www.ijecs.in

International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 4 Issue 4 April 2015, Page No. 11273-11278

HBase Performance Testing On Multi-node Cluster Setup

***G. Nataraja Sekhar*[1]*,Smt. S. Jessica Saritha*[2]*, C. Penchalaiah*[3]**

[1] P.G.Scholor of Computer Science and EngineeringDept, JNTUACEP, Pulivendula,
YSR Kadapa (District), Andhra Pradesh-516390, INDIA
rajgotla@gmail.com

[2] Assistant Professor in Department of CSE JNTUA College of Engineering, Pulivendula ,Andhra Pradesh, India.pin-516390.
sarithajntucep@gmail.com

[3] Dept. of Computer Science and Engineering, JNTUACEP, Pulivendula,
YSR Kadapa (District), Andhra Pradesh-516390, INDIA
penchalaiah2000@gmail.com

Abstract: HBase is also called the Hadoop database since it is a NoSQL database that runs on top of Hadoop. It combines the scalability of Hadoop by running on the Hadoop Distributed File System (HDFS), with real-time data right of entry as a key and value store and deep analytic capabilities of MapReduce. This article introduces HBase and describe how it organizes and manages data and then demonstrates how to set up a local HBase environment and interact with data using the HBase shell. It is a data model that is related to Google's big table designed to provide quick random access to huge amounts of structured data. It leverages the fault-tolerance provided by the Hadoop File System (HDFS). It is a part of the Hadoop ecosystem that provides random real-time read or write access to data in the Hadoop File System. One can store the data in HDFS either directly or during HBase. Data consumer reads and accesses the data in HDFS at randomly using HBase. It stands on top of the Hadoop File System and provides read and write access. HBase is key and value store specifically it is a reliable, Distributed, Multidimensional, Sorted map. HBase stores data in the form of a cells here cells are grouped by a row key into something that looks like a row and cells are stored individually the storage is sparse. HBase Performance testing on single node cluster step-up. but here query performance is slow under single node cluster set-up so this reason you have to introduce the multi-node cluster set-up and improvement of the query performance in HBase Cluster Environment.

Keywords: Hadoop, Mapreduce, hdfs, Hbase, NoSQL and Multi-node Cluster environment.

1. Introduction

The term “Big Data” refers to the continuous massive

The term “Big Data” refers to the continuous massive expansion in the data volume and variety as well as the velocity of data processing.1. Volume refers to the scale of the data and processing needs.2.Variety refers to the different layouts of data like images, documents, videos, streams etc.3.Velocity refers to the speed at which the data is being generated. Data is increasing drastically by day to day. The IDC estimate and put the size of the “digital universe” at 0.18 zetta bytes in 2006, and is forecasting a tenfold growth by 2011 to 1.8 zetta bytes.

This overflow of data is coming from many sources such as Twitter[2], Facebook, LinkedIn, New York stock exchange ike that. This entire thing is creating huge volume of data. This resulted in the development of BIG DATA[1], but it has become hard for storing and analyzing this data.

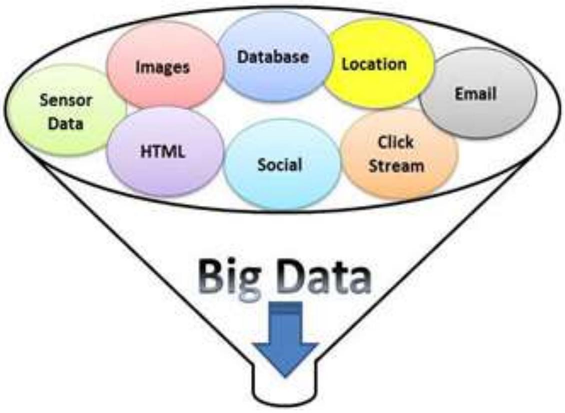


Figure 1: Big data flow

1.1 Hadoop: Doug Cutting was created by Hadoop, the creator of Apache License, the extensively used text search library. Hadoop has its genesis in Apache Nutch, an open source web search engine. Hadoop was ready its own top-level project at Apache in January 2008, which was developed by Doug Cutting after joining his team in Yahoo. Hadoop distributes data and computes across a large number of computers. Hadoop is an open source, horizontally scalable system for dependably storing and processing huge amounts of data across many commodity servers. Hadoop is a distributed file system and how data is retrieved and distributed. There are primarily two major components of Hadoop. They are 1. MapReduce: Large-Scale Data Processing .Map Reduce Architecture provides automatic parallelization & distribution, fault tolerance, I/O scheduling Restricted parallel programming model meant for large clusters.

