International Journal of Engineering Inventions
e-ISSN: 2278-7461, p-ISSN: 2319-6491
Volume 4, Issue 6 (November 2014) PP: 09-18

Hadoop Job Runner UI Tool

Deepu Sojan[1], Mr. Sujith kumar P. S.[2]
[1]*M.Tech Computer Science and Engineering, Sree Narayana Gurukulam College of Engineering, Kadayirippu Ernakulam, Kerala*
[2]*Asst. Prof, Dept of Computer Science and Engineering, Sree Narayana Gurukulam College of Engineering, Kadayirippu Ernakulam, Kerala*

Abstract: *To do research on Hadoop in virtual environment, an experimental environment is needed. This paper firstly introduces some technologies used such as Distributed environment and Map Reduce. Based on that, a method to deploy Distributed environment is given. Then we discuss how to deploy Hadoop in virtual machines which can be obtained from network by some means, then an algorithm to solve the problem that all the virtual machines which are created by distributed environment. After that we run some Hadoop programs under the cluster. Important part of this project is focussed on the Hadoop Job Runner UI Tool. This application can perform all most every job that's normally works in command mode. Basic concepts behind this application are makes Hadoop job running easy. You can browse the input data from local. Application Copies the input file to HDFS. Copies the selected class to be executed to remote server. Run the job. Copy the result from HDFS to local. Display the Result and job statistics once completed.*

I. Introduction

When computer was just invented, data and compute resources were centralized, computer users used terminal to access them. And with the development of hardware, personal computer comes into our life. But now it shows a trend that data and compute resources are centralized again which called cloud computing. Every moment services emit large amounts of data, which brings a problem: how to deal with the immense data set. Search engine leader Google uses a programming model called MapReduce can process 20PB data per day. Hadoop is an open source implementation of Map Reduce, which is sponsored by Yahoo. As free and open source software, Hadoop is developing fast; most recently its first stable version is released.. Not only researchers but also enterprises are using Hadoop.The map-reduce programming model enables large datasets to be divided and assigned to available computer nodes where the data can be processed locally, avoiding network-transfer delays.

Apache's Hadoop is an open source implementation of Google's Map/Reduce framework. It enables distributed, data intensive and parallel applications by decomposing a massive job into smaller tasks and a massive data set into smaller partitions such that each task processes a different partition in parallel.The main abstractions are MAP tasks that process the partitions of a data set using key/value pairs to generate a set of intermediate results and REDUCE tasks that merge all intermediate values associated with the same intermediate key. Hadoop uses Hadoop Distributed File System (HDFS) which is an implementation of the Google File System (GFS) for storing data.HDFS cluster has master/slave architecture with a single Name Node as the master server which manages the file system namespace and regulates access to files by clients. The slaves are a number of Data Node.

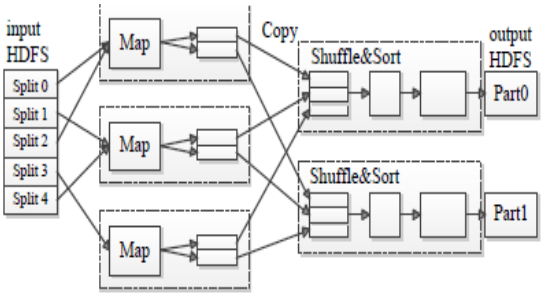


Fig 1.1 Mapreduce Architecture

GUI- Important part of this project is focussed on the Hadoop Job Runner UI Tool. This application can perform all most every job that's normally works in command mode. Basic concepts behind this application are makes Hadoop job running easy. You can browse the input data from local. Application Copies the input file to HDFS. Copies the selected class to be executed to remote server. Run the job. Copy the result from HDFS to local. Display the Result and job statistics once completed.

