CS 6301- Big Data Analytics and Management

Fall 2013

Homework # 2

Due Date : October 26, 2014

In this homework you will learn how to implement join (both map-side and reduce side) using Map Reduce. Please apply Hadoop map-reduce framework to derive some statistics from IMDB movie data.

Data Files : We will use the same dataset as the previous homework-1.

Resource : Map-Reduce Design Pattern – Donald Miner & Adam Shook

Q1. Given a movieID as input, Find the number of male users who has rated that movie - using map-side join.

In this problem you have will take 'movieID' as a command line input and return the number of male users who has rated this movie. For example, if 661 is the input then you have to return the total number of male users who has rated the movieID = 661. So your output will one integer. We do not want the list of male users.

You have to solve it using **map side join**. You will need the ratings.dat and users.dat to solve it. Put the smaller file(users.dat) in the memory - known as distributed cache method.

Q2. You will solve an extension of HW-1, Q-2 :

Find top 10 average rated movie names with descending order of rating - using reduce side join.

In HW-1, Q-2 : we have solved that how to get the top 10 movies with descending order of rating. But there you returned the movie ID, so only one file 'ratings.dat' was enough to get the information. But in this assignment, you have to return the movie name. So, you will also need the file 'movies.dat'. Because 'ratings.dat ' does not have the movie name in it. To get the movie name from a movie Id you have to refer to the 'movies.dat' file.

So, basically, you have to extend your hw-1, Q-2. Use your code of that to get the top 10 movieIds and then using 'movies.dat' find the names of the movies. You have to solve it using **reduce side join**.

Submission ::

You have to upload your submission via e-learning before due date. Please upload the following to eLearning:

1. Two jar files, one for each problem/ One jar file containing all solutions.
2. The Two/one jar-matching java files which have the source code.
3. ***A Readme text file about how to run your jar file. Give the command to run your jar file.