**Documentation of Where Clause - Map reduce**

-Prepared by Vignesh.R

Now the previous result is refined to add a WHERE clause. The results are filtered to include flights that had a delay in arrival or departure by more than 10 minutes. Also an additional attribute that indicates whether the delay was during arrival, departure, or both is added. This attribute is called Point Of Delay, and it takes one of the following values:

* for delays at origin
* D for delays at destination
* B for delays at both origin and destination

Note that we used values O, D, and B instead of Origin, Destination and Both. This substitution is not just for brevity. When running a typical MapReduce program, considerable output can be produced, and the main bottleneck for most MapReduce programs is disk or network I/O. When large amounts of data are being written out to disk or moved over the network, restricting the size of a field can result in a significant improvement in the runtime performance of MapReduce jobs.

The program that performs the WHERE clause is org.apress.prohadoop.c5.WhereClauseMRJob.

**Creating.jarfile for Where clause and exporting it:-**

**Step 1:**

Open Eclipse and Click on File > New > Java Project.

**Step 2:**

Give the name ‘Where’ as your project name and click ‘Finish’.

**Step 3:**

Right click on ‘Where’ project and select ‘Properties’. Click ‘Java Build Path’ and switch to Libraries tab and click on ‘Add external JARs’.

**Step 4:**

Select all the JAR files in usr >> lib >> hadoop directory to add them.

**Step 5:**

Again add all jar files in usr >> lib >> hadoop >> client directory and press OK.

**Step 6:**

**Right click on src, New >>Class.**

**Step 7:**

Enter the project name as ‘WhereClauseMRJob’ and click ‘Finish’.

**Step 8:**

Open browser and copy and paste the Java Source code of WhereClause program from the link given. The packages are automatically generated by the Eclipse.

**Website link:**<https://github.com/Apress/pro-apache-hadoop/blob/master/prohadoop/src/main/java/org/apress/prohadoop/c5/WhereClauseMRJob.java>

**Website link:**<https://github.com/Apress/pro-apache-hadoop/blob/master/prohadoop/src/main/java/org/apress/prohadoop/utils/AirlineDataUtils.java>

**Website link:**<https://github.com/Apress/pro-apache-hadoop/blob/master/prohadoop/src/main/java/org/apress/prohadoop/c6/MonthDoWOnlyWritable.java>

**Website link:**<https://github.com/Apress/pro-apache-hadoop/blob/master/prohadoop/src/main/java/org/apress/prohadoop/c6/MonthDoWWritable.java>

**Website link:**<https://github.com/Apress/pro-apache-hadoop/blob/master/prohadoop/src/main/java/org/apress/prohadoop/c6/DelaysWritable.java>

**Step 9:**

Right click on the WhereClauseMRJava project and select Export >> Java >> JAR file. Then click on ‘Next’.

**Step 10:**

Name the JAR file and click ‘Finish’.