II. Background Theory

In general the big data analysis involves:

1. Large amount of unstructured data.
2. Distributed environment (Resource sharing environment).
3. MapReduce algorithms

2.1. Large amount of unstructured data.

Today the term big data draws a lot of attention, but behind the hype there's a simple story. For decades, companies have been making business decisions based on transactional data stored in relational databases. Beyond that critical data, however, is a potential treasure trove of non-traditional, less structured data: weblogs, social media, email, sensors, and photographs that can be mined for useful information. Decreases in the cost of both storage and compute power have made it feasible to collect this data -which would have been thrown away only a few years ago. As a result, more and more companies are looking to include non-traditional yet potentially very valuable data with their traditional enterprise data in their business intelligence analysis. To derive real business value from big data, you need the right tools to capture and organize a wide variety of data types from different sources, and to be able to easily analyze it within the context of all your enterprise data. Oracle offers the broadest and most integrated portfolio of products to help you acquire and organize these diverse data types and analyzes them alongside your existing data to find new insights and capitalize on hidden relationships. With the recent introduction of Oracle Big Data Appliance and Oracle Big Data Connectors.

2.2. Distributed Environment.

In recent years, there has been an increasing interest in running Hadoop clusters and analysis programs in the "cloud". This implies starting a Hadoop cluster with a remote shared infrastructure to conduct a data analysis task on demand. The most common example of cloud computing is Amazon's EC2 web service that allows developers to run virtual Hadoop clusters, paying only for the computing they use. In this model, a user first requests a set of computing nodes from a remote system. Then a Hadoop cluster instance can be started directly or through loading prebuilt virtual machine images to carry out data analysis tasks. CloudStack is an open source software platform that pools computing resources to build public, private, and hybrid Infrastructure a Service clouds.CloudStack manages the network, storage, and compute nodes that make up a cloud infrastructure. CloudStack can be used to deploy, manage, and configure cloud computing environments .Set up an on-demand, elastic cloud computing service. Service providers can sell self-service virtual machine instances, storage volumes, and networking configurations over the Internet. Set up an on-premise private cloud for use by employees. Rather than managing virtual machines in the same way as physical machines, with Cloud Stack an enterprise can offer self-service virtual machines to users without involving IT departments. Cloud Computing provides large scale computing capability and data storage capacity as services by organizing massive machines and clusters. MapReduce becomes one of the de-facto standards for processing large data sets in cloud computing environment. It is a distributed programming model for parallel processing on TB level data sets on a large cluster consisting of thousands of machines. One of the most successful applications of the MapReduce model is the index of the Google's search engine.

2.3. MapReduce.

MapReduce is a parallel data processing paradigm targeted at cluster-based computing architectures. Its advantage is that it allows programmers to abstract from the issues of parallelization, scheduling, input partitioning, failover, replication and focus on designing their application data flows consisting of filtering and aggregation steps. The MapReduce programming model consists of encoding data processing in terms of two functions: Map and Reduce. Input data is partitioned into fixed sized blocks and fed into parallel Map tasks which process the data chunks and produce intermediate output as a collection of key-value pair tuples. These tuples are shuffled across different reduce nodes based on key values. Each Reduce task performs three steps: copy - the map output is copied to reducer nodes, sort - the collected map output is sorted based on key values and reduce - reduce function e.g. aggregation is applied to the data. Various efforts like Hive, Pig offer friendlier interfaces in the form of high level query or dataflow languages as SQL. This enables users to encode their tasks in terms of query operators that are automatically compiled into Hadoop jobs (MapReduce workflows) instead of as low-level Map and Reduce functions. The name MapReduce comes from the two kinds operations in functional programming language: Map and Reduce. In functional programming language, function has no side-effect, which means that programs written by functional programming language can be more optimized in parallel programming. In functional ogramming language, Map and Reduce take functions as parameters, which are fully used in MapReduce.

In conventional programming, the data is copying to the location where the actual job is running. But in Hadoop Map Reduce, the job is copied to the location, where the actual data resides. Hadoop uses HDFS as its primary data storage area. The intermediate output produced by the map worker node is a collection of key-

value pair tuples. During the Reduce phase, each reduce worker node gets the intermediate output by using the key and sort them by checking the values and aggregate them by using a reduce function. A scheduling policy is used in Hadoop when the number of map and reduce task slots requested by the jobs exceeds the limit since each worker node has a fixed number of the map and reduce task slots that determine how many map and reduce tasks it can run at the same time.

FIFO is a simple job scheduling policy used as a default option by Hadoop, which allows the job submitted earlier has preference over those jobs submitted later. Besides FIFO, Capacity Scheduler is developed by Yahoo, which restricts the number of slots occupied by the jobs requested by the same user and addresses a fair allocation of slots amongst users, especially in a large number of users.