Map Reduce divides the workload into multiple independent tasks and schedule them across cluster nodes. In a Map Reduce cluster, data is scattered to all the nodes of the cluster as it is being loaded in. 2.HDFS(Hadoop Distributed File System)[3]: It is refers to “Hadoop Distributed File System”. HDFS is designed for the storing of very large files with streaming data access patterns, organization on clusters of commodity hardware. The key part of HDFS is its architecture itself. The HDFS is designed in the form of a master and slave pattern. Here, the master is the Name Node(NN) and the slave is the DataNode(DT). HDFS is used for delivering a high throughput of data. So, it will not work efficiently for low-latency data access. The limit to the number of files in a file system could be controlled by the amount of memory on the namenode. Each file, directory and block can take about 150 bytes.In HDFS only a single writer is present i.e., no several writers. Files in HDFS are busted into block-sized chunks, and stored as independent units called as Blocks. The default block size is 64MB in Hadoop Environment. Replication factor is the most important observed in HDFS. Replication refers to the copying or storing of data in more than one location on a cluster (default replication is 3).

1.2 HBase

HBase is a column-oriented database it’s an open-source implementation of Google’s Big-Table storage structural design. It can manage structured and semi-structured data and it has some built-in features such as versioning, scalability, density and garbage collection. Since its uses write-ahead logging and distributed configuration, it can provide fault-tolerance and fast recovery from individual server failures. HBase[5] built on top of Hadoop / HDFS and the data stored in HBase can be manipulated using Hadoop’s MapReduce[4] capabilities.

Let’s now take a look on how HBase (a column-oriented database) is dissimilar from some other data structures and semi-structured concepts that we are familiar with Row-Oriented vs. Column-Oriented data stores. As shown below, in a row-oriented data store, a row is a unit of data that is read or written together. In a column-oriented data store, the data in a column is stored mutually and hence speedily retrieved. Data is stored and retrieved one row at a time and hence could read redundant data if only some of the data in a row is required. Easy to read and write records. Well suited for OLTP systems. Not professional in performing operations applicable to the entire dataset and hence aggregation is an special operation. Typical solidity mechanisms provide less efficient results than those on column-oriented data store.

2. Existing Techniques and Approaches

However, a challenge remains: how do you move an reachable data shipping to Hadoop, when that transportation is based on traditional relational databases and the Structured Query Language (SQL). What about the large base of SQL users, both expert database designers and administrators, as well as casual users who use SQL to extract in sequence from their data warehouses. This is where Hive[6] comes in. Hive provides an SQL vernacular, called Hive Query Language (abbreviated HiveQL or just HQL) for querying data stored in a Hadoop cluster. The Query Processing quickness will be growing by using Apache Hbase in multi node cluster setup.

3. Proposed System

HBase Performance testing on single-node cluster step-up. But the query performance is slow because it is in under single node cluster set-up so in this reason. Perform and set-up have to introduce the multi-node cluster set-up and improvement of the query performance in HBase[7] Cluster Environment.

4. Methodology Approches

4.1 Hadoop

HDFS has a master and slave architecture. An HDFS cluster consists of a single NameNode(NN), a master server that manages the file system namespace and regulates access to files by customers. It also consist of secondary namenode(SNN). It updates the namespace image with datalog.

A addition requested by an application is much more resourceful if it is executed near the data it operates on. This is especially true when the size of the data set is huge. The statement is that it is often better to migrate the computation closer to where the data is located slightly than moving the data to where the application is running. HDFS provides interfaces for applications to move themselves closer to where the data is located.