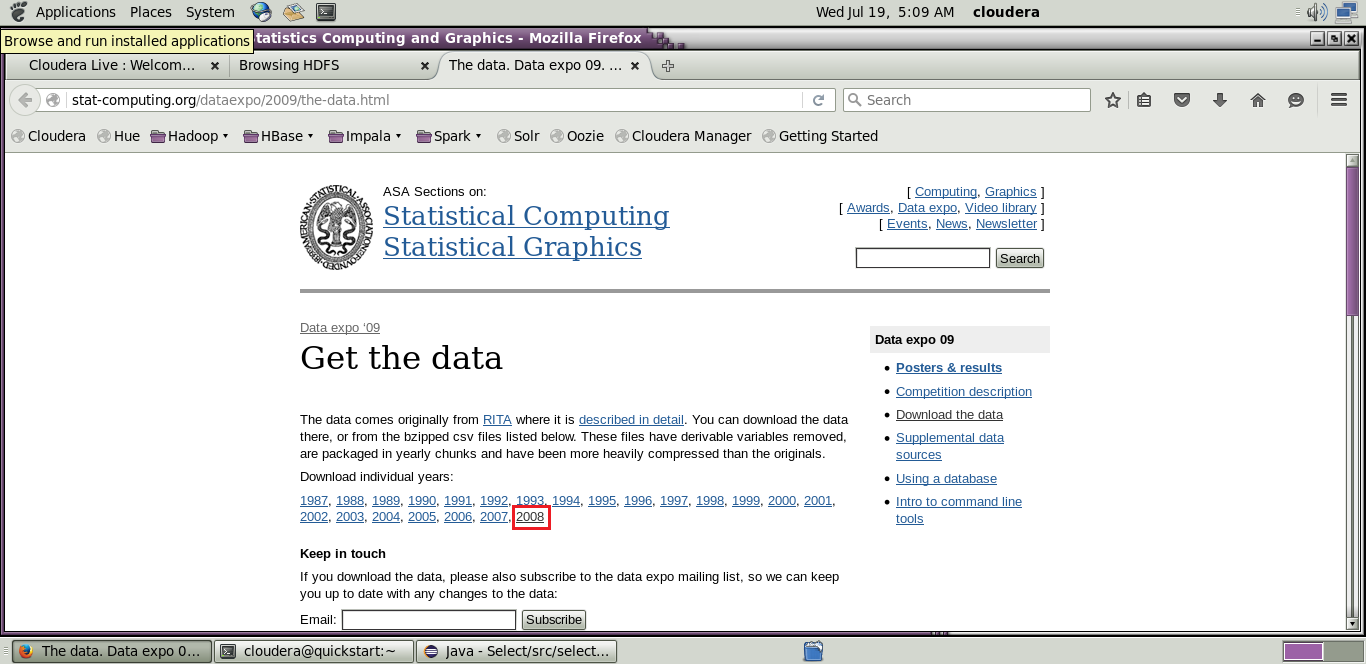
**Creating a file for Mapreduce job to work on:**

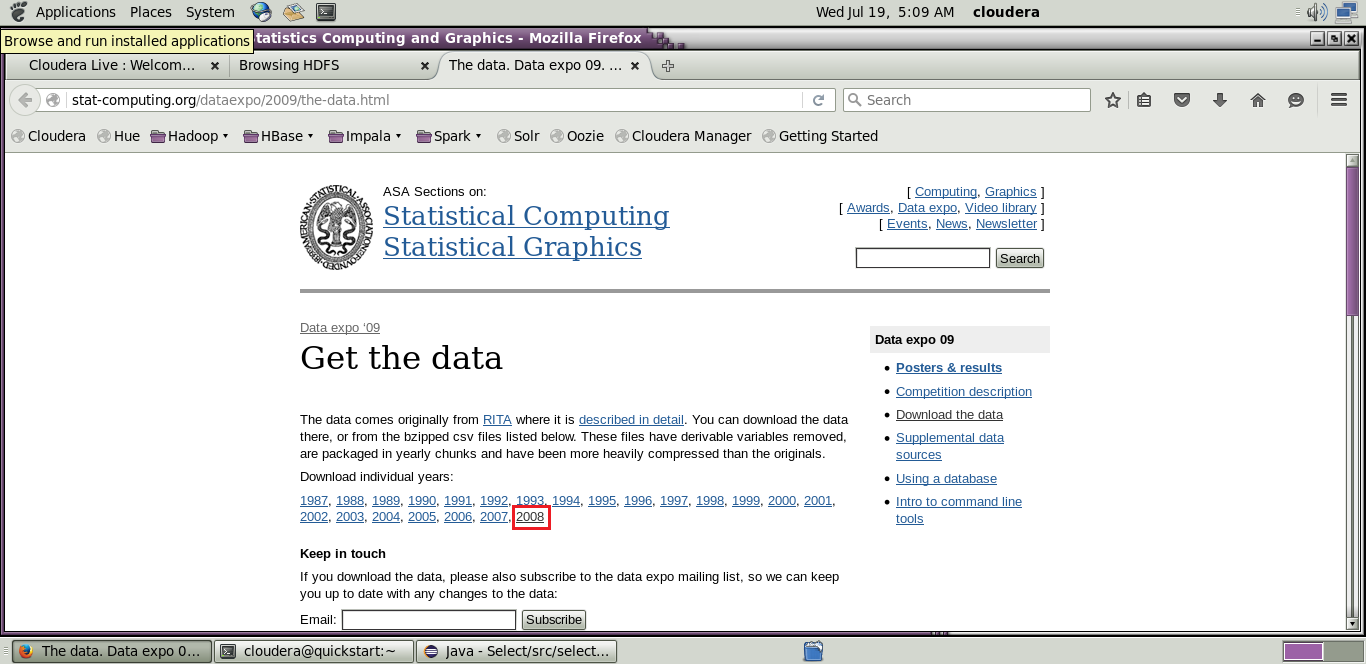
**Step 11:**Make a new Directory using the following command.

