Book- Crossing Project Solution

# Problem Statement 1

* Find out the frequency of books published each year. (Hint: Use Boooks.csv file for this)

**Solution**

This part of the problem was solved using MapReduce program. The maven project will compile and generate the jar files for MapReduce jobs. Then the MapReduce jobs can be run using hadoop command line interface.

Compile the Java programs using maven

$ mvn clean install

The above command will generate:

1. A jar file containing a MapReduce job for finding the frequency of the books published each year.

Jar location in the project:

**PBX\_BookFrequency/target/PBX\_BookFrequency-0.0.1-SNAPSHOT.jar**

1. A jar file containing a MapReduce job for processing BX-Books.csv file.

Jar location in the project:

**PBX\_Books/target/PBX\_Books-0.0.1-SNAPSHOT.jar**

1. A jar file containing a MapReduce job for processing BX-Book-Ratings.csv.

Jar location in the project:

**PBX\_BookRatings/target/PBX\_BookRatings-0.0.1-SNAPSHOT.jar**

Copy all jars to hadoop node.

On hadoop node

First put all threee csv files (BX-Books.csv, BX-Users.csv and BX-Book-Ratings.csv) into HDFS

$ hdfs dfs -mkdir /BookCrossing

$ hdfs dfs -put BX-Books.csv /BookCrossing/

$ hdfs dfs -put BX-Users.csv /BookCrossing/

$ hdfs dfs -put BX-Book-Ratings.csv /BookCrossing/

Run Java program to find the frequency of the books published each year.

$ hadoop jar PBX\_BookFrequency-0.0.1-SNAPSHOT.jar /BookCrossing/BX-Books.csv /BookFrequency/out

Review the output in location /BookFrequency/out HDFS dir for solution.

**HDFS File: /BookFrequency/out/part-r-00000**

**1376 1**

**1378 1**

**1806 1**

**1897 1**

**1900 3**

**1901 7**

**1902 2**

**1904 1**

**1906 1**

**1908 1**

**1909 2**

**1910 1**

**1911 19**

**1914 1**

**1917 1**

**1919 1**

**1920 33**

**1921 2**

**1922 2**

**1923 11**

**1924 2**

**1925 2**

**1926 2**

**1927 2**

**1928 2**

**1929 7**

**1930 13**

**1931 3**

**1932 5**

**1933 4**

**1934 1**

**1935 3**

**1936 7**

**1937 5**

**1938 7**

**1939 9**

**1940 36**

**1941 10**

**1942 14**

**1943 8**

**1944 4**

**1945 8**

**1946 13**

**1947 14**

**1948 9**

**1949 11**

**1950 32**

**1951 40**

**1952 34**

**1953 63**

**1954 54**

**1955 70**

**1956 74**

**1957 76**

**1958 77**

**1959 103**

**1960 133**

**1961 132**

**1962 123**

**1963 132**

**1964 149**

**1965 173**

**1966 183**

**1967 176**

**1968 233**

**1969 339**

**1970 459**

**1971 540**

**1972 773**

**1973 924**

**1974 1024**

**1975 1219**

**1976 1599**

**1977 1897**

**1978 2131**

**1979 2208**

**1980 2676**

**1981 3279**

**1982 4197**

**1983 4499**

**1984 4986**

**1985 5343**

**1986 5841**

**1987 6529**

**1988 7493**

**1989 7937**

**1990 8661**

**1991 9389**

**1992 9906**

**1993 10602**

**1994 11796**

**1995 13548**

**1996 14031**

**1997 14892**

**1998 15767**

**1999 17432**

**2000 17235**

**2001 17360**

**2002 17628**

**2003 14359**

**2004 5839**

**2005 46**

**2006 3**

**2008 1**

**2010 2**

**2011 2**

**2012 1**

**2020 3**

**2021 1**

**2024 1**

**2026 1**

**2030 7**

**2037 1**

**2038 1**

**2050 2**

***Note****: Invalid years containing just 0 and non numbers were ignored. They were not used for calculating the frequency of the books published each year.*

# Problem Statement 2

* Find out in which year maximum number of books were published

**Solution**

This part of the solution was solved using PIG latin scripts. The scripts are in file: MaxFrequencyYear.pig

Run the scripts using PIG. The scripts will load the output file of “Problem Statement 1”.

$ pig MaxFrequencyYear.pig

The scripts will output the year having maximum frequency into HDFS location:

**/BookMaxFreqYear/out/part-r-00000**

**2002 17628**

# Problem Statement 3

* Find out how many book were published based on ranking in the year 2002.­ ( Hint: Use Book.csv and Book-Ratings.csv)

**Solution**

This part of the problem was solved using MapReduce programs and Hive scripts. The MapReduce programs are in jar files discussed in “Problem statement 1”. The Books and BookRatings csv files are used to find the number of books published based on the ranking in the year 2002.

Books

The books csv file data was processed using MapReduce job to remove invalid years.

Book Ratings

The book ratings csv file data was processed using MapReduce job to clean ISBN’s and aggregate multiple ratings for the same ISBN.

ISBN – The ISBN’s were cleaned to remove non digits and non letters. There are invalid ISBN’s containg all-zeros and not compliant with ISBN standards (<http://en.wikipedia.org/wiki/International_Standard_Book_Number> ). These ISBN’s were still used in processing as it would not matter as the invalid ISBN will not match the ISBN in the Books csv file. The Hive JOIN statement will filter the invalid ISBN’s.

Rating/Ranking – Books are rated by multiple users. Hence there are multiple ratings for the same ISBN. For this exercise, the maximum rating aggregated using MapReduce job.

Run Java program to find the frequency of the books published each year.

$ hadoop jar PBX\_BookRatings-0.0.1-SNAPSHOT.jar /BookCrossing/BX-Book-Ratings.csv /BookRatings/out

$ hadoop jar PBX\_Books-0.0.1-SNAPSHOT.jar /BookCrossing/BX-Books.csv /Books/out

$ hive -f BooksByRating.q

At the end of Hive scripts, the number of books published based on the rating/ranking will be displayed.

**17544**

**Notes:**

(1) Books MapReduce job ignored the Year-Of-Publication equal to ‘0’. There were 4620 such records.

Standard output of Books MapReduce job.

Books - Invalid published year encounted and ignored;(ISBN,YearOfPublication)-[0752841092,0]

(2) BookRatings MapReduce job ignored the invalid BookRating containg non-digit. There was 1 such record.

Standard output of BookRatings MapReduce job.

BookRatings - Invalid ISBN encounted and ignored; (ISBN,BookRating)-[ISBN,Book-Rating]

(3) The zero rating was also used in finding the number of books published in the year 2002. If the zero ratings were ignored, the published books were 11689.

Use the following hql statements from hive shell to find the number of books published based on ranking when zero-rating is ignored:

**hive> Use bookxdb;**

**hive> SELECT COUNT(Books.Year) from BookRatings JOIN Books ON (Books.ISBN=BookRatings.ISBN) WHERE Books.Year='2002' AND BookRatins.rating !=0;**

Output

**11689**