The client opens the file it wishes to read by calling open() on the FileSystem object,

which for HDFS is an instance of DFS. It calls the namenode, using RPC, to determine the locations of the blocks for the first few blocks in the file. For each block, the namenode(NN) returns the addresses of the datanodes that have a copy of that block and the datanodes are sorted according to their immediacy to the client. The DFS returns a FSData Input Stream (an input stream that supports file seeks) to the client for it to read data from.

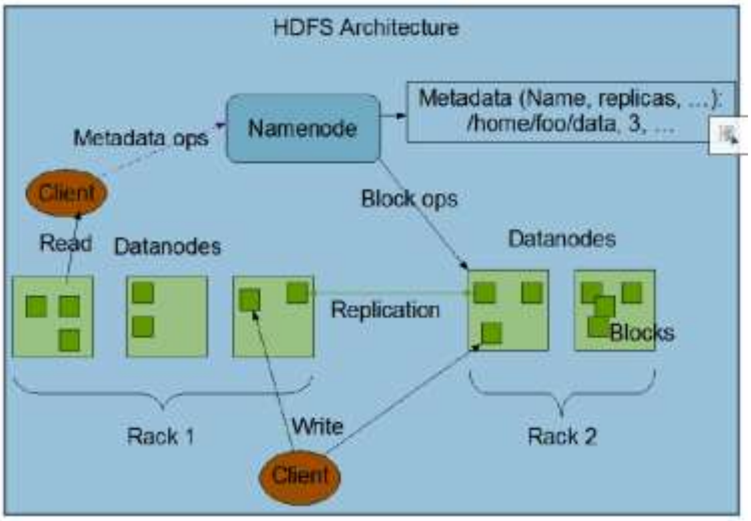


Figure 2: HDFS Structure

4.2 NoSQL(Not only SQL)

Directory of NoSQL[9],[11] databases with basic information on the individual datastores.

4.2.1. ACID

All RDBMS[10] will follows the ACID Properties:

ACID (an contraction for Atomicity Consistency Isolation Durability) is a concept that Database Professionals generally look for when evaluating databases and application architectures. For a dependable database all this four attributes should be achieved.

Atomicity is an all-or-none proposition.(roll back is there).

Consistency guarantees that a transaction never leaves our database in a half-finished state.(two(2) people are Quantating the same thing).

Isolation keeps transactions separated from each other until they are finished.

Durability guarantees that the database will keep track of awaiting changes in such a way that the server can recover from an abnormal termination.

Table 1: RDMS Vs. NoSQL

RDBMS	NOSQL
1. ACID	1. NO ACID
2. Schema oriented	.Schema less
3. Vertically Scalable(increase the RAM or HD)	3.Horizontally Scalable(increase the nodes(computers)).
4. Joins are Supported	Joins are not supported.
5. Primary Key is there	5.Primary Key is there
6. CRUD allowed	6.CRUD allowed
7. Store Small Data	7.Store the large

size(<=1TB)	Amount of Data(>1TB)

Categories in NOSQL:

1. Key-value → memcache, dynamo DB.
2. Column oriented→ [HBase[15],Cassandra,Big Table]→ Column Family.
3. Document oriented→ Mongo DB, couch DB(Query is flexible).
4. Graph oriented → Neo4j, Giraph(shortest path).

Big Table Solving the ‘Low Latency and Large Tables.

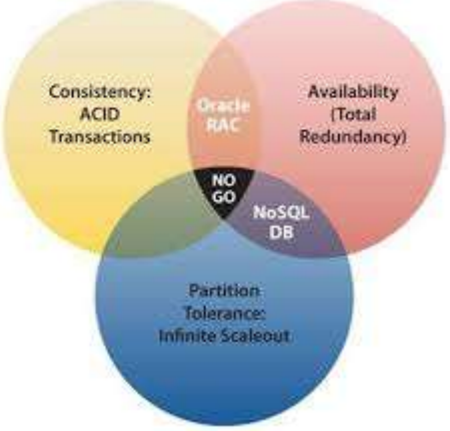

Consistency: ACID Transactions
Oracle RAC
Availability (Total Redundancy)
NO GO
NoSQL DB
Partition Tolerance: Infinite Scaleout

Figure 3: ACID properties

2. CAP Theorem.

Table 2: CAP Table

	Hadoop 1.X	Hadoop 2.X	HBase	Cassandra
Consistency	TRUE	TRUE	TRUE	FALSE
Availability	FALSE	TRUE	FALSE	TRUE
Partitioning	TRUE	TRUE	TRUE	TRUE

Consistency: If changes are one side, it will be reflects all sides.

