Name : Bhavani Rajpurohit

Class : AIA-3 Subject: DBMS LAB

Roll No : 2213688 Batch : B

ASSIGNMENT NO: 10

Aim: Hadoop and HBase installation on single node.

Software required:

1. Ubuntu 18 / 18

2. Hadoop 3.0.0 and above

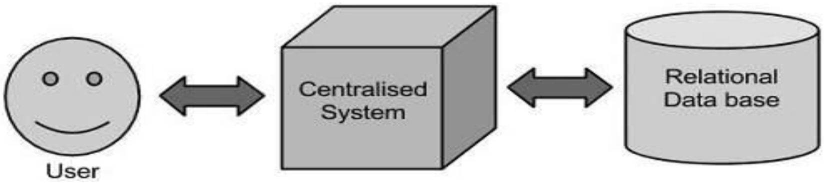
Theory:

Hadoop

Hadoop is an open source software framework written in Java for distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware. All the modules in Hadoop are designed with a fundamental assumption that hardware failures (of individual machines or racks of machines) are common and thus should be automatically handled in software by the framework.

Traditional Approach

In this approach, an enterprise will have a computer to store and process big data. Here data will be stored in an RDBMS like Oracle Database, MS SQL Server or DB2 and sophisticated software can be written to interact with the database, process the required data and present it to the users for analysis purpose.

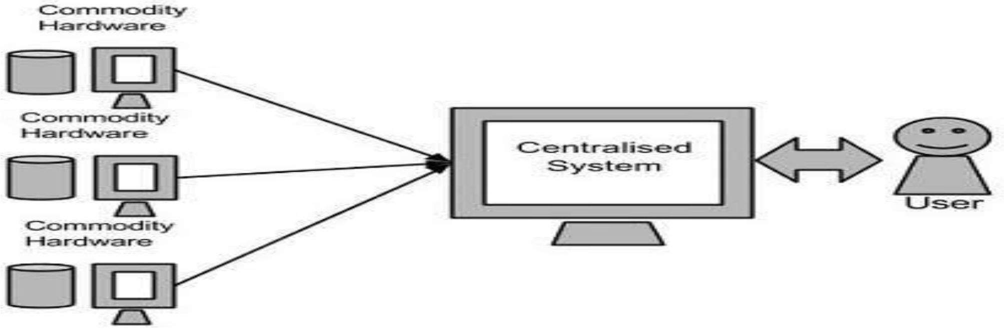


Limitation

This approach works well where we have less volume of data that can be accommodated by standard database servers, or up to the limit of the processor which is processing the data. But when it comes to dealing with huge amounts of data, it is really a tedious task to process such data through a traditional database server.

Google’s Solution

Google solved this problem using an algorithm called MapReduce. This algorithm divides the task into small parts and assigns those parts to many computers connected over the network, and collects the results to form the final result dataset.

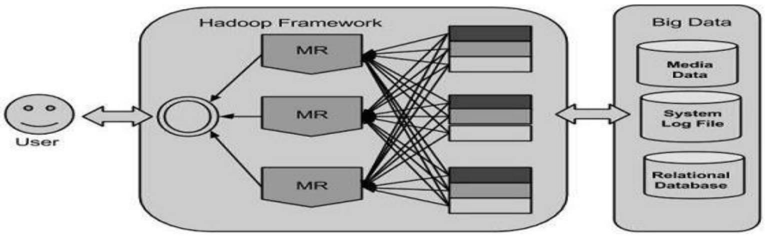


Above diagram shows various commodity hardware's which could be single CPU machines or servers with higher capacity.

Hadoop

Doug Cutting, Mike Cafarella and team took the solution provided by Google and started an Open Source Project called HADOOP in 2005 and Doug named it after his son's toy elephant. Now Apache Hadoop is a registered trademark of the Apache Software Foundation.

Hadoop runs applications using the MapReduce algorithm, where the data is processed in parallel on different CPU nodes. In short, Hadoop framework is capable enough to develop applications capable of running on clusters of computers and they could perform complete statistical analysis for huge amounts of data.



Hadoop Architecture

Hadoop framework includes following four modules:

Hadoop Common: These are Java libraries and utilities required by other Hadoop modules. These libraries provides filesystem and OS level abstractions and contains the necessary Java files and scripts required to start Hadoop.

Hadoop YARN: This is a framework for job scheduling and cluster resource management.

