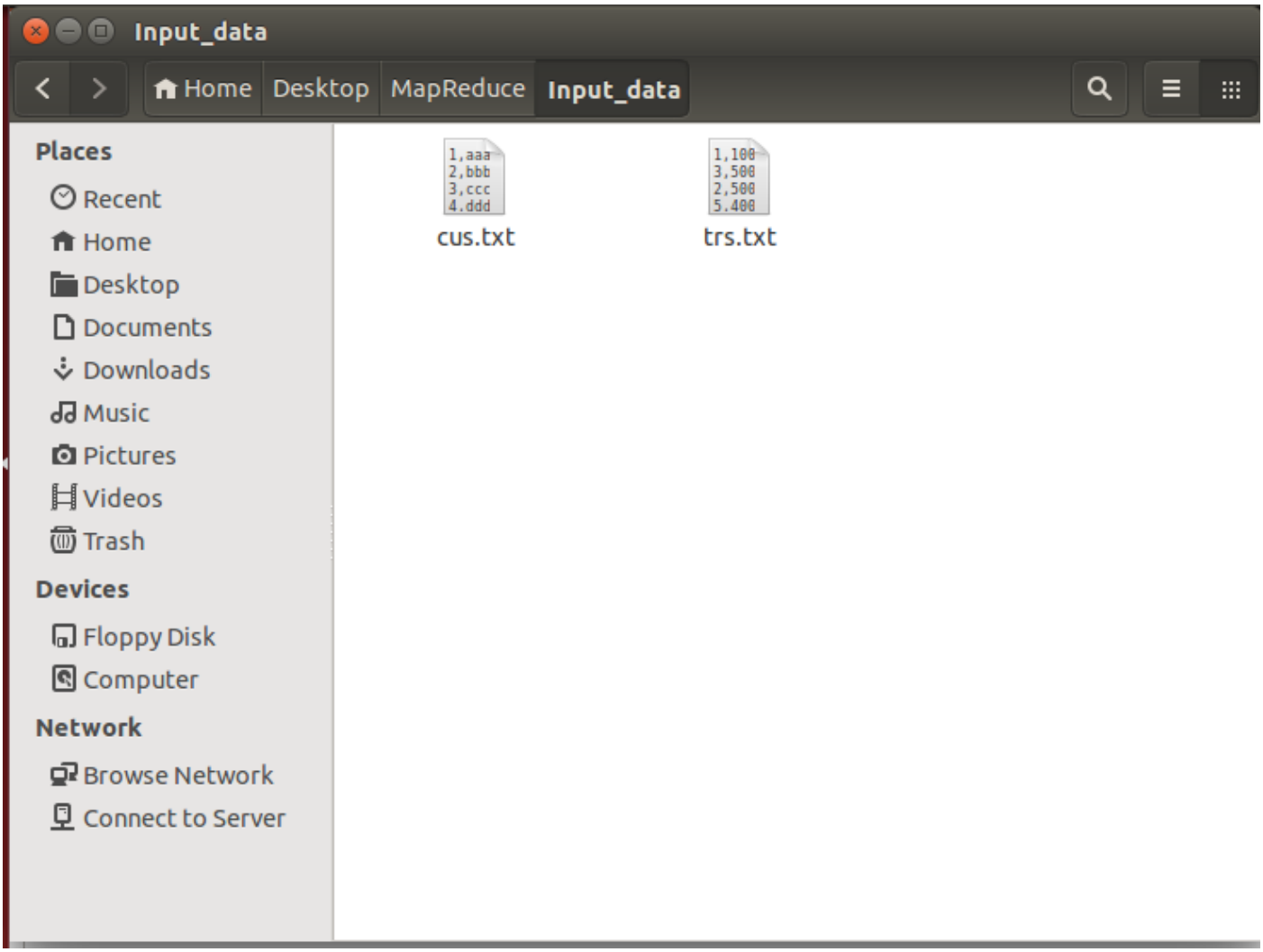
1. Firstly, we will start our Hadoop system and check if is working correctly or not, with that we will export the environment variable of Hadoop and echo it back to the terminal.

```
ponny@ubuntu:~$ start-all.sh
Warning: $HADOOP_HOME is deprecated.

starting namenode, logging to /home/ponny/hadoop/libexec/../logs/hadoop-ponny-namenode-ubuntu.out
localhost: starting datanode, logging to /home/ponny/hadoop/libexec/../logs/hadoop-ponny-datanode
-ubuntu.out
localhost: starting secondarynamenode, logging to /home/ponny/hadoop/libexec/../logs/hadoop-ponny
-secondarynamenode-ubuntu.out
starting jobtracker, logging to /home/ponny/hadoop/libexec/../logs/hadoop-ponny-jobtracker-ubuntu
.out
localhost: starting tasktracker, logging to /home/ponny/hadoop/libexec/../logs/hadoop-ponny-taskt
racker-ubuntu.out
ponny@ubuntu:~$ jps
13155 Jps
12871 SecondaryNameNode
12954 JobTracker
12539 NameNode
13116 TaskTracker
12704 DataNode
ponny@ubuntu:~$ javac -version
javac 1.7.0_201
ponny@ubuntu:~$ export HADOOP
ponny@ubuntu:~$ export HADOOP_CLASSPATH=$(hadoop classpath)
Warning: $HADOOP_HOME is deprecated.

ponny@ubuntu:~$ echo $HADOOP_CLASSPATH
/home/ponny/hadoop/libexec/../conf:/usr/lib/jvm/java-7-openjdk-i386/lib/tools.jar:/home/ponny/had
oop/libexec/..:/home/ponny/hadoop/libexec/../hadoop-core-1.0.4.jar:/home/ponny/hadoop/libexec/../
lib/asm-3.2.jar:/home/ponny/hadoop/libexec/../lib/aspectjrt-1.6.5.jar:/home/ponny/hadoop/libexec/
../lib/aspectjtools-1.6.5.jar:/home/ponny/hadoop/libexec/../lib/commons-beanutils-1.7.0.jar:/home
/ponny/hadoop/libexec/../lib/commons-beanutils-core-1.8.0.jar:/home/ponny/hadoop/libexec/../lib/c