III. Proposed Method

3.1 HADOOP ARCHITECTURE

Big enterprises are moving away from the traditional way of setting up the analytical databases in high end proprietary machines and moving towards cheaper commodity hardware and they need to analyze terabytes and peta bytes of data every day. There are arguments favoring two technologies for data analysis in these cases. The proponents of parallel databases believe that the performance and efficiency of parallel databases make them well suited for analytical processing.

The other argument is that Map/Reduce based systems are more suitable because of their scalability, fault tolerance and flexibility to handle unstructured data. Both approaches have advantages as well as limitations. Parallel databases are not so good in fault tolerance and working with heterogeneous environment properties. In case of trade-off between fault tolerance and performance, parallel database selects performance extreme. The limitation with Map/Reduce is performance. In Map/Reduce, the user has to first model and load data before processing and this makes limited utilization of performance enhancing tools. Moreover, the traditional analytical data processing queries and reports are not well suited for one time query processing of Map/Reduce. HadoopDB is a hybrid system that has the best features of both parallel databases and Map/Reduce. HadoopDB approaches parallel database in performance and efficiency.

MapReduce's execution model has two phases, map and reduce both requiring moving data across nodes. Map tasks can be run in parallel and are distributed across nodes available. The input data, that is stored in HDFS prior to beginning of the job, is divided into datasets or splits, and for each split a map task is assigned to TaskTracker on a node that is close to that split. The map computation begins when its input split is fetched by the TaskTasker, which processes the input data according to the map function provided by the programmer, generates intermediate data in the form of <key,value> tuples, and partitions them for the number of reduce tasks.

In the reduce phase, the TaskTrackers assigned to do the reduce task fetch the intermediate data from the map TaskTrackers, merge the different partitions, perform reduce computation, and store the results back into HDFS.

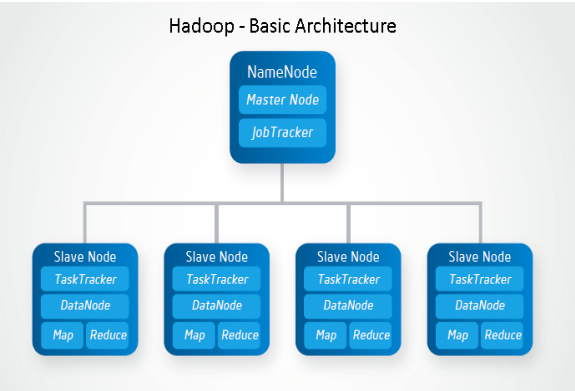


Fig3.1.1: Hadoop basic architecture

It explores the effect of network on Hadoop performance. The work identifies the bursty nature of Hadoop traffic, the overall network speed and Hadoop's design for reliability as some of the factors affecting Hadoop performance.

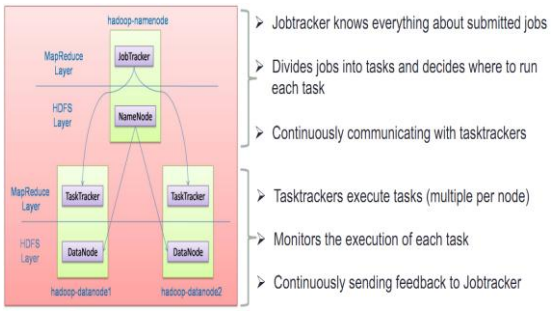


Fig3.1.2: Hadoop's job assigning method

Many industry and research efforts are addressing these issues. Explores the use of Infinite band interconnect in Hadoop clusters. Presents a new pipeline design that overlaps the shuffle, merge and reduce phases, along with an alternative protocol to enable data movement via RDMA (Remote Direct Memory Access). Instead, we focused on utilizing the network control capabilities provided by OpenFlow to provide resources to straggling Reducers, who can be speeded up by prioritizing their use of the network. Network traffic in Hadoop can be separated into several.