Availability: It will checking is there or not.

Partitioning: Dividing into Partitions.

CAP Theorem is saying at a time to Achieve two things only.

Hadoop 1.X has Consistency and Partitioning is there but Availability is not here. Because if suppose NameNode(NN) will be done remaining all will be failure. Consistency and Partitioning maintaining why because partitioning when ever storing the data in a single place , can’t store large amount of data, that means distributing the data multiple places doing data partitioning always in Hadoop. We will go for DFS must and should follow Partitioning condition. We will go for HBase, Cassandra Partition must and should factor. Otherwise we can’t store large amount of data.

4.3 Hbase multi-node cluster

Data stored in a distributed that can scale straight to 1,000s of commodity servers and peta bytes of indexed storage. Designed to operate on top of the Hadoop distributed file system (HDFS)[12] OR Kosmos File System (KFS, aka Cloud store) for scalability, fault-tolerance, and high availability. It is Table-like in data structure, multi-dimensional map, High scalability, High availability and performance. Hbase[8] is a Bigtable clone and open source, good community and guarantee for the future. It is developed on top of and has good integration for the Hadoop platform, if you are using Hadoop already. It has a Cascading connector.

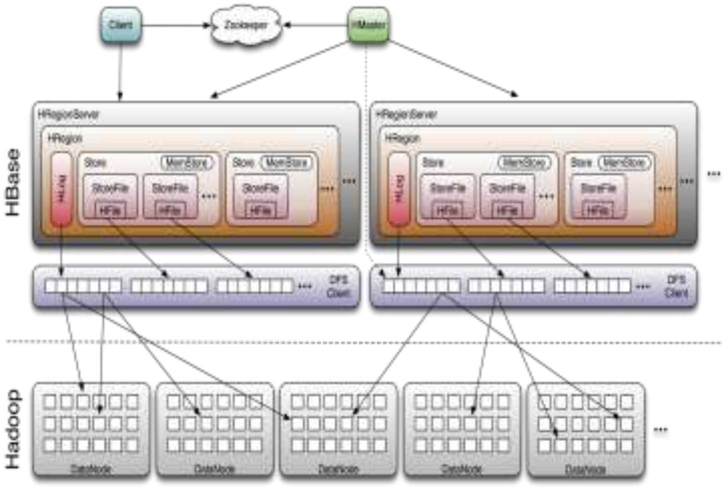


Figure 4: Hbase cluster

5. Implement and Results of Multi-node cluster setup

5.1 Hadoop single node cluster setup

The report here will describe the required steps for setting up a single-node Hadoop cluster[14] backed by the Hadoop. Distributed File System, running on Ubuntu Linux. Hadoop is a framework written in Java for running applications on large clusters of commodity hardware and incorporates features similar to those of the Google File System (GFS) and of the MapReduce computing paradigm. Hadoop’s HDFS is a highly fault-tolerant distributed file system and, like

Hadoop in general, designed to be deployed on low-cost hardware. It provides high throughput entrée to application data and is suitable for applications that have large data sets.

Before we start, we will recognize the meaning of the following:

DataNode(DT): A DataNode stores data in the Hadoop File System. A functional file system has more than one DataNode, with the data replicated across them.

NameNode(NN): The NameNode is the centre piece of an HDFS file system. It keeps the directory of all files in the file system, and tracks where across the cluster the file data is kept. It does not store the data of these file itself.

Jobtracker(JT): The Jobtracker is the service within hadoop that forms out MapReduce[13] to specific nodes in the cluster, ideally the nodes that have the data, or atleast are in the same rack.