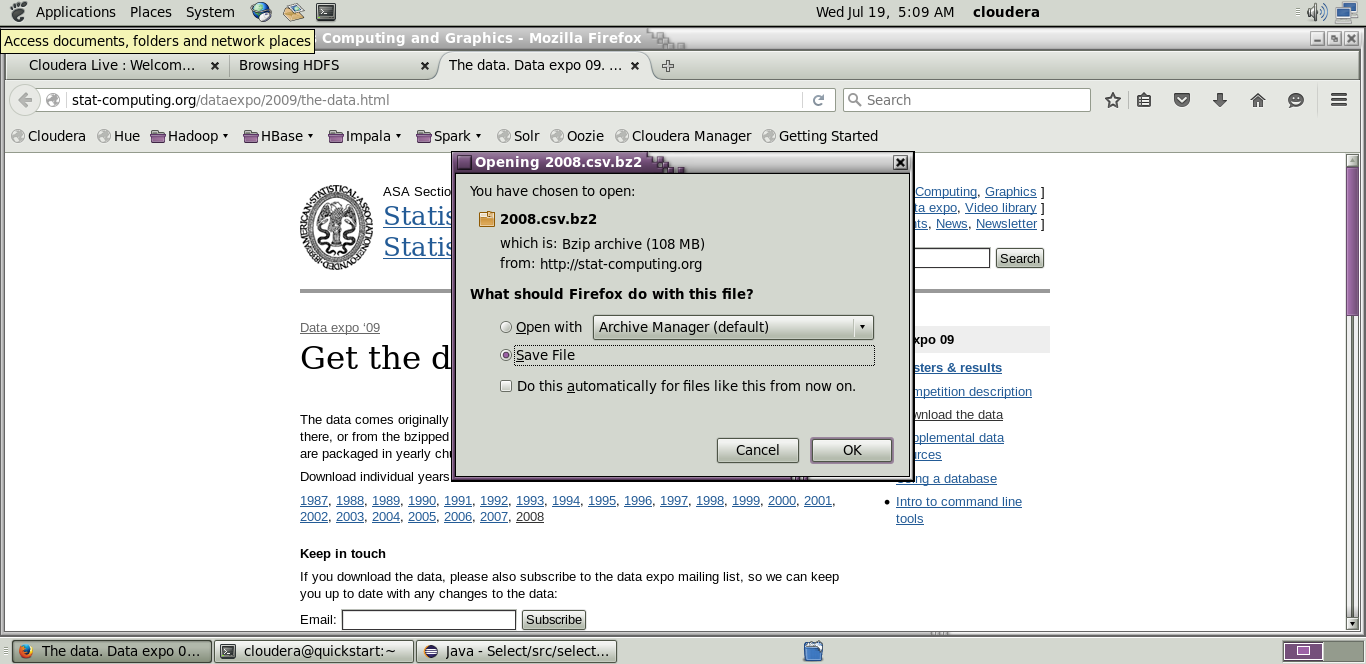
**Code:hadoop fs -mkdir /airline**

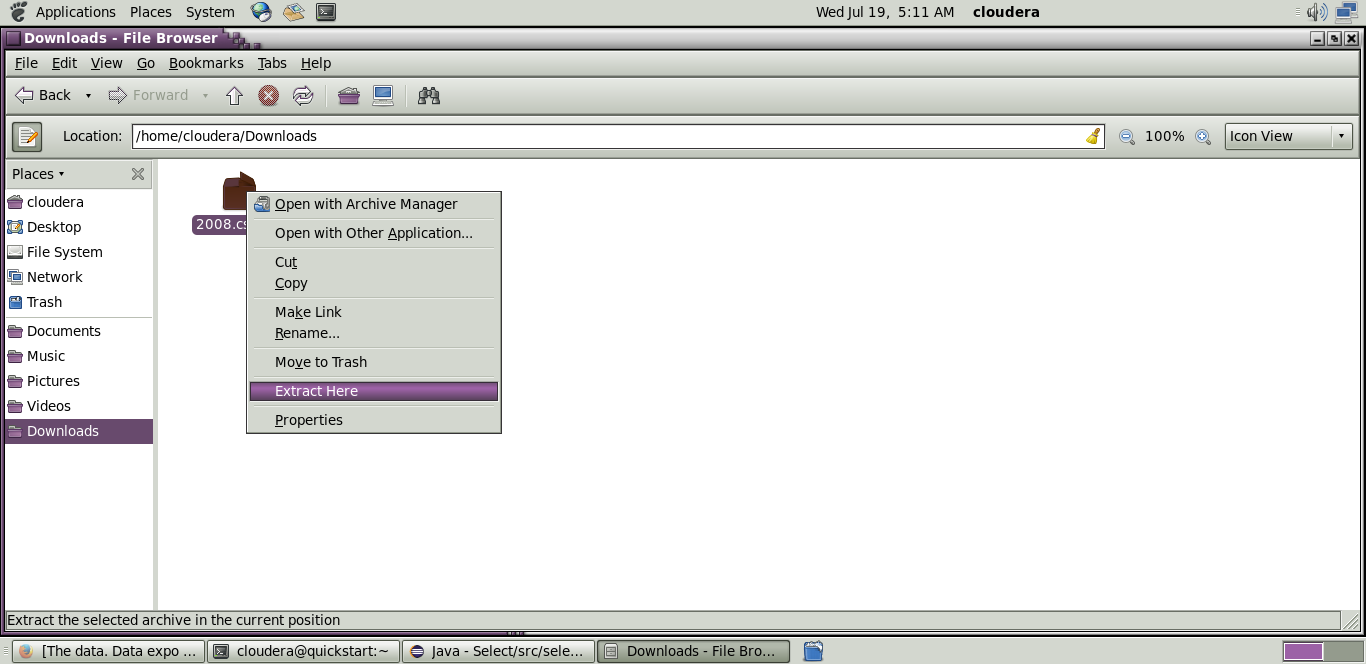
**Step 12:**Use the following link to download the csv file.