3.2 HDFS ARCHITECTURE

HDFS - **H**adoop **D**istributed **F**ile **S**ystem

- Primary storage system for Hadoop
- Distributed, portable, scalable file system written in Java
- Files are broken down and stored in multiple machines
- Designed for large scale distributed data processing
- Follows master/slave architecture

HDFS consists of:

- NameNode server to host the filesystem index
- DataNodes, where the data blocks are stored
- Secondary NameNode takes snapshots of NameNode's memory structure

The Hadoop Distributed File System (HDFS) is a distributed file system designed to run on commodityhardware. Even though there are many similarities with existing distributed file systems, they are very much different.HDFS has a high degree of fault-tolerance and is developed to be deployed on low-cost hardware.

HDFS provides efficient access to application data and is suitable for applications having big data sets. HDFS cluster has master/slave architecture with a single Name Node as the master server which manages the file system namespace and regulates access to files by clients. The slaves are a number of Data Nodes, usually one per node in the cluster, which manage storage attached to the nodes that they run on. HDFS provides a file system namespace and allows user data to be stored in files. A file is partitioned into one or more blocks and these blocks are stored in a set of Data Nodes. The file system namespace operations like opening, closing, and renaming of files and directories are done by the Name Node.

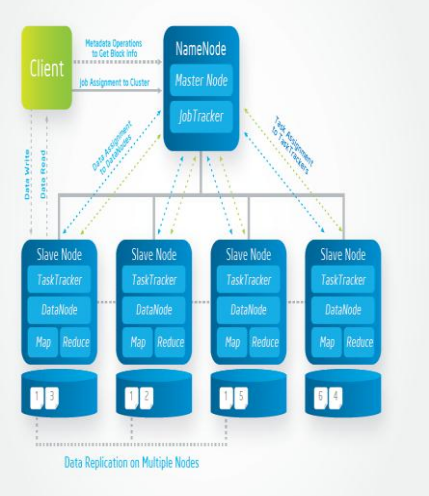


Fig3.2.1: HDFS architecture

The Data Nodes are responsible for serving read and write requests from the file system's clients, block creation, replication and deletion upon instructions from the Name Node.A large number of extensions to the existing Hadoop infrastructure have been proposed. Here some of the major extensions mainly focusing different data layouts, replication etc. Big enterprises are moving away from the traditional way of setting up the analytical databases in high end proprietary machines and moving towards cheaper commodity hardware and they need to analyze terabytes and peta bytes of data every day. There are arguments favoring two technologies for data analysis in these cases. The proponents of parallel databases believe that the performance and efficiency of parallel databases make them well suited for analytical processing.

In case of trade-off between fault tolerance and performance, parallel database selects performance extreme. The limitation with Map/Reduce is performance. In Map/Reduce, the user has to first model. And load data before processing and this makes limited utilization of performance enhancing tools. Moreover, the traditional analytical data processing queries and reports are not well suited for one time query processing of Map/Reduce. Recent research on scale-down in GFS and HDFS managed clusters propose maintaining a primary replica of the data on a small covering subset of nodes that are guaranteed to be on. However, these solutions suffer from degraded write-performance as they rely on write-offloading technique to avoid server wakeups at the time of writes.

HDFS (Hadoop Distributed File System) is a distributed file system that stores and manages the data in a Hadoop cluster. there is central node called NameNode to store the meta-data of the file system and other nodes called DataNodes that store the data. Files in HDFS are split into smaller blocks, typically 64MB, and each block is stored separately at a DataNode and replicated as per the specified replication factor to provide data durability. A HDFS client contacts the NameNode to get the location of each block, and then interacts with DataNodes responsible for the data. Hadoop MapReduce Framework consists of two types of components that control the job execution process: a central component called the JobTracker and a number of distributed components called TaskTrackers. Based on the location of a job's input data, the JobTracker schedules tasks to run on some TaskTrackers and coordinates their execution of the job. TaskTrackers run tasks assigned to them and send progress reports to the jobtracker.

Copying the remote data to HDFS is consists of two stages. First stage will copy the data to a Hadoop edge Node using any of the file transfer protocols such as ftp or sftp. Then in the second stage, the data is moved into HDFS using Hadoop copy From Local or put command utility. Another solution is copying the data directly from remote machine to HDFS using the Hadoop file system API. In both methods, only a single file channel is opened from source file system to HDFS for copying the data. The performance of file copying can be improved, if there are multiple channels opened from the source to HDFS by creating a map-reduce program for the second approach.