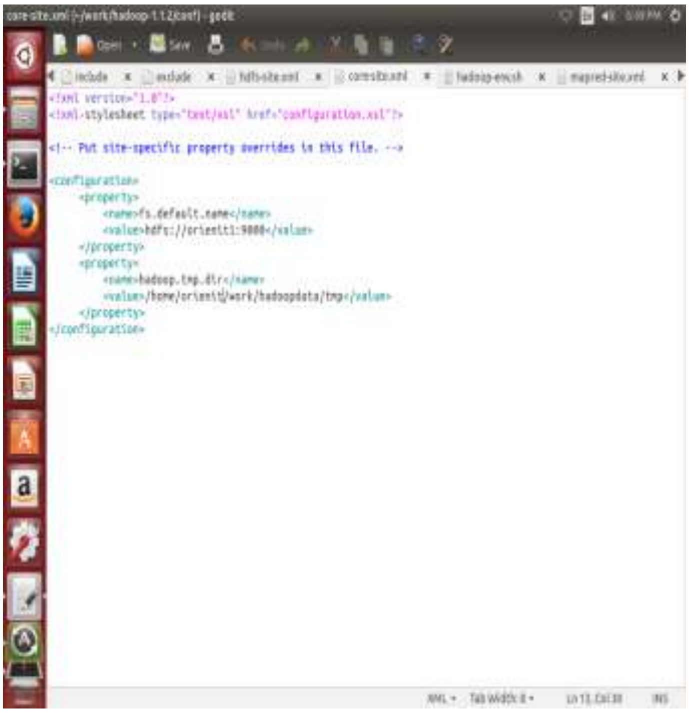
TaskTracker(TT): A TaskTracker is a node in the cluster that accepts tasks- MapReduce and scuffle operations –-from a Job Tracker.

Secondary Namenode(SNN): Secondary Namenode whole purpose is to have a checkpoint in HDFS. It is just a helper node for namenode. 1.1.2 Prerequisites: Java 1.6 JDK

Hadoop requires a working Java 1.6 installation.

Update the source list

cluster@cluster:$ sudo apt-get update(command)

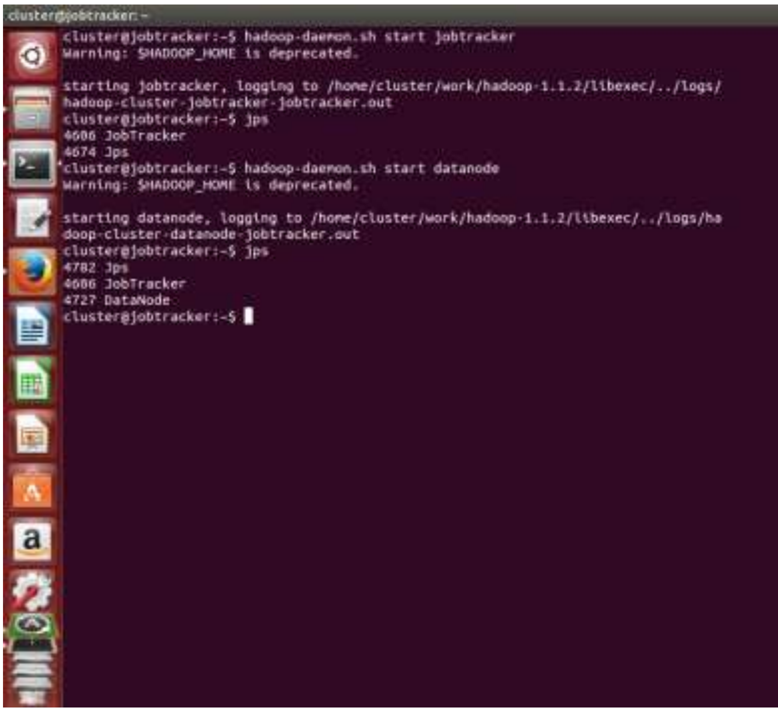


```
<?xml version="1.0"?>
<?xml-stylesheet type="text/xsl" href="configuration.xsl"?>

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

<configuration>
    <property>
        <name>fs.default.name</name>
        <value>hdfs://orientt1:9000</value>
    </property>
    <property>
        <name>hadoop.tmp.dir</name>
        <value>/home/orientt/work/hadoopdata/tmp</value>
    </property>
</configuration>
```

