Table of Contents

1. Introduction

This document contains the details related to sorting of 10GB and 100GB datasets using Hadoop and Spark. The following sections will have elaborate details with respect to each of the experiment.

2. Design of experiments

This section outlines the details on the approach taken for the logic for each of experiments.

2.1 Hadoop

The aim of the benchmark experiment is to sort a 10GB dataset by running Hadoop on a single node and to sort 100GB dataset by running Hadoop on 16 node cluster.

The experiment starts with configuring the Hadoop Configuration files, starting the HDFS and YARN daemons, and finally running a Hadoop Map Reduce sorting algorithm. The configuration files vary for one node and 16-node cluster. The Hadoop Map Reduce sorting algorithm is developed using JAVA.

The Hadoop experiments are run in the US-EAST-1 (N.Virginia) region.

2.2 Spark

The aim of the benchmark experiment is to sort a 10GB dataset by running Hadoop on a single node and to sort 100GB dataset by running Hadoop on 16 node cluster.

The experiment starts with starting a spark cluster using the '**spark-ec2**' script provided in the Spark's ec2 directory. Single node cluster involves creating a cluster of 1 master and 1 slave node and a 16-node cluster includes 1 master and 16 slave nodes. Once the cluster is up and running, sorting algorithm is run. The sorting algorithm is developed using SCALA.

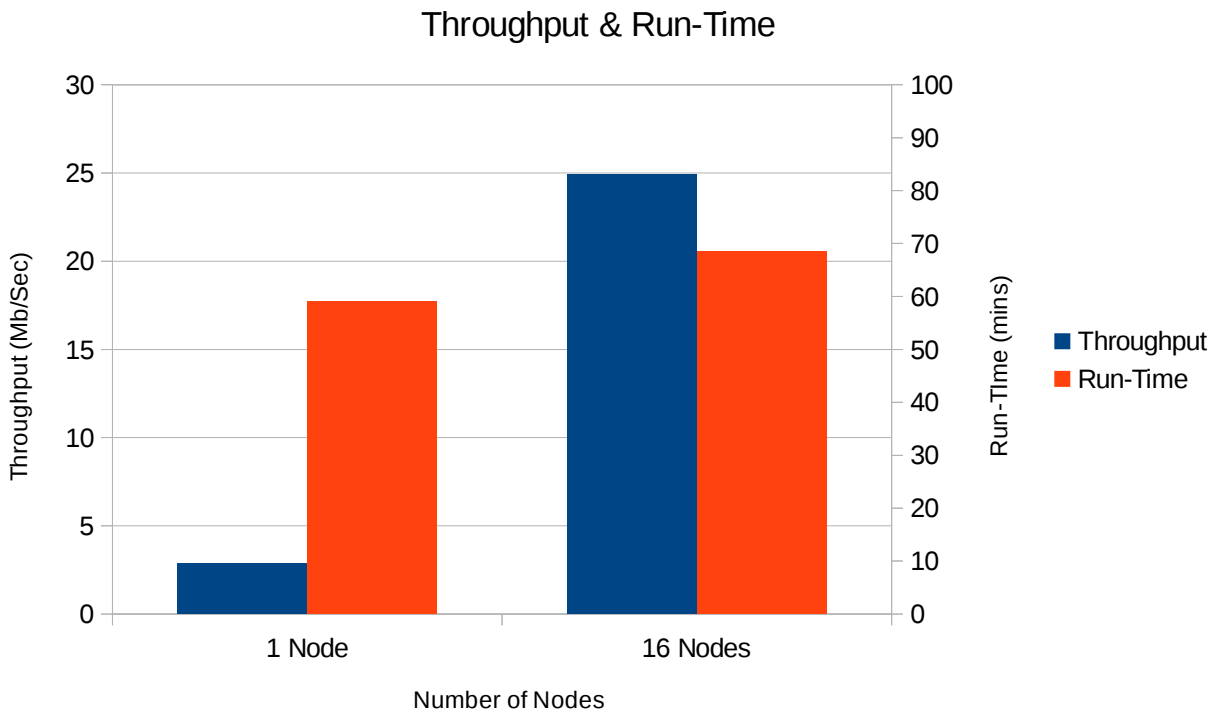
The Spark experiments are run in the US-EAST-1 (N.Virginia) region.

3. Performance Analysis

3.1 Hadoop

Below are the results of Hadoop Experiments.

No. of Nodes	Dataset Size (GB)	Execution time (sec)	Throughput (MB/sec)
1	10	3542	2.891
16	100	4113	24.8967



3.2 Spark

4. Hadoop Cluster Setup

This section further answers the questions given in the assignment.

- **What is Master Node? What is a Slave Node?**

Master Node is responsible for storing data(Name Node) and running computations on the data (Job Tracker). The slave node is responsible for running the computations on the data (Data Node) and communicating with the master node (Task Tracker).

- **Why do we need to set unique available ports to those configuration files on a shared environment? What errors or side-effects will show if we use same port number for each user?**

Each and every port is assigned to different functionalities, for example port number 8020 is for Namenode functionality. Port conflict will happen if we use the same port number and the program might produce adverse results.

- **How can we change the number of mappers and reducers from the configuration file?**

The number of mappers and reducers can be modified by changing the '**mapred-site.xml**' file. The parameter '**mapred.map.tasks**' represents the default number of map tasks per job and the parameter '**mapred.reduce.tasks**' represents the default number of reduce tasks per job.

Below lists out the changes made to different configuration files for changing the cluster from 1 node to 16 nodes.

- **Masters**

This files consists of the list of nodes that will run secondary namenodes.

- **Slaves**

This files consists of the list of nodes that will run datanode and task-tracker. Essentially, this file consists of list of nodes that do all the computations on the data.

- **Core-site.xml**

This file specifies the parameters related to Hadoop core. Parameters such as host of the Namenode, location of the hadoop temporary directory etc. are present in this file.

- **Hdfs-site.xml**

This file consists of parameters related to Hadoop file system. Examples, directory

location where the Name node should store the name table.

- **mapred-site.xml**

This file consists of parameters related to Map Reduce daemons. Job tracker address, maximum number of map-reduce tasks that are run simultaneously on a task tracker are some of the parameters.

5. References

- https://hadoop.apache.org/docs/r1.0.4/mapred-default.html
- https://wiki.apache.org/hadoop/HowManyMapsAndReduces
- https://hadoop.apache.org/docs/r1.2.1/cluster_setup.html#Configuration+Files