BLM4821
Big Data Processing and Analysis Course
Term Project
Spring 2021

Ayşe Hilal Doğan – 17011907
Betül Ön – 17011611

General information about application

This is a Multi-Node hadoop project for storing big data using mapreduce. MapReduce is a programming model and an associated implementation for processing and generating big data sets with a parallel, distributed algorithm on a cluster.

We used Docker and Ubuntu for creating a Multi-Node system. There are 3 slave nodes to store the data. Our data is a .csv file and 646.3 MB. Data has a csv file with 23 features.

The project copies the dataset from local system to HDFS and then do the mapreduce work in HDFS. After calculating the choosen descriptive statistics function result, a result file is created in HDFS and copies the file to local system.

Use case scenario

Hadoop is used for storing and processing big data. In Hadoop, data is stored on inexpensive commodity servers that run as clusters. It is a distributed file system that allows concurrent processing and fault tolerance. Hadoop MapReduce programming model is used for faster storage and retrieval of data from its nodes.

MapReduce is a programming model for processing large data sets with a parallel , distributed algorithm on a cluster. Map Reduce when coupled with HDFS can be used to handle big data. Semantically, the map and shuffle phases distribute the data, and the reduce phase performs the computation.

Technical challenges

First we installed Hadoop 3.2.1 in VirtualBoxVM. There were so many technical challenges while setting up Multi-Node Cluster (2 nodes) on Ubuntu 18 then we decided to try Ubuntu 20. After that we successfully installed multinode. But when it comes to running the jar file we got so many errors and we couldn't solve it so we installed Docker. In Docker we successfully set up Multi-Node Cluster and we used the jar files that we created before in VirtualBoxVM. We did not have a problem with Docker as in VirtualBox.

First we closed one of the nodes and used two nodes. We have seen that the overall performance increased. When we use three nodes it takes longer than two nodes.

Explanation of implementation

We used Docker with Ubuntu 20.04 for implementing a Multi-Node cluster with 3 nodes. We created an input file which contains the dataset in local system and copied to HDFS file system. In HDFS system mapreduce work is done. After mapreduce work, it puts the calculation result to a file and then copies this file to local system.

In this Java Project file we have a package and we have java classes that implement 5 different descriptive statistics function by using map reduce programming model.

1. Count

2. Min-max

3. Range

4. Average

5. Standard deviation

Count is a simple application that calculates the frequency of words in the given text.
Min-Max function is used for numerical values and gives the minimum and maximum.
Range is a function that calculates the difference between minimum and maximum values.
Average is a function to calculate the average value of the dataset values.
Standard deviation is a function to calculate the standard deviation of the dataset values.

Performance Evaluation

	646.3 MB
Average	**123**
Min Max	**115**
Standard Deviation	**103**
Word Count	**100**
Range	**116**

Experience and Discussion

We tried using Virtual Box first but it gave lots of errors and took too much time. Then we used Docker and it was faster than using VirtualBox to set Multi-Nodes. We have learned the methodology of map reduce and how to implement map reduce with different calculations.

MapReduce is a programming model for processing large data sets with a parallel , distributed algorithm on a cluster. Map Reduce when coupled with HDFS can be used to handle big data. Semantically, the map and shuffle phases distribute the data, and the reduce phase performs the computation.

The performances of descriptive statistics functions are very close to each other as we can see in the table above. To see the performance difference of two nodes and three nodes, first we closed one of the nodes and used two nodes. We have seen that the overall performance increased. When we use three nodes it takes longer than two nodes.