Assignment 2 - Map Reduce

.

**Introduction**

MapReduce is a framework using which we can write applications to process huge amounts of data, in parallel, on large clusters of commodity hardware in a reliable manner. It is a processing technique and a program model for distributed computing. The MapReduce algorithm contains two important tasks, namely Map and Reduce. Map takes a set of data and converts it into another set of data, where individual elements are broken down into tuples (key/value pairs). Secondly, reduce task, which takes the output from a map as an input and combines those data tuples into a smaller set of tuples. As the sequence of the name MapReduce implies, the reduce task is always performed after the map job. The master serves as a single point of contact for mappers and reducers, as well as orchestrates the architecture.

**Architecture**

**Master**

* The master is in charge of all communication and coordination in the system. It ensure the synchronization of the different elements within the system. It is also responsible for dividing the input data into equal parts based on the number of mappers and also allocates every chunk to every distinct mapper
* It is through the configuration file that inputs are passed to the mapper.
* Based on the mapper count, a number of mappers are spawned with their own distinct addresses and port numbers, and a call is made to each of them by the master.
* After mappers have finished processing, the master starts the reducers. It also acts as an obstacle, preventing further processing from beginning until all mappers have completed their work. This ensures that incomplete data isn’t being sent to the reduced otherwise we will get incorrect output because of less data.
* The results are stored to the key value store only after all reducers have confirmed that the operational processing is complete from their end. This is done with the help of an acknowledgement to the user.

**Mapper**

* Every mapper has an associated function assigned to it. Based on the input from the master, it runs the functions and maps the data
* It generates an output after running the function. Post that, it writes the output to the reducer while also indicating the process number. It eventually informs the master with the help of an acknowledgement that it’s process has been completed

**Reducer**

* The role of the reducer is to reduce the data based on its function and then with the help of files, store the data in it and send back an acknowledgement to the master

**Code Running Process**

**Steps**

* ssh-keygen (for generating the key)
* cat ~/.ssh/id\_rsa.pub >> ~/.ssh/authorized\_keys (for placing the public key in the authorized\_keys file)
* ssh localhost (to check whether you can ssh into the system)
* python3 main\_driver.py config.json

**Future Improvements**

* There is a scope for improvement in the fault tolerance of the system since mappers can crash sometimes due to overload.
* The whole system is dependent on the master and if the master crashes at any point in time, the whole system will crash as well.
* Currently, there is no mechanism to handle multiple files.