**NOTE: All the java class, Jar, Input and Output files, which I have mentioned in document, are enclosed in [Hadoop\_final\_Poc.Zip]**

**POC 1: Analyzing Book- Crossing Data**

**Data set Description:**

The Book-Crossing dataset consists of 3 tables.

BX-Users:

This file contains the list of the users, their age and where they are collected. If that data is unavailable for any field then it is filled with NULL.

BX-Books:

It gives us the details about the book such as Book-Title, Book-Author, Year-Of-Publication, Publisher, Image-URL and ISBN. Here ISBN will act as a unique code for a book. Invalid ISBNs have already been removed from the dataset. URLs linking to cover images are also given, appearing in three different flavors (`Image-URL-S`, `Image-URL-M`, `Image-URL-L`) i.e. small, medium, large. These URLs point to the Amazon web site.

BX-Book-Ratings:

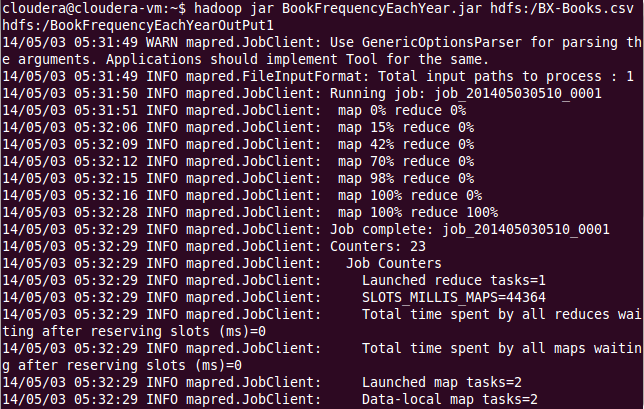
It contains the book rating information. Ratings are either explicitly expressed on a scale from 1-10 (higher values denoting higher appreciation) or implicitly expressed by 0.

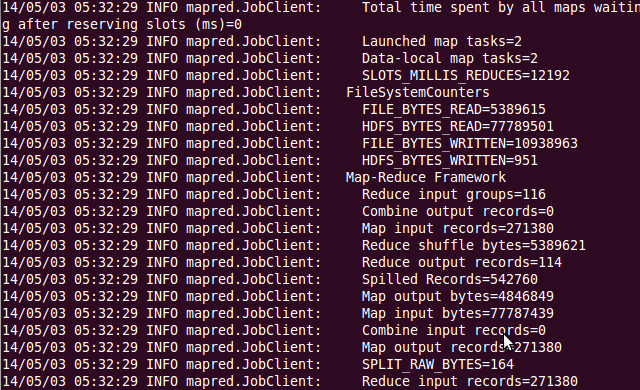
**Problem Statement 1:** Find out the frequency of books published each year.

**Solution:** Achieved Using “**Map Reduce Technique**” and used **BX-Books.csv** Data set.

**Java Class Name: BookFrequency.java** , Logic behind this class is Mapper function will take BX-Books.csv as a input file, And all the fields are formatted ie Double quotes will be remove and will consider **Year Of Publication as a key and ISBN as Value.** Reducer function will then combine each Map output for specific key and then it will provide Count for that particular Year

**Execution: BookFrequencyEachYear.jar** is used to execute this task**.**





**Output File: BookFrequencyEachYearOutPut**

**Output Format: Year Of Publication Book Published Count**

**Problem Statement 2:** out in which year maximum number of books were published.

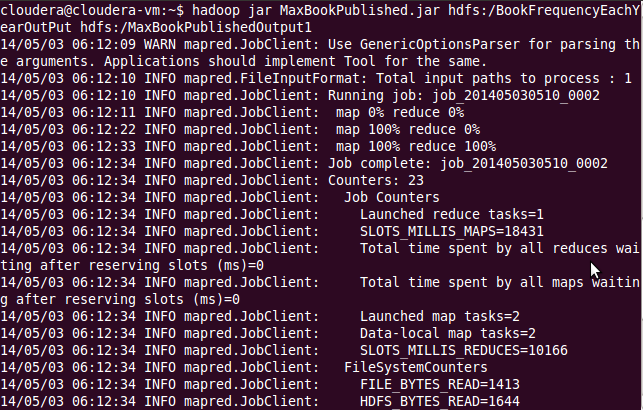
**Solution:** Achieved Using “**Map Reduce Technique**” and used **Output of First task as the Input ie BookFrequencyEachYearOutPut .**