IV. Graphical User Interface

It Makes Hadoop job running easy. You can browse the input data from local. Application Copies the input file to HDFS. Copies the selected class to be executed to remote server. Run the job. Copy the result from HDFS to local. Display the Result and job statistics once completed. The UI tool internally takes care of all the manual steps to be done in order to execute a Hadoop job. You can connect to any cluster running. Just provide the IP of any node hosting Hadoop and its credentials. It will Retrieve the job Statistics and Result. No need to copy the files manually to a remote server by scp command from terminal in order to put it to HDFS to get it executed.

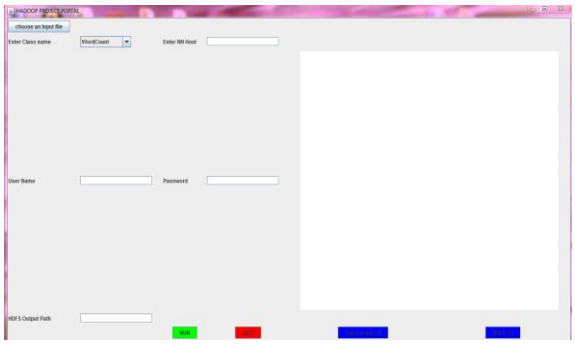


Fig 4.1:Hadoop job runner ui tool

This tool will help you to browse the file system to select the file. Copy the input file to the provided remote server. Once the input file is available in the local file system of the remote server, to run a Hadoop job this file should be available in HDFS.The tool Internally copies the input file from the text file system to HDFS Hence manual command execution of Hadoop fs –put <file name> is avoided. The tool copies the selected class

to be executed to the remote server internally. Hence another overhead is avoided. The tool internally runs the job and generates the output. Provides the map reduce outputs and save it to HDFS .It saves the job statistics and sends it across. To copy the result to the local server from the remote server, the result should be available in the text file system. The tool internally copies the result from HDFS to local file system of remote server. Once the Result and job statistics are available in local file system of remote server, the tool copies it to the local server. And it displays the result and statistics in word pad editor. Currently the tool is designed such a way that it can run both on Linux and Windows which has java installed. From any server to run a Hadoop job, you need not have Hadoop installed on it. Any number of Map-Reduce jobs can be incorporated and can be ran from the tool. This will be a first of its kind tool to run a Hadoop job.

V. System Design

5.1 DISTRIBUTED ENVIRONMENT

Assume that ip address of the master machine is x.x.x.x and that ip address of the slave achine is y.y.y.y .Configure Prerequisites for Pseudo -mode Cluster on both the machines.

Networking

Both machines must be able to reach each other over the network. The easiest way is to put both machines in the same network with regard to hardware and software configuration, for example connect both machines via a single hub or switch and configure the network interfaces to use a common network.

To make it simple, we will assign the IP address x.x.x.x to the master machine and y.y.y.y to the slave machine. Update /etc/hosts on both machines .For example:You need to have root access to update the hosts file.

SSH access

The hadoop user on the master must be able to connect. To its own user account on the master– i.e. ssh master in this context and not ssh localhost.

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>

<property>
        <name>mapred.job.tracker</name>
        <value>master:9001</value>
</property>

</configuration>
```

1. To the hadoop user account on the slave via a password-less SSH login. If you followed pseudo-mode Cluster Prequisities, you just have to add the hadoop@master‘s public SSH key (which should be in $HOME/.ssh/id_rsa.pub to the authorized_keys file of hadoop@slave (in this user’s $HOME/.ssh/authorized_keys). You can do this manually or use the following ssh command.$ ssh-copy-id -i $HOME/.ssh/id_rsa.pub hadoop@slave-This command will prompt you for the login password for user hadoop on slave, then copy the public SSH key for you, creating the correct directory and fixing the permissions as necessary.So, connecting from master to master and master to slave.

Configuration:-Conf/slaves

This conf/slaves file lists the hosts, one per line, where the Hadoop slave daemons (DataNodes and TaskTrackers) will be run. We want both the master box and the slave box to act as Hadoop slaves because we want both of them to store and process data.If you have additional slave nodes, just add them to the conf/slaves file, one per line Make the following files configuration entries for both master and slave machines.

Mapred-site.xml

hdfs-site.xml <value>home/hadoop/hadoop0.20full</v>

core-site.xml

Formatting the name node:-

$bin/hadoop namenode -format

To start all the daemons:-