ommons-cli-1.2.jar:/home/ponny/hadoop/libexec/../lib/commons-codec-1.4.jar:/home/ponny/hadoop/lib
exec/../lib/commons-collections-3.2.1.jar:/home/ponny/hadoop/libexec/../lib/commons-configuration
-1.6.jar:/home/ponny/hadoop/libexec/../lib/commons-daemon-1.0.1.jar:/home/ponny/hadoop/libexec/..
/lib/commons-digester-1.8.jar:/home/ponny/hadoop/libexec/../lib/commons-el-1.0.jar:/home/ponny/ha
doop/libexec/../lib/commons-httpclient-3.0.1.jar:/home/ponny/hadoop/libexec/../lib/commons-io-2.1
.jar:/home/ponny/hadoop/libexec/../lib/commons-lang-2.4.jar:/home/ponny/hadoop/libexec/../lib/com
mons-logging-1.1.1.jar:/home/ponny/hadoop/libexec/../lib/commons-logging-api-1.0.4.jar:/home/ponn
y/hadoop/libexec/../lib/commons-math-2.1.jar:/home/ponny/hadoop/libexec/../lib/commons-net-1.4.1.
jar:/home/ponny/hadoop/libexec/../lib/core-3.1.1.jar:/home/ponny/hadoop/libexec/../lib/hadoop-cap
acity-scheduler-1.0.4.jar:/home/ponny/hadoop/libexec/../lib/hadoop-fairscheduler-1.0.4.jar:/home/
ponny/hadoop/libexec/../lib/hadoop-thriftfs-1.0.4.jar:/home/ponny/hadoop/libexec/../lib/hsqldb-1.
8.0.10.jar:/home/ponny/hadoop/libexec/../lib/jackson-core-asl-1.8.8.jar:/home/ponny/hadoop/libexe
c/../lib/jackson-mapper-asl-1.8.8.jar:/home/ponny/hadoop/libexec/../lib/jasper-compiler-5.5.12.ja
r:/home/ponny/hadoop/libexec/../lib/jasper-runtime-5.5.12.jar:/home/ponny/hadoop/libexec/../lib/j
deb-0.8.jar:/home/ponny/hadoop/libexec/../lib/jersey-core-1.8.jar:/home/ponny/hadoop/libexec/../l
ib/jersey-json-1.8.jar:/home/ponny/hadoop/libexec/../lib/jersey-server-1.8.jar:/home/ponny/hadoop
/libexec/../lib/jets3t-0.6.1.jar:/home/ponny/hadoop/libexec/../lib/jetty-6.1.26.jar:/home/ponny/h
```

