**BigData And Hadoop**

**Project1.2 -** Titanic Data Analysis

**Task3**

**Dataset Description:**

The data set contains information about passengers who boarded Titanic ship.

Column 1: PassengerId

Column 2: Survived (survived=0 & died=1)

Column 3: Pclass

Column 4: Name

Column 5: Sex

Column 6: Age

Column 7: SibSp

Column 8: Parch

Column 9: Ticket

Column 10: Fare

Column 11: Cabin

Column 12: Embarked

**Problem Statement:**

Write a MapReduce/Pig program to find out number of males and females who died in each class.

**Solution**

**Code files are as follows:**

Mapper class: Task3Mapper.java

Reducer class: Task3Reducer.java

Combiner class: Task3Reducer.java

Driver class: Task3Driver.java

**Solution logic:**

We need to find the number of males and females who died in each class.

In the data, class is present in column 3, Person is alive or dead in is column2 and if the person is male/female is in column 5.

In mapper, we have filtered the records and kept only those that we need for this problem statement. The condition used is:

Column 2<Survived> = 1 as we need only those records where passenger died.

After we get the required records, the mapper emits a composite key and 1 as its value into the context.

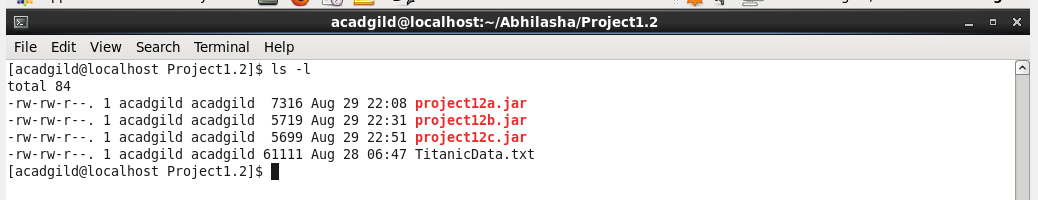
The composite is made up of <Class>-<Sex>. This is because we need to group the records on the class and sex.

In reducer, we count the number of records for each class and sex, composite key containing the class and sex attributes being the key of the mapper, helps us do so.

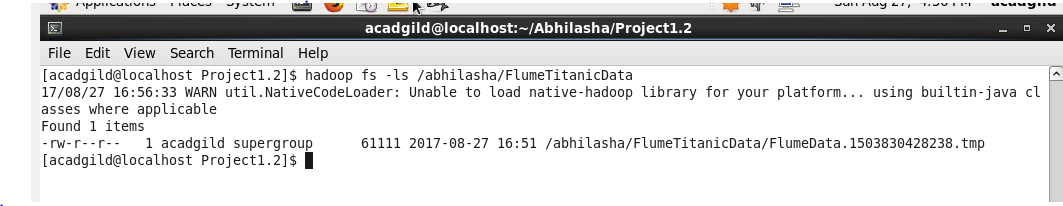
We have used combiner as well, to improve performance, which is same as the reducer class.

**Snapshots of the output are as follows:**

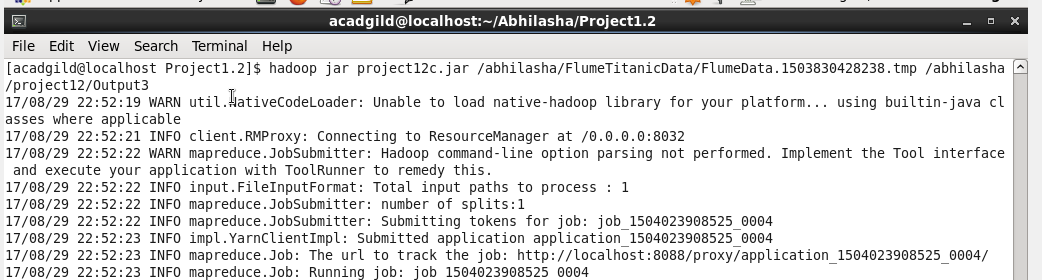
1. The jar is compiled and placed at location ‘/home/acadgild/Abhilasha/Project1.2’. Name of the jar is project12c.jar