```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

<!-- Put site-specific property overrides in this file. -->

<configuration>
  <property>
      <name>fs.tmp.dir</name>
      <value>/home/hadoop/hadoop-0.20.2/full</value>
      <description>A base for other temporary directories.</description>
  </property>



  <property>
       <name>fs.default.name</name>
       <value>hdfs://master:9000</value>
  </property>

</configuration>
```

$ bin/start-all.sh

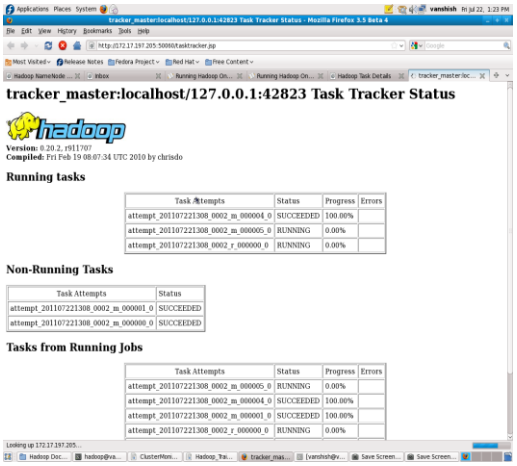
To see the started daemons-$jp

To see files in HDFS:-

$bin/hadoop fs –lsr hdfs://master:9000/

Running a MapReduce job

Copy local data to HDFS



Run the MapReduce job

$ bin/hadoop jar hadoop-0.18.3-examples.jar wordcount hdfs://localhost:9000/input hdfs://localhost:9000/output1

$ bin/hadoop fs –copyToLocal hdfs://localhost:9000/output/part-r-00000 .

Output is as shown below

Hadoop Web Interfaces

Hadoop comes with several web interfaces which are by default available at these locations:

- http://localhost:50070/ – web UI for HDFS name node(s)
- http://localhost:50030/ – web UI for MapReduce job tracker(s)
- http://localhost:50060/ – web UI for task tracker(s)

These web interfaces provide concise information about what's happening in your Hadoop cluster.

HDFS Name Node Web Interface

The name node web UI shows you a cluster summary including information about total/remaining capacity, live and dead nodes. Additionally, it allows you to browse the HDFS namespace and view the contents of its files in the web browser. It also gives access to the "local machine's" Hadoop log files.

MapReduce Job Tracker Web Interface

The job tracker web UI provides information about general job statistics of the Hadoop cluster, running/completed/failed jobs and a job history log file. It also gives access to the "local machine's" Hadoop log files (the machine on which the web UI is running on)

Task Tracker Web Interface

fig5.1.1: Task Tracker Web Interface

The task tracker web UI shows you running and non-running tasks. It also gives access to the"local machine's" Hadoop log files.

VI. Graphical User Interface

1. Makes Hadoop job running easy.
2. You can browse the input data from local.
3. Application Copies the input file to HDFS.
4. Copies the selected class to be executed to remote server.
5. Run the job.
6. Copy the result from HDFS to local.
7. Display the Result and job statistics once completed.
8. Submitting job from out the hadoop environment.

1. Makes Hadoop job running easy.

- The UI tool internally takes care of all the manual steps to be done in order to execute a Hadoop job.
- You can connect to any cluster running.
- Just provide the IP of any node hosting Hadoop and its credentials.
- It will retrieve the job Statistics and Result.

2. You can browse the input data from local.

- No need to copy the files manually to a remote server by scp command from terminal in order to put it to HDFS to get it executed.
- This tool will help you to browse the file system to select the file.
- Copy the input file to the provided remote server.