Figure 5: Configure the core-site.xml



```
cluster@jobtracker:~$ hadoop-daemon.sh start jobtracker
Warning: $HADOOP_HOME is deprecated.

starting jobtracker, logging to /home/cluster/work/hadoop-1.1.2/libexec/../logs/
hadoop-cluster-jobtracker-jobtracker.out
cluster@jobtracker:~$ jps
4606 JobTracker
4674 Jps
cluster@jobtracker:~$ hadoop-daemon.sh start datanode
Warning: $HADOOP_HOME is deprecated.

starting datanode, logging to /home/cluster/work/hadoop-1.1.2/libexec/../logs/ha
doop-cluster-datanode-jobtracker.out
cluster@jobtracker:~$ jps
4782 Jps
4606 JobTracker
4727 DataNode
cluster@jobtracker:~$ 
```

Figure 6: Assign the jobtracker

5.2 Multi-node Cluster set-up

5.2.1 Running Hadoop on Ubuntu Linux (Multi-Node Cluster) with in Hbase

Hadoop will build a multi-node cluster unite two or more single-node clusters into one multi-node cluster in which one Ubuntu box will become the designated master but also act as a slave , and the other box will become only a slave.

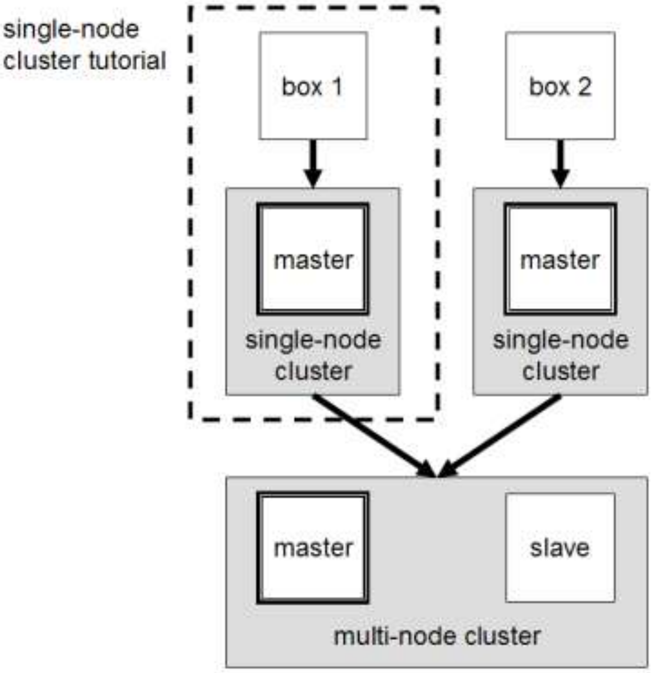


Figure 7: Single node and Multi-node

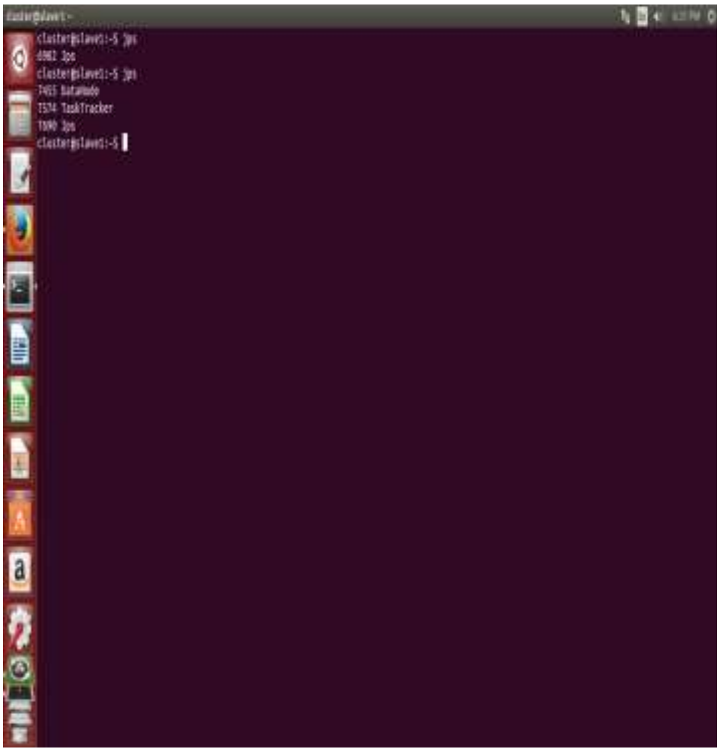
Prerequisites

Configuring single-node clusters first,here we have used two single node clusters. Shutdown each single-node cluster with the following command.

cluster@cluster:$ bin/stop-all.sh(command)

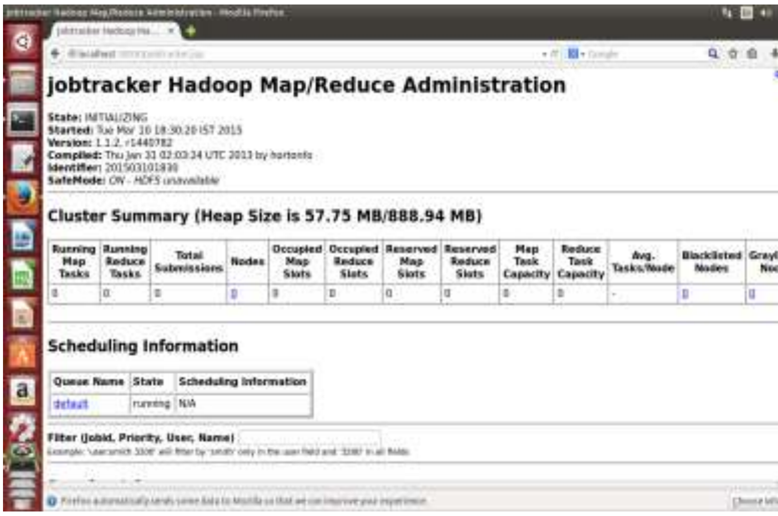
Networking