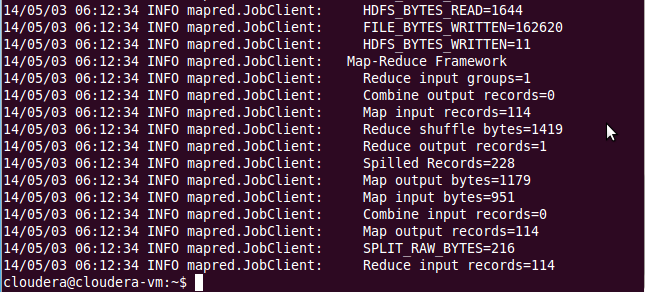
**Java Class Name: MaxBookPublished.java** , Logic behind this class is Mapper function will take BookFrequencyEachYearOutPut as a input file, And Output **Year Of Publication as a key and Book Published Count as Value.** Reducer function will then combine each Map output for specific key and then it will calculate Max book published

**Example:** Input will be 2002 100 , 2003 200 , 2004 150 so after running this task output will be 2003 200 .

**Execution: BookFrequencyEachYear.jar** is used to execute this task**.**

**Output File: MaxBookPublishedOutput**



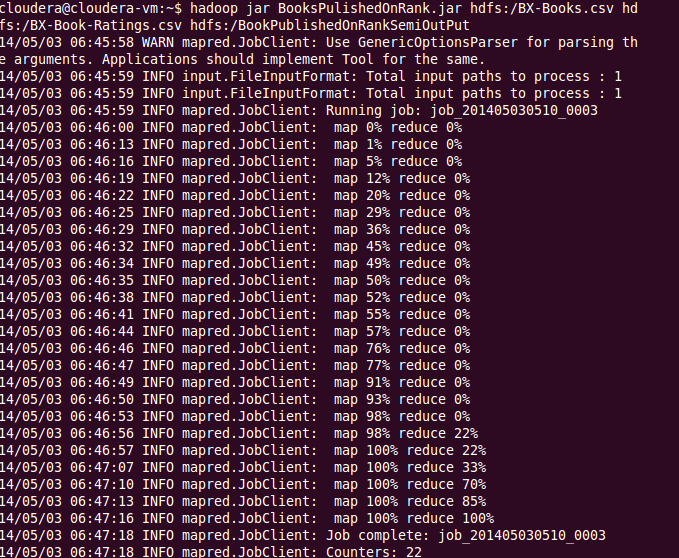


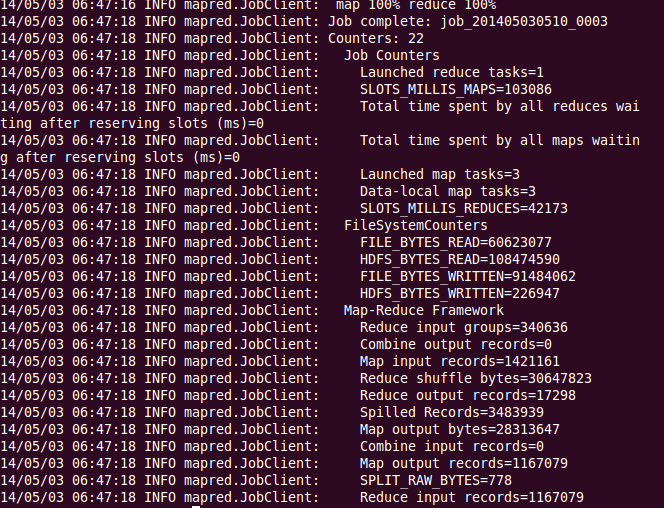
**Problem Statement 3:** Findout how many book were published based on ranking in the year 2002.

**Solution:** Achieved Using “**Map Reduce Technique**” and used **BX-Books.csv** and **Book-Ratings.csv** Data set. Since there are mapping between two dataset, I have used **ReduceSideJoin** Technique for this. Considered ISBN as a key for mapping, **“I also found that for some data, there is no mapping present in Book-Ratings.csv, so I have considered it as Un-Rated in the code”.**

**Java Class Name:** I Have achieved this task using two Map reduce Functions**,** First Map Reduce **BookPublishedOnRanking.java will** join both the input file and generates the output with **BookRating As a key and ISBN as Value.** Second Map Reduce **BookPublishedOnRanking2.java** will take input from first and generates the final OutPut **as Book Rating and Count Of Books Published for 2002**