3. Application Copies the file to HDFS.

- Once the input file is available in the local file system of the remote server, to run a Hadoop job this file should be available in HDFS.
- The tool Internally copies the input file from the ext file system to HDFS
  Hence manual command execution of Hadoop fs –put <file name> is avoided.

4. Copies the selected class to be executed to remote server.

- The tool copies the selected class to be executed to the remote server internally. Hence another overhead is avoided

5. Run the job.

- The tool internally runs the job and generates the output.
- Provides the map reduce outputs and save it to HDFS.
- It saves the job statistics and send it across.

6. Copy the result from HDFS to local.

- To copy the result to the local server from the remote server, the result should be available in the text file system.
- The tool internally copies the result from HDFS to local file system of remote server.

7. Display the Result and job statistics once completed.

- Once the Result and job statistics are available in local file system of remote server, the tool copies it to the local server.
- And it displays the result and statistics in word pad editor.

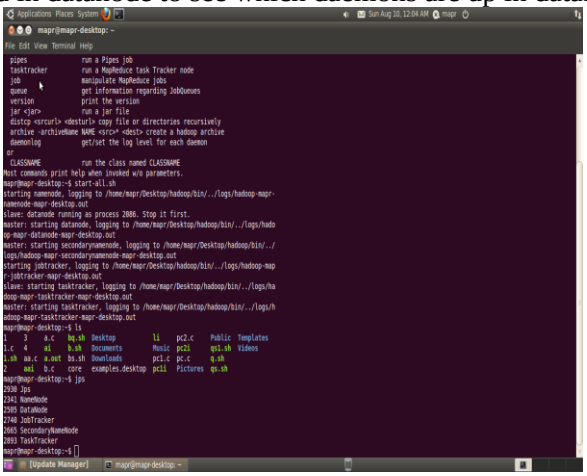
8. Submitting job from out the hadoop environment.

- We can submit jobs from outside the hadoop environment means that without a hadoop system we can get the functionalities of hadoop.
- Without a linux platform and hadoop infrastructure you can submit a job in to the cloud.
- If you are in windows(os) and you have a connectivity then you can submit jods.
- Give your job to name node of the distributed hadoop environment and also select file which you want to perform operations from local. Result will pop up on your tool.

VII. Implementation Results

After the successful installation of hadoop in a distributed environment we need to run hadoop. If the entire configuration is perfect then we get a fully function able network with hadoop infrastructure. The main components are hadoop configured linux os which are used for configuring hadoop and also integrate the functionalities of name node and data node. The second thing is the shared network of hadoop in this one node is act as name node and others as data note. all connectivity are in order then we can submit jobs from different data nodes to name node with the help of GUI tool. Result will obtain in the local node.

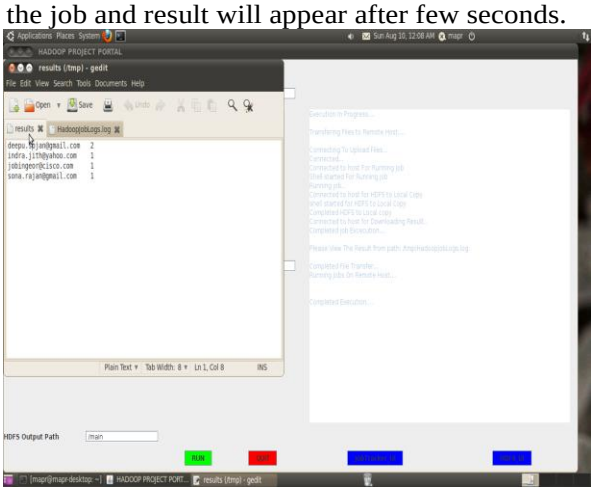
1. In namenode start hadoop and check all the daemons are up or not.
2. Run the same command in datanode to see which daemons are up in datanode.



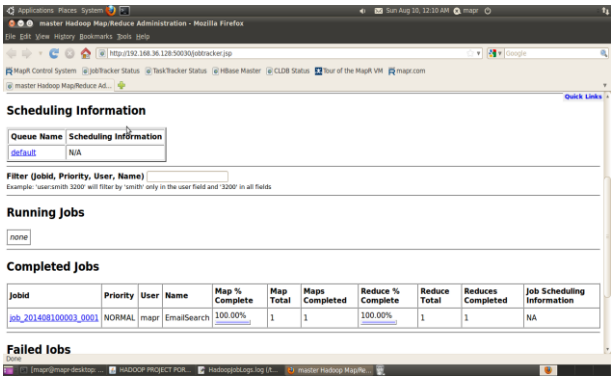
3. Run command hadoop fs –ls / for HDFS both in client and server.
4. Open hadoop job runner GUI tool for submitting job to namenode.

(a).Select file from local for performing the operation.
(b).Select what operation you want to perform.
(c).Enter the namenode address for submitting the job.
(d).Enter the out path in hdfs for storing the final result.

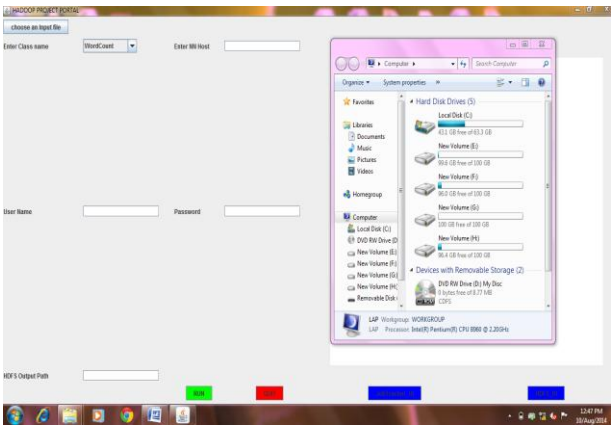