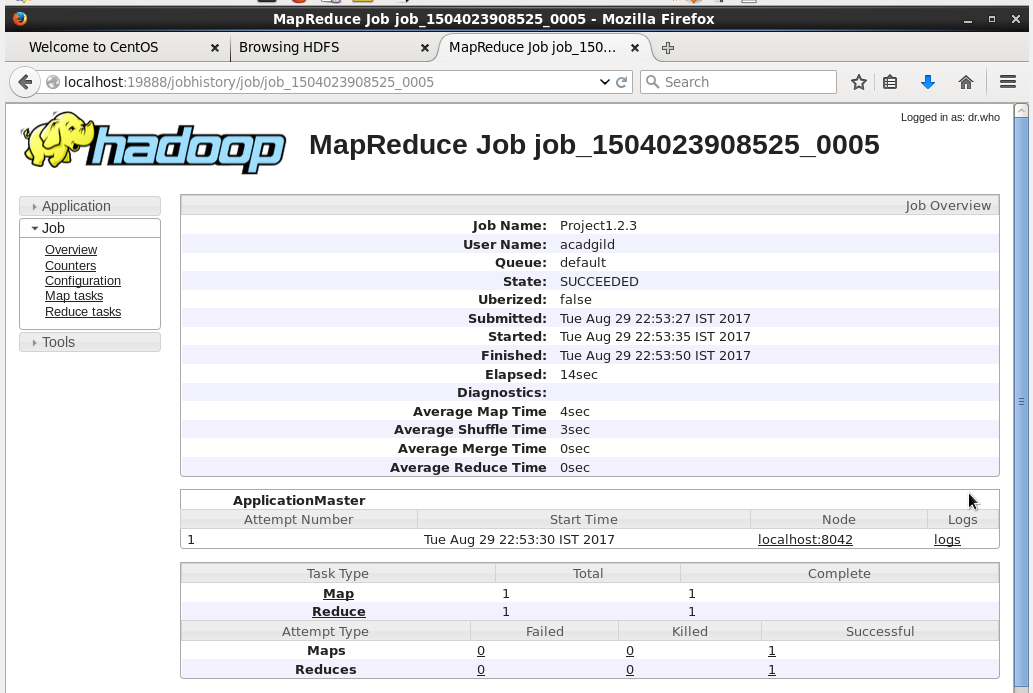
1. Dataset to be used is put into hdfs using flume at the location ‘/abhilasha/FLumeTitanicData’ as follows:



1. We now execute the map-reduce code as follows. In this, we are using the dataset put in hdfs using flume as input. The output folder mentioned is ‘/abhilasha/project12/Output3’