2. Then we will create one folder named **MapReduce** inside that we will make **Input_data** and **classfile** named folders and **msp2.java** file

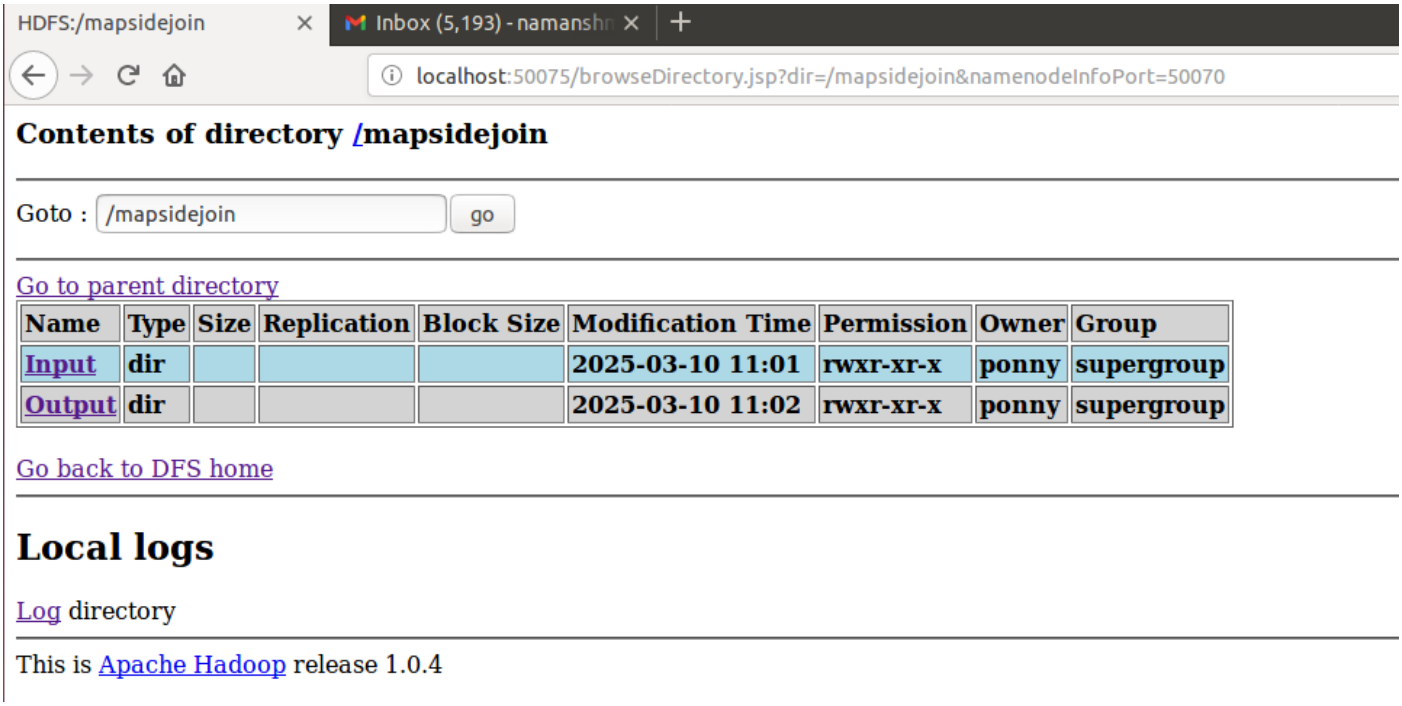


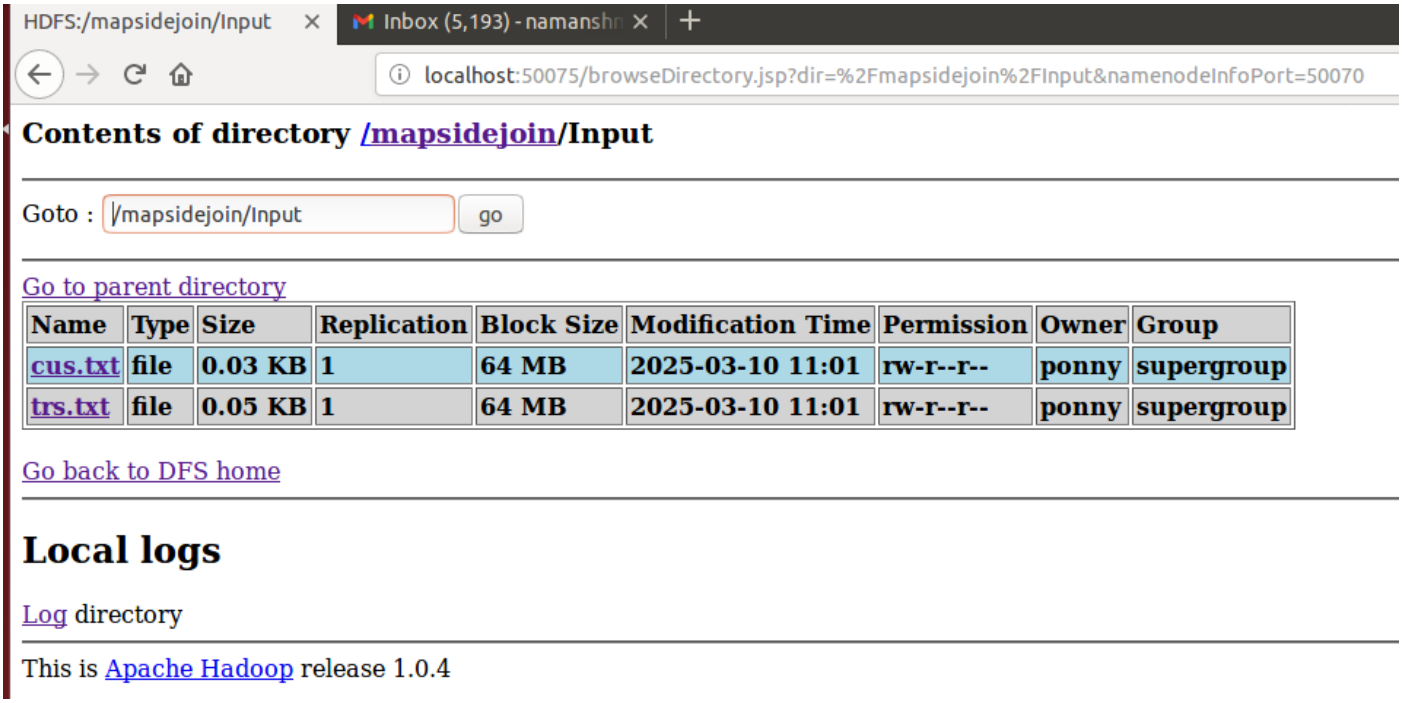
3. Inside **Input_data** we will create two text files named **cus.txt** and **trs.txt**