5. Click run button to run the job and result will appear after few seconds.



6. Click job tracker_ui button to see the user interface of hadoop for job tracker.
7. Click HDFS_ui button to see the user interface of hadoop for data base.



8. Run the tool from outside the cluster (Tool run from windows to submit job.)



VIII. Conclusion

This Project talk about hadoop distributed enviornment, MapReduce programming model,HDFS and job runner tool. MapReduce use two operations in functional programming language map and reduce, which allows distributed parallel running. Then we discuss how to deploy Hadoop in a live network. Then MapReduce algorithm to solve the problem that all the hadoop machines which are in the distributed environment that we are created. After that we run some Hadoop programs under the cluster, which shows that it is feasible to deploying Hadoop in this way. Find out why it is feasible to deploy Hadoop in a real world network also discussing the advantages and disadvantages of Hadoop Environment. The advantages are that it can ease the management, fully utilize the computing resources, make Hadoop more reliable and save power and so on. Thus created a real time hadoop environment and it works properly with all concepts of hadoop like name node, data node, distributed file system, submitting jobs from data node, perform operation on the files in local, get the output and hadoop built in GUI on the web browser etc...

After all this the project is moved to the important phase-Hadoop job runner GUI tool. This application can perform all most every job that's normally works in command mode. Basic concepts behind this application are makes Hadoop job running easy. You can browse the input data from local. Application Copies the input file to HDFS. Copies the selected class to be executed to remote server. Run the job. Copy the result from HDFS to local. Display the Result and job statistics once completed. The biggest advantage of this application is that we can submit our jobs from out the hadoop environment even if we are not in linux platform. All these concepts are tested and gets a satisfied result.

REFERENCES

[1] Deploying and Researching Hadoop in Virtual Machines. Guanghui Xu, Feng Xu*, Hongxu Ma College of Computer and Information Hohai University Hohai, Nanjing 211100, China jerryxgh@gmail.com..

[2] Analysis and Optimization of Data Import with Hadoop. Wei Luo Beijing Document Service Beijing, P.R. China lwowen79@gmail.com

[3] A Review on Hadoop – HDFS Infrastructure Extensions Kala Karun. A, Chitharanjan. K Sree Chitra Thirunal College of Engineering Thiruvananthapuram kalavipin@gmail.com, chitharanjank@yahoo.com

[4] Evaluation and Analysis of GreenHDFS: A Self-Adaptive, Energy – Conserving Variant of the Hadoop Distributed File System. Rini T. Kaushik University of Illinois, Urbana-Champaign kaushik1@illinois.edu Milind Bhandarkar Yahoo! Inc.

[5] Hadoop Acceleration in an OpenFlow-based cluster Sandhya Narayan, Stu Bailey InfoBlox Inc. Santa Clara, U.S.A snarayan@infobox.com Anand Daga