Project for Data Science

Project Description:

You are asked to design **a MapReduce based algorithm** (you can pick any machine learning algorithm you like) which can handle huge datasets using a large number of computers. In addition, you should provide proper data analysis for your algorithm. Questions you need to answer in your report such as:

1. What is your data processing pipeline? (graphs and words description)
2. What kind of analytics do you apply on the dataset?

Project Submission:

Your project submission should include your code and a project report.

Grading guidelines:

The evaluation will be based on two parts:

Project proposal ~5%

Project progress report ~5%

Project Presentation ~10%

Project source code ~20%

Project Report ~20%

Late submission policy:

Late submission will not be accepted after the due date.

MapReduce

MapReduce is a framework for processing parallelizable problems across huge datasets using a large number of computers.

Map step: The master node takes the input, divides it into smaller sub-problems, and distributes them to worker nodes. The worker node processes the smaller problem, and passes the answer back to its master node.

Reduce step: The master node then collects the answers to all the sub-problems and combines them in some way to form the output the answer to the problem it was originally trying to solve.

MapReduce allows for distributed processing of the map and reduction operations. Provided that each mapping operation is independent of the others, all maps can be performed in parallel. Similarly, a set of reducers can perform the reduction phase, provided that all outputs of the map operation that share the same key are presented to the same reducer at the same time, or that the reduction function is associative. Another way to look at MapReduce is as a 3-step parallel and distributed computation:

$< K1, V1 > -map- < K2, V2 > -shuffle- < K2, alistof V2 > -reduce- < K3, V3 > (2)$

1. Prepare the Map() input: the MapReduce system designates Map processors, assigns the K1 input key value each processor would work on, and provides that processor with all the input data associated with that key value. Run the user-provided Map() code Map() is run exactly once for each K1 key value, generating output organized by key values K2.

2. Shuffle the Map output to the Reduce processors: the MapReduce system designates Reduce processors, assigns the K2 key value each processor would work on, and provides that processor with all the Map-generated data associated with that key value.

3. Run the user-provided Reduce() code: Reduce() is run exactly once for each K2 key value produced by the Map step. Produce the final output the MapReduce system collects all the Reduce output, and sorts it by K2 to produce the final outcome.

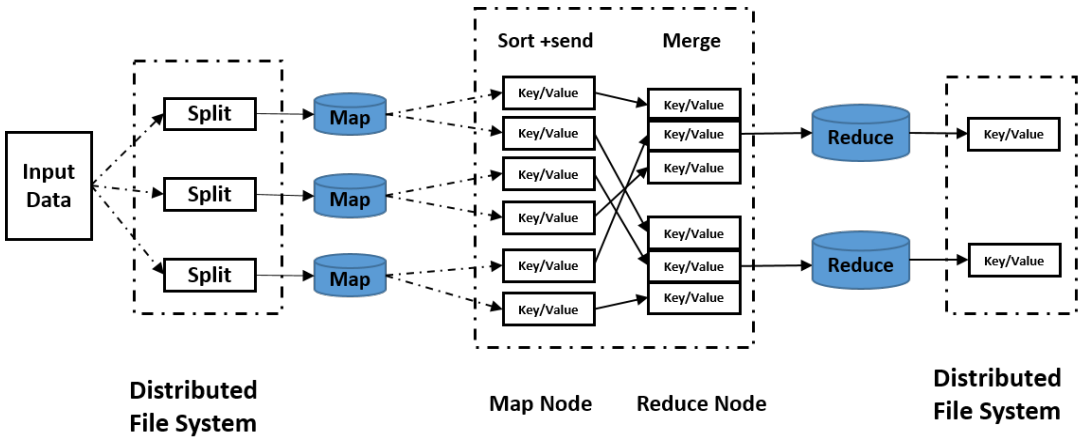

Figure 1: A general framework of a MapReduce system.

Amazon Elastic MapReduce

Amazon Elastic MapReduce (Amazon EMR) is a web service that makes it easy to quickly and cost-effectively process vast amounts of data. Amazon EMR uses Hadoop, an open source framework, to distribute your data and processing across a resizable cluster of Amazon EC2 instances.

Useful Links:

https://www.kaggle.com/learn/overview

https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-common/SingleCluster.html

https://docs.aws.amazon.com/emr/latest/ManagementGuide/emr-gs.html