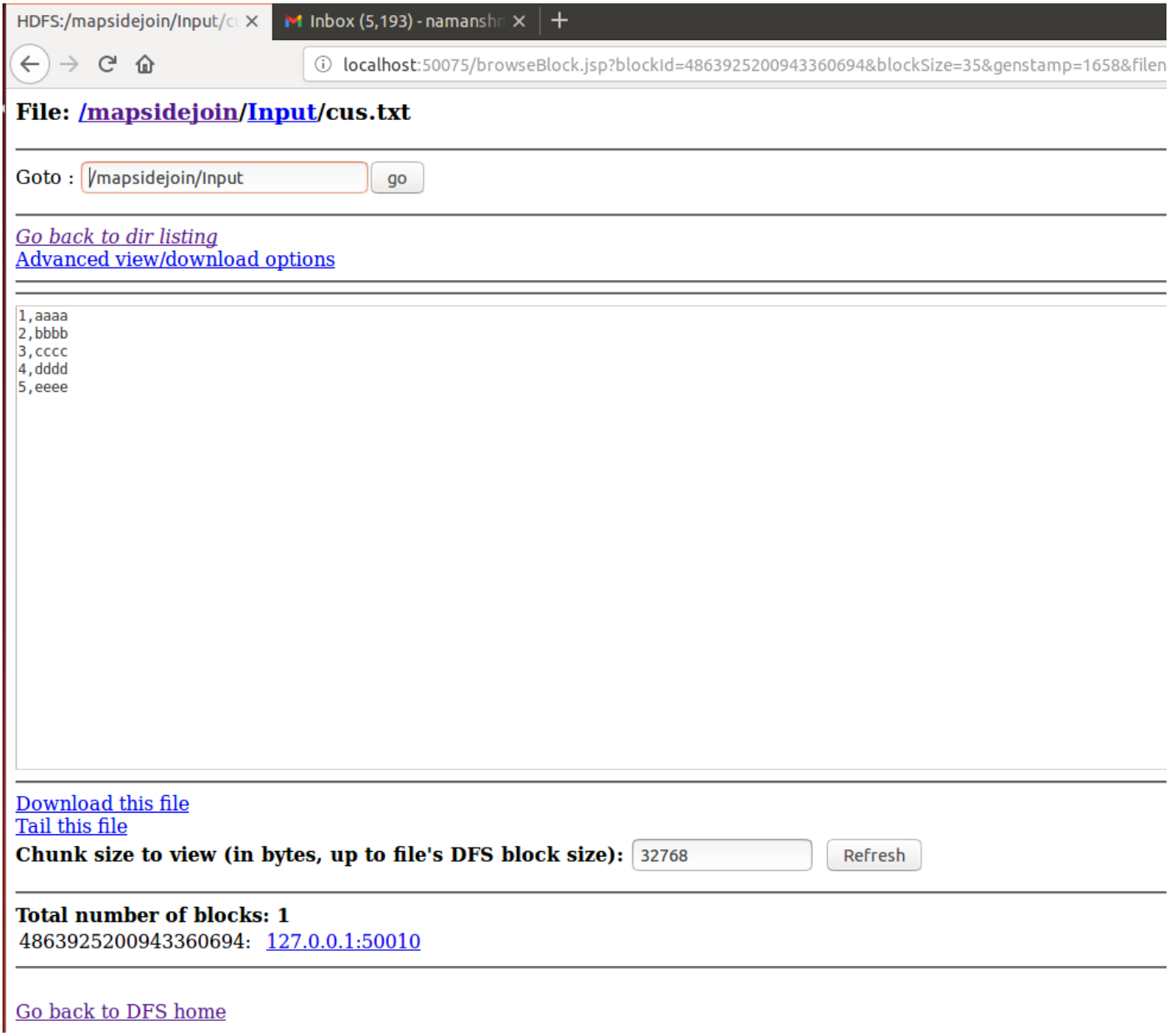
4. After this we will make **mapsidejoin** directory in the Hadoop system we can view that using **localhost**, and we will put both the text file inside **mapsidejoin/Input** folder. Then we will enter the **MapReduce** directory for making required files inside **classfile**.

```
t-1.4.1.jar:/home/ponny/hadoop/libexec/../lib/core-3.1.1.jar:/home/ponny/hadoop/libexec/../lib/ha
doop-capacity-scheduler-1.0.4.jar:/home/ponny/hadoop/libexec/../lib/hadoop-fairscheduler-1.0.4.ja
r:/home/ponny/hadoop/libexec/../lib/hadoop-thriftfs-1.0.4.jar:/home/ponny/hadoop/libexec/../lib/h
sqldb-1.8.0.10.jar:/home/ponny/hadoop/libexec/../lib/jackson-core-asl-1.8.8.jar:/home/ponny/hadoo
p/libexec/../lib/jackson-mapper-asl-1.8.8.jar:/home/ponny/hadoop/libexec/../lib/jasper-compiler-5
.5.12.jar:/home/ponny/hadoop/libexec/../lib/jasper-runtime-5.5.12.jar:/home/ponny/hadoop/libexec/
../lib/jdeb-0.8.jar:/home/ponny/hadoop/libexec/../lib/jersey-core-1.8.jar:/home/ponny/hadoop/libe
xec/../lib/jersey-json-1.8.jar:/home/ponny/hadoop/libexec/../lib/jersey-server-1.8.jar:/home/ponn
y/hadoop/libexec/../lib/jets3t-0.6.1.jar:/home/ponny/hadoop/libexec/../lib/jetty-6.1.26.jar:/home
/ponny/hadoop/libexec/../lib/jetty-util-6.1.26.jar:/home/ponny/hadoop/libexec/../lib/jsch-0.1.42.
jar:/home/ponny/hadoop/libexec/../lib/junit-4.5.jar:/home/ponny/hadoop/libexec/../lib/kfs-0.2.2.j
ar:/home/ponny/hadoop/libexec/../lib/log4j-1.2.15.jar:/home/ponny/hadoop/libexec/../lib/mockito-a
ll-1.8.5.jar:/home/ponny/hadoop/libexec/../lib/oro-2.0.8.jar:/home/ponny/hadoop/libexec/../lib/se
rvlet-api-2.5-20081211.jar:/home/ponny/hadoop/libexec/../lib/slf4j-api-1.4.3.jar:/home/ponny/hado
op/libexec/../lib/slf4j-log4j12-1.4.3.jar:/home/ponny/hadoop/libexec/../lib/xmlenc-0.52.jar:/home
/ponny/hadoop/libexec/../lib/jsp-2.1/jsp-2.1.jar:/home/ponny/hadoop/libexec/../lib/jsp-2.1/jsp-ap
i-2.1.jar
ponny@ubuntu:~$ hadoop fs -mkdir /mapsidejoin
Warning: $HADOOP_HOME is deprecated.

mkdir: org.apache.hadoop.hdfs.server.namenode.SafeModeException: Cannot create directory /mapside
join. Name node is in safe mode.
ponny@ubuntu:~$ hadoop dfsadmin -safemode leave
Warning: $HADOOP_HOME is deprecated.

Safe mode is OFF
ponny@ubuntu:~$ hadoop fs -mkdir /mapsidejoin
Warning: $HADOOP_HOME is deprecated.

ponny@ubuntu:~$ hadoop fs -mkdir /mapsidejoin/Input
Warning: $HADOOP_HOME is deprecated.

ponny@ubuntu:~$ hadoop fs -put '/home/ponny/Desktop/MapReduce/Input_data/cus.txt'  /mapsidejoin/I
nput
Warning: $HADOOP_HOME is deprecated.

ponny@ubuntu:~$ hadoop fs -put '/home/ponny/Desktop/MapReduce/Input_data/trs.txt' /mapsidejoin/In
put
Warning: $HADOOP_HOME is deprecated.

ponny@ubuntu:~$ cd Desktop
ponny@ubuntu:~/Desktop$ cd MapReduce
ponny@ubuntu:~/Desktop/MapReduce$ javac -classpath ${HADOOP_CLASSPATH} -d /home/ponny/Desktop/Map
Reduce/classfile /home/ponny/Desktop/MapReduce/msp2.java
ponny@ubuntu:~/Desktop/MapReduce$ jar -cvf mapsidejar.jar -C classfile/ .
added manifest
adding: msp2.class(in = 1560) (out= 874)(deflated 43%)
adding: msp2$JoinMapper.class(in = 3402) (out= 1452)(deflated 57%)
```