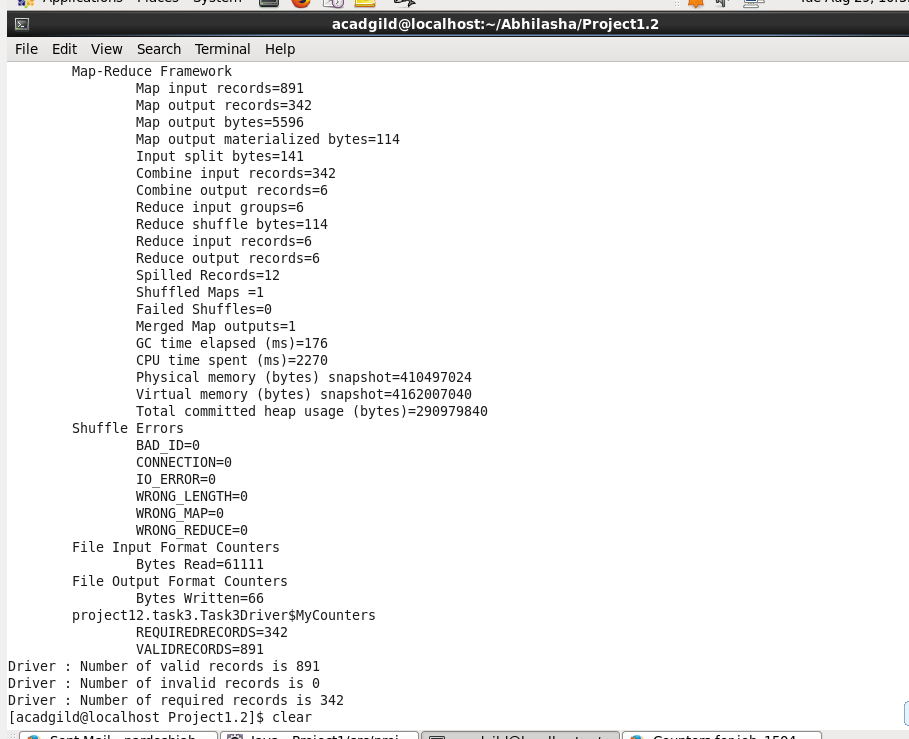


1. Job’s successful completion is shown on the job history server as follows:

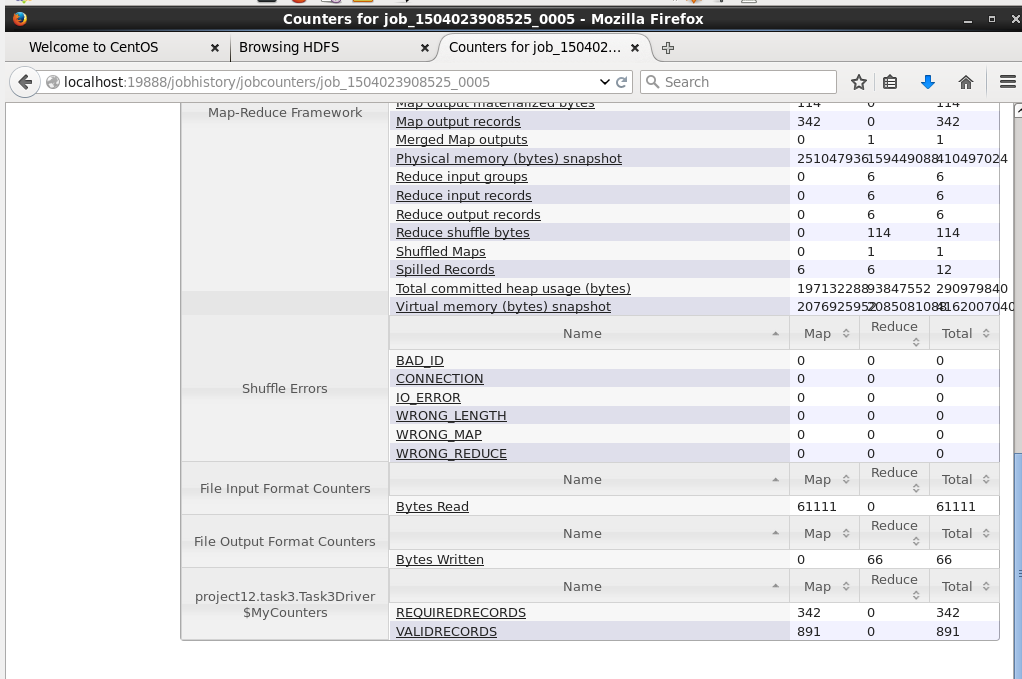


1. We used custom counters in the code, their values are as shown in the screen shot. We used these counters to find number of valid, required and invalid records. Invalid are the ones that have missing columns in the record or that have empty value in column of survived, sex or class. Required records are the ones that satisfy the condition in the mapper i.e.

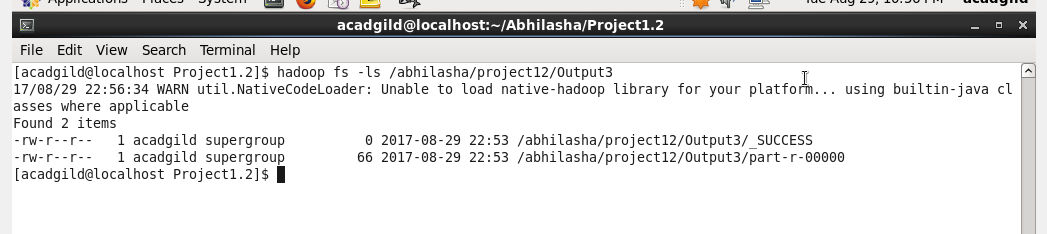
Column 2<Survived> = 1



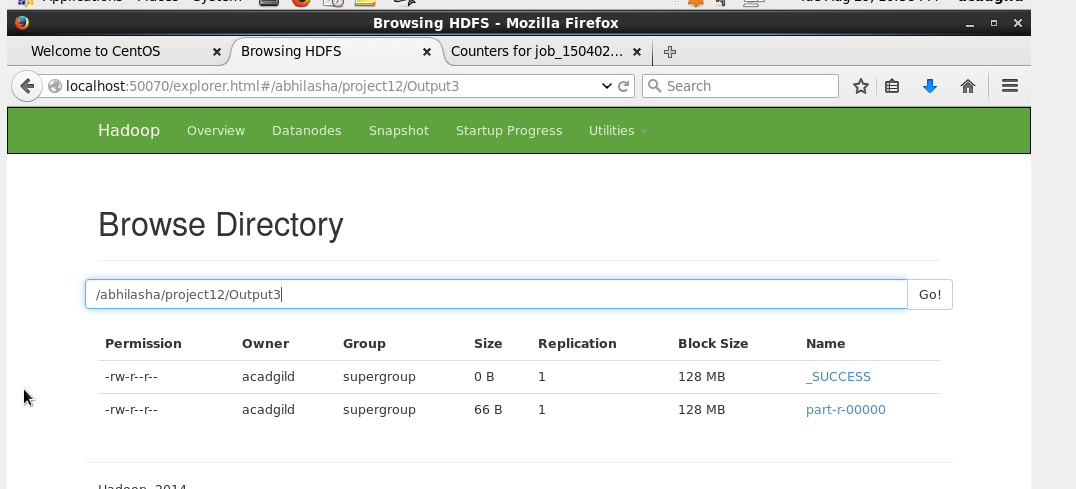
These custom counters are also visible on history server in Counters section as follows:



1. Listing the directory Output3, which contains output of map-reduce as follows:



We can also see this from HDFS UI as follows:



1. The output of the map-reduce program is as follows:

We have used cat command to display the output file ‘/abhilasha/project12/Output3/part-r-00000’ on the terminal.

Format of the output is

<Class>-<Sex> <Count>