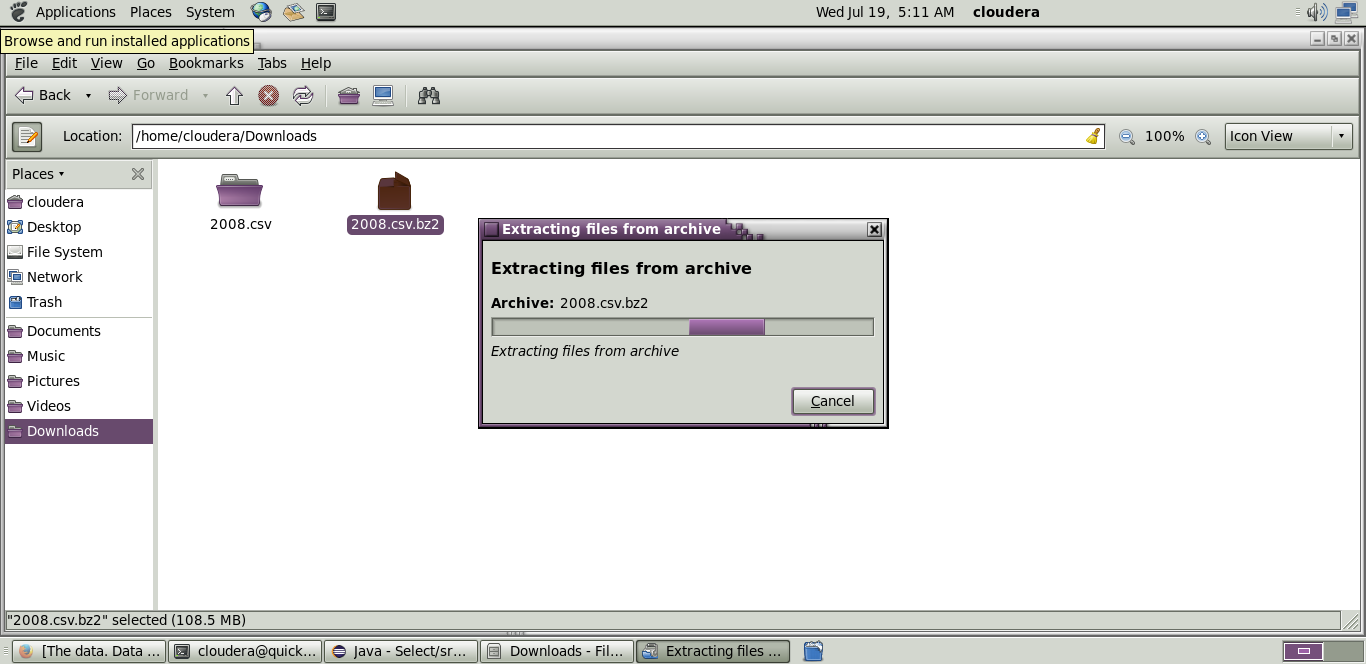
**Website link:**

[http://stat-computing.org/dataexpo/2009/the-data.html](http://stat-computing.org/dataexpo/2009/the-data.html )

[](http://stat-computing.org/dataexpo/2009/the-data.html )