And here are the contents of the files


HDFS:/mapsidejoin/Input/c
Inbox (5,193) - namanshr
localhost:50075/browseBlock.jsp?blockId=4863925200943360694&blockSize=35&genstamp=1658&filen
File: /mapsidejoin/Input/cus.txt
Goto : /mapsidejoin/Input
go
Go back to dir listing
Advanced view/download options
1,aaaa
2,bbbb
3,cccc
4,dddd
5,eeee
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
4863925200943360694: 127.0.0.1:50010
Go back to DFS home

HDFS:/mapsidejoin/Input/tr × Inbox (5,193) - namanshr ×

localhost:50075/browseBlock.jsp?blockId=-7858327703898079541&blockSize=49&gens

File: /mapsidejoin/Input/trs.txt

Goto : /mapsidejoin/Input go

Go back to dir listing
Advanced view/download options

```
1,100
3,5000
2,500
5,400
4,600
3,200
4,300
1,700
```

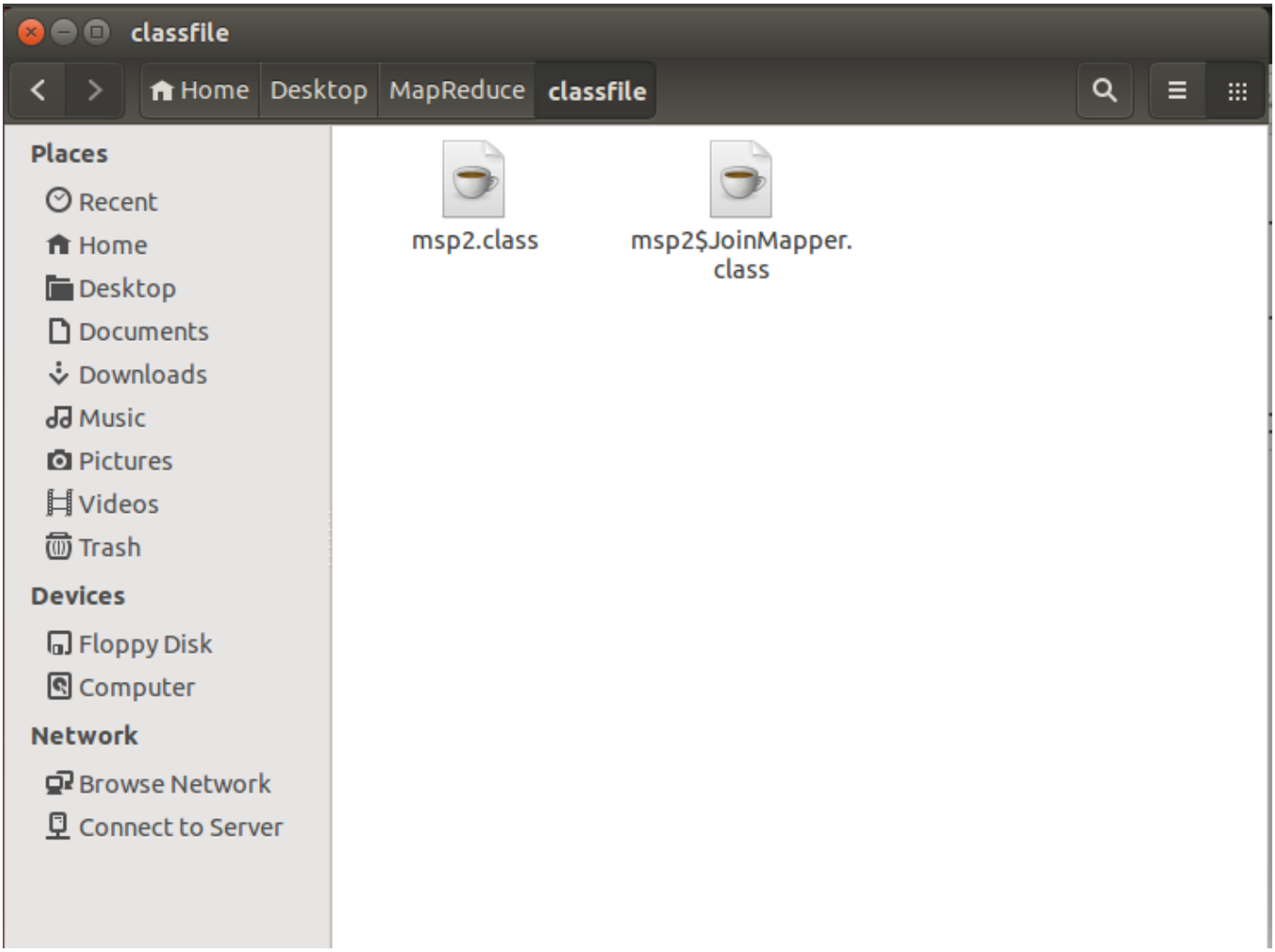
Download this file
Tail this file
Chunk size to view (in bytes, up to file's DFS block size): 32768 Refresh

