UNIVERSITY OF ILLINOIS AT CHICAGO, INFORMATION AND DECISION SCIENCES

MapReduce Programming

Suppose you have a large text file or a collection of text files with size over hundreds of GB. There are three fields in this data (separated by tab): user id, Facebook brand id, and a timestamp. Each line represents an activity record of a user on a Facebook brand. See example:

1000 brand1 ts4
1005 brand1 ts3
1000 brand2 ts1
1000 brand3 ts3
1005 brand3 ts2
1000 brand4 ts2
1005 brand4 ts1

The first column is user id (unique identifier of a user), the second column is a Facebook brand id (unique identifier of a company), and the final column is a timestamp (activity time: making comments). This log file maintains a history of personal interest transitions assuming for the same member. We define an "interest pair" to be a pair of consecutive interests one member had activities (ordered by timestamp). Suppose $ts1 < ts2 < ts3 < ts4$, then we will have the following pairs:

For 1000, we have: (brand2, brand4), (brand4, brand3), (brand3, brand1)
For 1005, we have: (brand4, brand3), (brand3, brand1)

Please write a MapReduce program to output all the interest pairs in the file and for each pair, you are expected to output the count, which is the number of unique members that had such interest transition. For the example input, the output would be

(fields are separated by tab):
(brand2, brand4) 1
(brand4, brand3) 2
(brand3, brand1) 2
.....

Once we have this file, we can build an interest transition graph where each node is a Facebook brand, the direction is the interest transition, and the weight of each edge is the number of unique members that had sub interest transition. Some further graph mining algorithms can be applied to discover interesting results.

Notes:

- The records in the file are not ordered by timestamp.
- DO NOT make any assumption about the number of members, the number of brands and the length of time range. Your code should be running on one machine and be scalable to handle any size of input.
- The code should handle very large file. That means you cannot store everything in the memory (there are millions of members and some member may have a relatively long history of the brands he/she was interested in). But you can store information on local file system (which has unlimited capacity).
- You may need multiple MapReduce jobs rather one job to finish this assignment.
- Compilers: JAVA JDK 6+, Python 2.7, Hadoop 1.0.3
- Your codes should be well commented.
- The required classes you need to write: Mapper, Reducer, Partitioner, and Driver.
- Submit a zip file including all source files, REAME file, and test cases (if any. hint: create a data generator)