**Step 13:** Copy the downloaded file to the new directory created in HDFS.

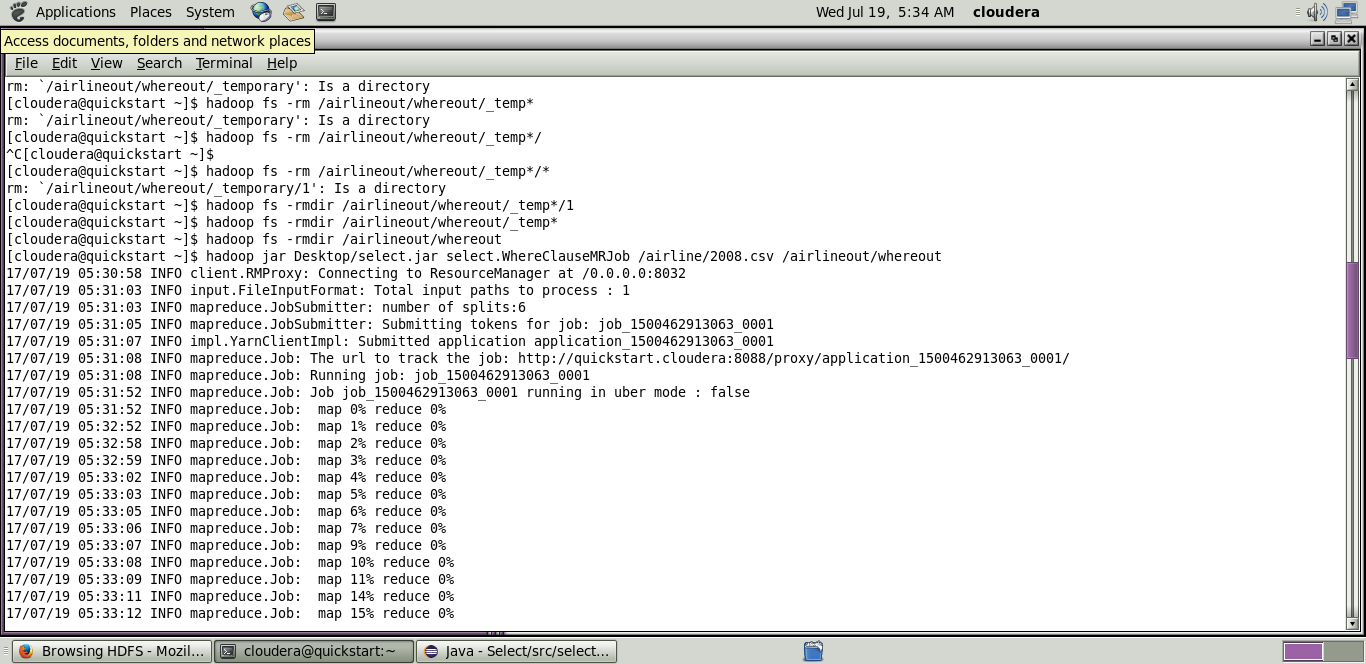
**Code:hadoop fs -put /home/cloudera/Downloads/2008.csv/airline**