Hadoop Distributed File System (HDFS): A distributed file system that provides high-throughput access to application data.

Hadoop MapReduce: This is YARN-based system for parallel processing of large data sets.

MapReduce

Hadoop **MapReduce** is a software framework for easily writing applications which process big amounts of data in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.

The term MapReduce actually refers to the following two different tasks that Hadoop programs perform:

- **The Map Task:** This is the first task, which takes input data and converts it into a set of data, where individual elements are broken down into tuples (key/value pairs).
- **The Reduce Task:** This task takes the output from a map task as input and combines those data tuples into a smaller set of tuples. The reduce task is always performed after the map task.

Typically both the input and the output are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks.

The MapReduce framework consists of a single master JobTracker and one slave TaskTracker per cluster-node. The master is responsible for resource management, tracking resource consumption/availability and scheduling the jobs component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves TaskTracker execute the tasks as directed by the master and provide task-status information to the master periodically.

The JobTracker is a single point of failure for the Hadoop MapReduce service which means if JobTracker goes down, all running jobs are halted.

Hadoop Distributed File System

Hadoop can work directly with any mountable distributed file system such as Local FS, HFTP FS, S3 FS, and others, but the most common file system used by Hadoop is the Hadoop Distributed File System (HDFS).

The Hadoop Distributed File System (HDFS) is based on the Google File System (GFS) and provides a distributed file system that is designed to run on large clusters (thousands of computers) of small computer machines in a reliable, fault-tolerant manner.

HDFS uses a master/slave architecture where master consists of a single **NameNode** that manages the file system metadata and one or more slave **DataNodes** that store the actual data.

How Does Hadoop Work?

Stage 1

A user/application can submit a job to the Hadoop (a hadoop job client) for required process by specifying the following items:

1. The location of the input and output files in the distributed file system.
2. The java classes in the form of jar file containing the implementation of map and reduce functions.
3. The job configuration by setting different parameters specific to the job.

Stage 2

The Hadoop job client then submits the job (jar/executable etc) and configuration to the JobTracker which then assumes the responsibility of distributing the software/configuration to the slaves, scheduling tasks and monitoring them, providing status and diagnostic information to the job-client.

Stage 3

The TaskTrackers on different nodes execute the task as per MapReduce implementation and output of the reduce function is stored into the output files on the file system.

Advantages of Hadoop

- Hadoop framework allows the user to quickly write and test distributed systems. It is efficient, and it automatic distributes the data and work across the machines and in turn, utilizes the underlying parallelism of the CPU cores.
- Hadoop does not rely on hardware to provide fault-tolerance and high availability (FTHA), rather Hadoop library itself has been designed to detect and handle failures at the application layer.

a. Single Node:

Steps for Compilation & Execution

- sudo apt-get update
- sudo apt-get install openjdk-8-jre-headless
- sudo apt-get install openjdk-8-jdk
- sudo apt-get install ssh
- sudo apt-get install rsync

# **Download hadoop from:**

https://archive.apache.org/dist/hadoop/common/hadoop-3.0.0/ hadoop-3.0.0.tar.gz

- # copy and extract hadoop-3.0.0.tar.gz in home folder
- # rename the name of the extracted folder from hadoop-3.0.0 to hadoop
- readlink -f /usr/bin/javac
- gedit ~/hadoop/etc/hadoop/hadoop-env.sh
- # add following line in it
- **# for 32 bit ubuntu**
- export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-i386
- **# for 64 bit ubuntu**
- export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
- # save and exit the file

- # to display the usage documentation for the hadoop script try next command
- ~/hadoop/bin/hadoop

#Setup passphraseless/passwordless ssh

- ssh-keygen -t dsa -P '' -f ~/.ssh/id_dsa
- cat ~/.ssh/id_dsa.pub >> ~/.ssh/authorized_keys
- export HADOOP_PREFIX=/home/**your_user_name**/hadoop
- ssh localhost

# type **exit** in the terminal to close the ssh connection (very important)

Exit

# The following instructions are to run a MapReduce job locally.

- **Format the filesystem:(Do it only once)**

~/hadoop/bin/hdfs namenode -format

- **Start NameNode daemon and DataNode daemon:**

~/hadoop/sbin/start-dfs.sh

- **Browse the web interface for the NameNode; by default it is available at:**

http://localhost:50070/

Conclusion:

In this way the Hadoop was installed & configured on Ubuntu for BigData.