---

Total number of blocks: 1
-7858327703898079541: 127.0.0.1:50010

Go back to DFS home

5. Now using this command **javac -classpath ${HADOOP_CLASSPATH} -d /home/ponny/Desktop/MapReduce/classfile /home/ponny/Desktop/MapReduce/msp2.java**, we will get the following two files in the **classfile** directory.



6. Now we will compile the java file then we will get jar file and we will execute the code and will get the required output, inside the output folder on Hadoop.

```
ponny@ubuntu:~/Desktop$ cd MapReduce
ponny@ubuntu:~/Desktop/MapReduce$ javac -classpath ${HADOOP_CLASSPATH} -d /home/ponny/Desktop/Map
Reduce/classfile /home/ponny/Desktop/MapReduce/msp2.java
ponny@ubuntu:~/Desktop/MapReduce$ jar -cvf mapsidejar.jar -C classfile/ .
added manifest
adding: msp2.class(in = 1560) (out= 874)(deflated 43%)
adding: msp2$JoinMapper.class(in = 3402) (out= 1452)(deflated 57%)
ponny@ubuntu:~/Desktop/MapReduce$ hadoop jar mapsidejar.jar msp2 /mapsidejoin/Input/trs.txt /maps
idejoin/Output /mapsidejoin/Input/cus.txt
Warning: $HADOOP_HOME is deprecated.

25/03/10 11:02:29 WARN mapred.JobClient: Use GenericOptionsParser for parsing the arguments. Appl
ications should implement Tool for the same.
25/03/10 11:02:29 INFO input.FileInputFormat: Total input paths to process : 1
25/03/10 11:02:29 INFO util.NativeCodeLoader: Loaded the native-hadoop library
25/03/10 11:02:29 WARN snappy.LoadSnappy: Snappy native library not loaded
25/03/10 11:02:29 INFO mapred.JobClient: Running job: job_202503101059_0001
25/03/10 11:02:30 INFO mapred.JobClient:  map 0% reduce 0%
25/03/10 11:02:47 INFO mapred.JobClient:  map 100% reduce 0%
25/03/10 11:02:52 INFO mapred.JobClient: Job complete: job_202503101059_0001
25/03/10 11:02:52 INFO mapred.JobClient: Counters: 19
25/03/10 11:02:52 INFO mapred.JobClient:   Job Counters
25/03/10 11:02:52 INFO mapred.JobClient:     SLOTS_MILLIS_MAPS=13306
25/03/10 11:02:52 INFO mapred.JobClient:     Total time spent by all reduces waiting after reserv
ing slots (ms)=0
25/03/10 11:02:52 INFO mapred.JobClient:     Total time spent by all maps waiting after reserving
 slots (ms)=0
25/03/10 11:02:52 INFO mapred.JobClient:     Launched map tasks=1
25/03/10 11:02:52 INFO mapred.JobClient:     Data-local map tasks=1
25/03/10 11:02:52 INFO mapred.JobClient:     SLOTS_MILLIS_REDUCES=0
25/03/10 11:02:52 INFO mapred.JobClient:   File Output Format Counters
25/03/10 11:02:52 INFO mapred.JobClient:     Bytes Written=73
25/03/10 11:02:52 INFO mapred.JobClient:   FileSystemCounters
25/03/10 11:02:52 INFO mapred.JobClient:     HDFS_BYTES_READ=162
25/03/10 11:02:52 INFO mapred.JobClient:     FILE_BYTES_WRITTEN=21661
25/03/10 11:02:52 INFO mapred.JobClient:     HDFS_BYTES_WRITTEN=73
25/03/10 11:02:52 INFO mapred.JobClient:   File Input Format Counters
25/03/10 11:02:52 INFO mapred.JobClient:     Bytes Read=49
25/03/10 11:02:52 INFO mapred.JobClient:   Map-Reduce Framework
25/03/10 11:02:52 INFO mapred.JobClient:     Map input records=8
25/03/10 11:02:52 INFO mapred.JobClient:     Physical memory (bytes) snapshot=45903872
25/03/10 11:02:52 INFO mapred.JobClient:     Spilled Records=0
25/03/10 11:02:52 INFO mapred.JobClient:     CPU time spent (ms)=80
25/03/10 11:02:52 INFO mapred.JobClient:     Total committed heap usage (bytes)=16252928
25/03/10 11:02:52 INFO mapred.JobClient:     Virtual memory (bytes) snapshot=384790528
25/03/10 11:02:52 INFO mapred.JobClient:     Map output records=8
25/03/10 11:02:52 INFO mapred.JobClient:     SPLIT_RAW_BYTES=113
ponny@ubuntu:~/Desktop/MapReduce$
```