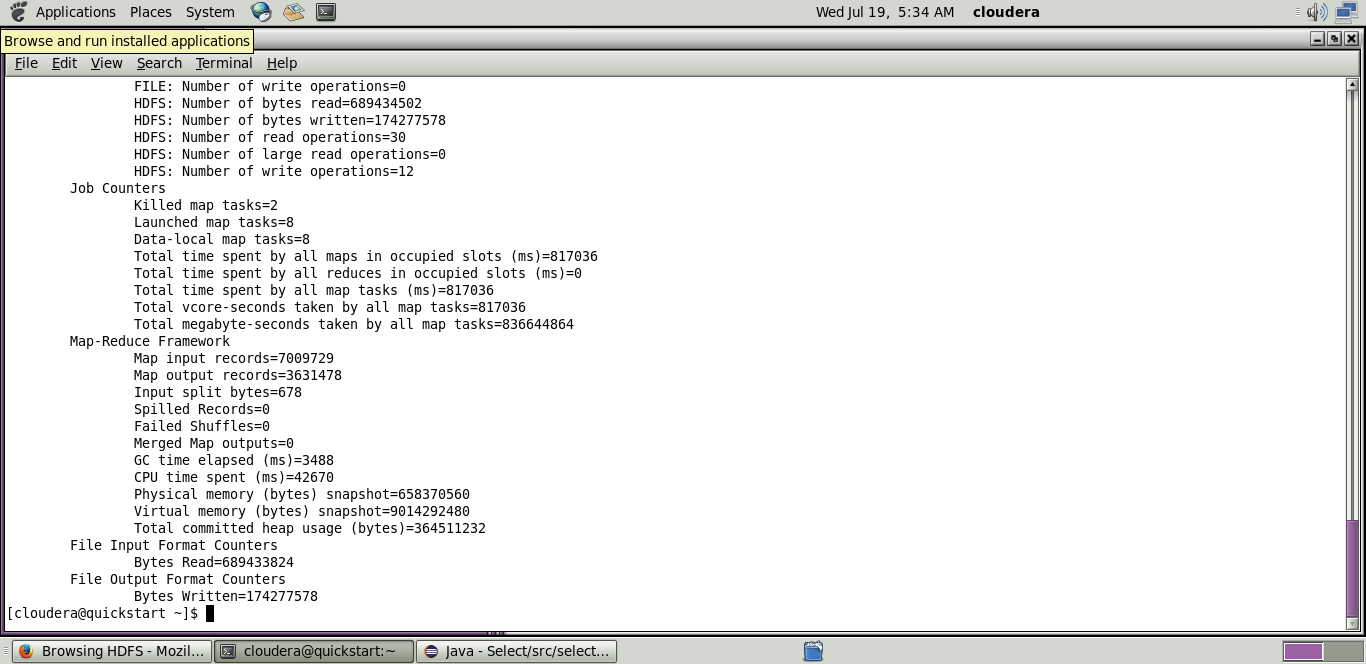
**Initializing mapreduce job:**

**Step 14:**

Initialize the mapreduce job by giving the following command and wait for sometime.

**Code:**hadoop jar Desktop/select.jar select.WhereClauseMRJob /airline/2008.csv /airlineout/whereout





Now wait for about 50-70 seconds while the mapreduce job is being performed for the data created earlier.

**Output mapreduce job:**

**Step 15:**

The output directory of the mapreduce program is listed using the following command.

**Code:**hadoop fs -ls/airlineout/whereout/\*

**Step 16:**

The final output of the mapreduce program is found using the following command.

**Code:**hadoop fs -cat /airlineout/whereout/\*

