Autumn 2017

Hands-on MapReduce programming exercise

Department of Computer Science
University of Helsinki

General instruction: In this exercise, you are supposed to use Hadoop to perform two table join with a large data set. We provide you the data sets. You may use Ukko cluster (or other available computing machines) to run the programs. Please read the instructions on Hadoop programing and WordCount example at https://www.cs.helsinki.fi/u/jilu/dataset/New_instructions-hadoop-programming.pdf

Grading will be based on the correctness and the running time of your programs, the quality of the program code, and associated documentation. Make sure to find all correct results and then try to reduce the running time of your programs.

1. Implement one executable Hadoop MapReduce program to perform the inner join of two tables based on "*Student ID*" to satisfy the following two filtering conditions simultaneously:
   a). The year of birth is greater than (>) 1990 and;
   b). the score of course 1 is greater than (>) 80 and that of course 2 is no more than (<=) 95.

   Sample data of Student table:

Student ID	Name	Year of Birth
20170126453	Kristalee Copperwaite	2000
20170433596	Roeberta Naden	1997

   Sample data of Score table:

Student ID	Score for course1	Score for course2	Score for course3
20170126453	93	97	80
20170140241	86	85	87
20170433596	82	60	80

   Join result:

Student ID	Name	Year of Birth	Score for course1	Score for course2	Score for course3
20170433596	Roeberta Naden	1997	82	60	80

   Download datasets: https://www.cs.helsinki.fi/u/jilu/dataset/TwotablesJoin.zip

There are two files in the unzipped folder: one *Score* tables and one *Student* table. *Student* table is a big table, but *Score* table is small with three thousand tuples. The *Reduce-side join* algorithm in the lecture may not be the most efficient one in this case. For example, please consider to use the *distributed cache* with Hadoop.
(https://hadoop.apache.org/docs/r2.4.1/api/org/apache/hadoop/filecache/DistributedCache.html).

Run your MapReduce programs and then report the result size and the performance, e.g. the elapsed time. (Hint: you may get the job information and the elapsed time through the Web interface of master node, and the default address is *hostname:8088* in Hadoop cluster)

In the documentation, you should explain how your codes solve the problems and how they use Hadoop. In particular, analyze your results by answering the following questions:

a. What are the result size of the join and the elapsed time of your programs?
b. How many computer nodes did you use to run the program? Have you tried to reduce the running time by using more nodes in Ukko cluster?
c. Upload your source codes and analyze the performance of your program codes. What optimizations have you tried to reduce the running time?

Return: Store all the files in a directory and zip this directory, name the zip-file "*username_studentID_Hadoop.zip*", and return the zip-files. Please indicate clearly your name and student ID in every source code file as well.

Ethics and plagiarism

This exercise is individual work. You can discuss any problems you encounter with other classmates and teacher, but sharing code and documentation is not allowed and if found, will be considered as plagiarism.

Reference for your programming

- Miner D, Shook A. MapReduce Design Patterns: Building Effective Algorithms and Analytics for Hadoop and Other Systems [M]. " O'Reilly Media, Inc.", 2012.
- Lam C. Hadoop in action [M]. Manning Publications Co., 2010.
- Configure memory of cluster: https://hortonworks.com/blog/how-to-plan-and-configure-yarn-in-hdp-2-0/.