The easiest is to put both apparatus in the same network with regard to hardware and software configuration. Update /etc/hosts on both machines .Put the alias to the IP addresses of all the machines. Here we are creating a cluster of 2 machines , one is master cluster@cluster :$ sudo gedit /etc/hosts. Add the following lines for two node cluster 172.16.2.19(choose your ifconfig)1 master (IP address of the master node) 172.16.2.231(choose you are ifconfig) slave1 (IP address of the slave node) SSH access. The hduser(hadoop user) user on the master (cluster@master) must be able to connect: 1. to its own user account on the master - i.e. ssh master in this context. 2. to the hduser user account on the slave (i.e. cluster@slave1) via a password-less SSH login. Add the cluster@master public SSH key using the following command and other is slave 1.



```
cluster@slave1:~$ jps
6982 Jps
cluster@slave1:~$ jps
7455 DataNode
7574 TaskTracker
7690 Jps
cluster@slave1:~$
```

Figure 8: cluster in datanode



jobtracker Hadoop Map/Reduce Administration

State: INITIALIZING
Started: Tue Mar 10 18:30:20 IST 2015
Version: 1.1.2, r1440782
Compiled: Thu Jan 31 02:03:24 UTC 2013 by hortonfo
Identifier: 201503101830
SafeMode: ON - HDFS unavailable

Cluster Summary (Heap Size is 57.75 MB/888.94 MB)

Running Map Tasks	Running Reduce Tasks	Total Submissions	Nodes	Occupied Map Slots	Occupied Reduce Slots	Reserved Map Slots	Reserved Reduce Slots	Map Task Capacity	Reduce Task Capacity	Avg. Tasks/Node	Blacklisted Nodes	Grayl Nod
0	0	0	0	0	0	0	0	0	0	-	0	0