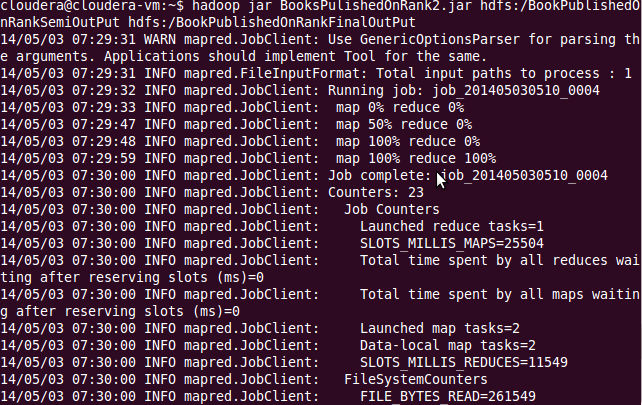
**Execution 1 : BooksPulishedOnRank.jar** is used to execute this task**.**

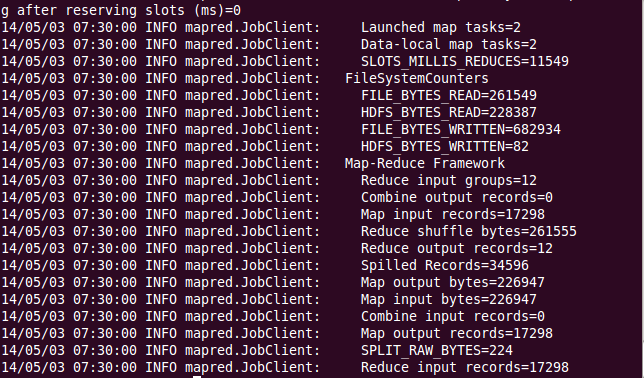




**Output File: BookPublishedOnRankSemiOutPut**

**Execution 2 : BooksPulishedOnRank2.jar** Takes input from first task**.**





**Output File: BookPublishedOnRankFinalOutPut**

**POC 2: Crime Data Analysis**

**Data set Description:**

**Data URL :** http://data.gov.in/catalog/crime-head-wise-cases-reported-under-indian-penal-code-ipc#web\_catalog\_tabs\_block\_10

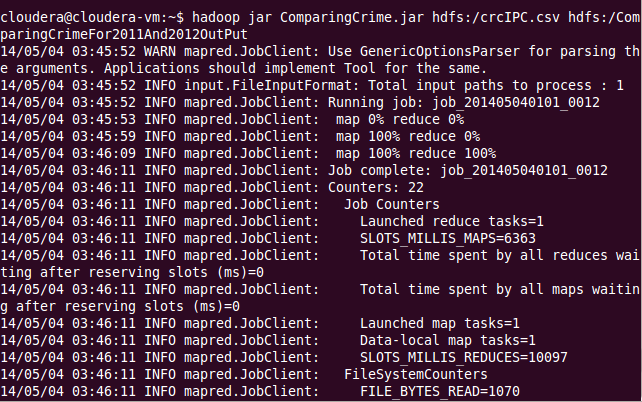
**Data Set :** **crcIPC.csv** , Contains 14 column where column1 = State Name , column2 = Crime Category , and rest other column are crime reported count from 2001 to 2012

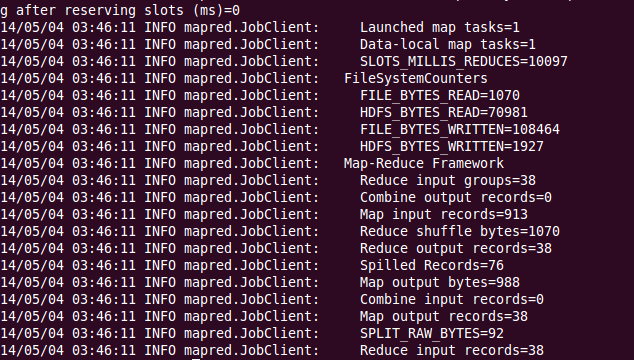
**Problem Statement:** Idea is to compare crime reported for year 2011 and 2012 for each state and for crime category Murder and to find out whether crime reported has been increased or decreased or it is same between 2011 and 2012.

**Solution:** Achieved Using “**Map Reduce Technique**” and used **crcIPC.csv** Data Set.

**Java Class Name:** ComparingCrime.java , Logic behind this class is Mapper function will take crcIPC.csv file as input and extracts only Murder category data and provides output with Key= State + Crime Category and Value = 2011 Count + 2012 Count. Reducer will take above key Value from Mapper and calculates difference and will find out whether murder reported has been increased or decreased etc.

**Execution : ComparingCrime.jar** is used to execute this task**.**





**Output File: ComparingCrimeFor2011And2012OutPut**

**Output Format: Crime Category, State Name, 2011 Count, 2012 Count, Difference, Increased or Decreased statement**