Program:

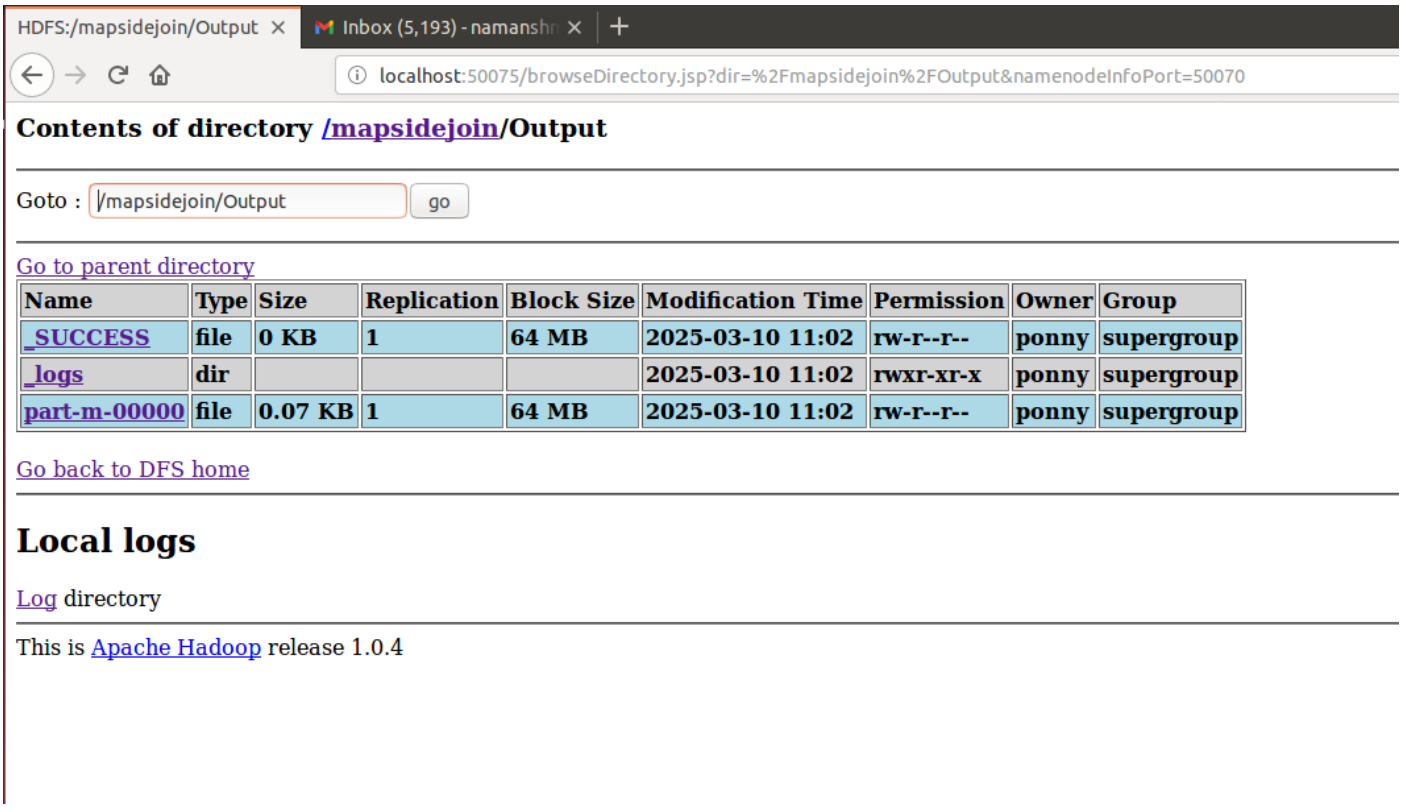
Mapsidejoin Java program

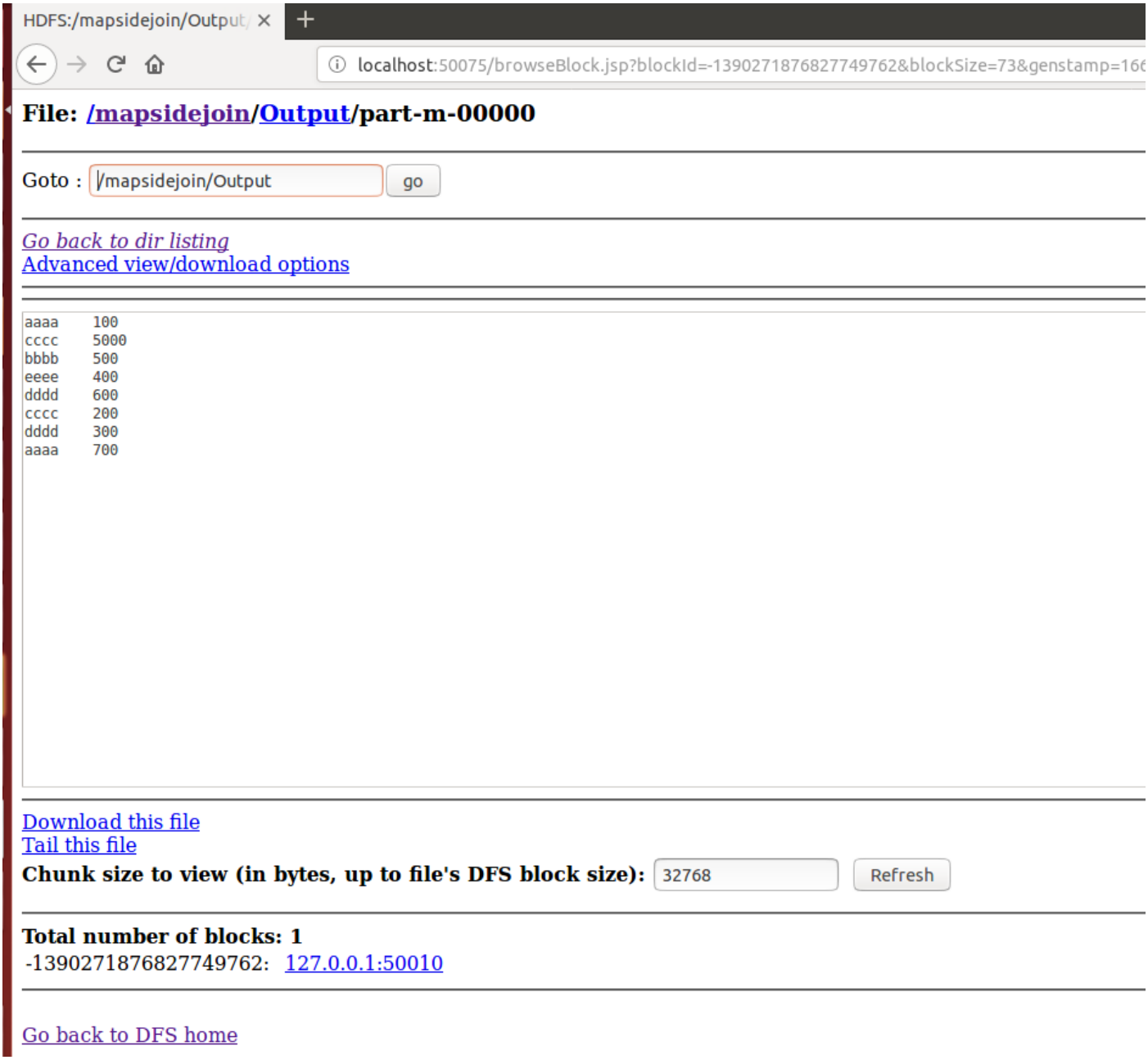
```
import java.io.*;
import java.net.URI;
import java.util.HashMap;
import java.util.Map;
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.conf.Configuration;
import org.apache.hadoop.filecache.DistributedCache;
import org.apache.hadoop.io.*;
import org.apache.hadoop.mapreduce.*;
import org.apache.hadoop.mapreduce.lib.input.FileInputFormat;
import org.apache.hadoop.mapreduce.lib.output.FileOutputFormat;

public class msp2 {

    public static class JoinMapper extends Mapper<LongWritable, Text, Text, Text> {
        private Map<String, String> userMap = new HashMap<>();
        private Text outputKey = new Text();
        private Text outputValue = new Text();

        @Override
        protected void setup(Context context) throws IOException, InterruptedException {
            // Read the distributed cache file
            Configuration conf = context.getConfiguration();
            Path[] cacheFiles = DistributedCache.getLocalCacheFiles(conf);
            if (cacheFiles != null && cacheFiles.length > 0) {
                BufferedReader reader = new BufferedReader(new
FileReader(cacheFiles[0].toString()));
                String line;
                while ((line = reader.readLine()) != null) {
                    String[] parts = line.split(",");
                    userMap.put(parts[0], parts[1]);  // UserID -> UserName
                }
                reader.close();
        System.err.println("deb"+ userMap.size() + "users.");

            } else {
throw new IOException("dis cah is missing");}
        }

        @Override
        protected void map(LongWritable key, Text value, Context context) throws
IOException, InterruptedException {
            String[] transaction = value.toString().split(",");
            String userId = transaction[0];
            String amount = transaction[1];

            if (userMap.containsKey(userId)) {
                outputKey.set(userMap.get(userId));  // UserName
                outputValue.set(amount);
                context.write(outputKey, outputValue);
```

```
            }
        }
    }

    public static void main(String[] args) throws Exception {
if(args.length<3){
System.err.println("err");
System.exit(1);
}
        Configuration conf = new Configuration();
        DistributedCache.addCacheFile(new URI(args[2]),conf);
        Job job = new Job(conf);

        job.setJarByClass(msp2.class);
        job.setMapperClass(JoinMapper.class);

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(Text.class);

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        // Add users.txt to the Distributed Cache


        job.setNumReduceTasks(0);  // No reducer required for mapside join

        System.exit(job.waitForCompletion(true) ? 0 : 1);
    }
}
```

Output:

After Joining both the text files we will be getting the output as given in the below attached image.





Result:

The Join approach in Hadoop MapReduce was successfully implemented to merge employee details with their respective department names using the distributed computing framework, enabling efficient data processing at scale.