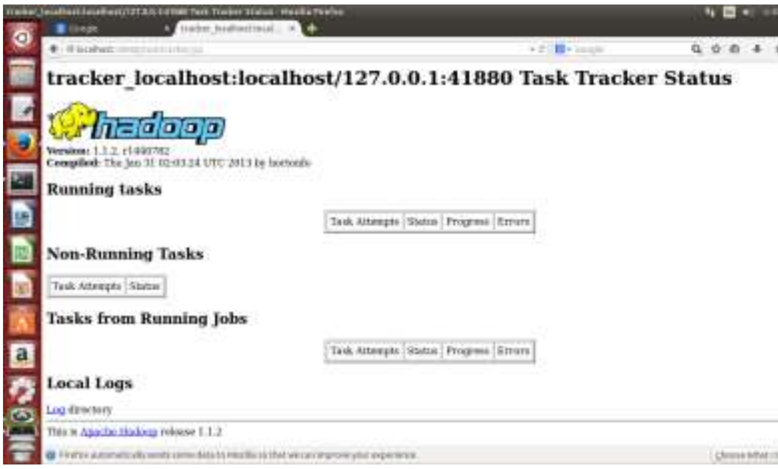
Scheduling Information

Queue Name	State	Scheduling Information
default	running	N/A

Filter (Jobid, Priority, User, Name)
Example: 'user:smith 3200' will filter by 'smith' only in the user field and '3200' in all fields

Firefox automatically sends some data to Mozilla so that we can improve your experience.

Figure 9: Job tracker in browser



tracker_localhost:localhost/127.0.0.1:41880 Task Tracker Status

Version: 1.1.2, r1440782
Compiled: Thu Jan 31 02:03:24 UTC 2013 by hortonfo

Running tasks

Task Attempts	Status	Progress	Errors

Non-Running Tasks

Task Attempts	Status

Tasks from Running Jobs

Task Attempts	Status	Progress	Errors

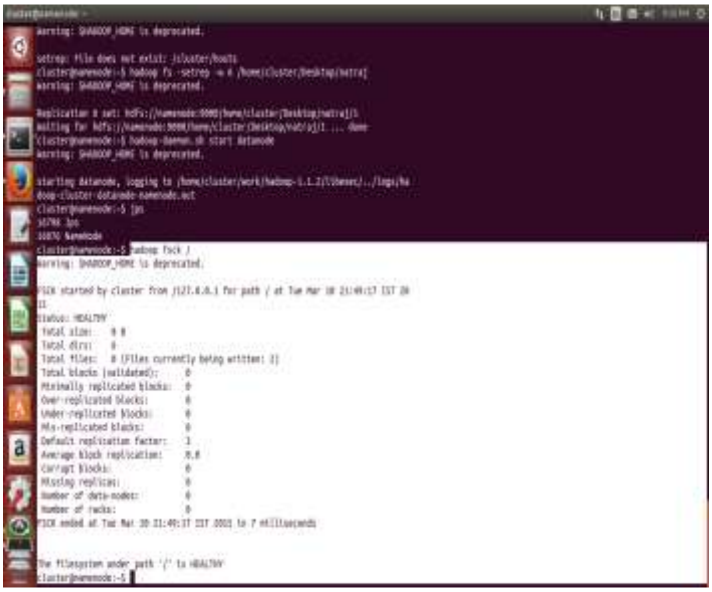
Local Logs

Log directory

This is Apache Hadoop release 1.1.2

Firefox automatically sends some data to Mozilla so that we can improve your experience.

Figure 10: Task Tracker in Browser



```
Warning: $HADOOP_HOME is deprecated.

setrep: File does not exist: /cluster/hosts
cluster@namenode:~$ hadoop fs -setrep -w 4 /home/cluster/Desktop/natraj
Warning: $HADOOP_HOME is deprecated.

Replication 4 set: hdfs://namenode:9000/home/cluster/Desktop/natraj/1
Waiting for hdfs://namenode:9000/home/cluster/Desktop/natraj/1 ... done
cluster@namenode:~$ hadoop-daemon.sh start datanode
Warning: $HADOOP_HOME is deprecated.

starting datanode, logging to /home/cluster/work/hadoop-1.1.2/libexec/../logs/ha
doop-cluster-datanode-namenode.out
cluster@namenode:~$ jps
16798 Jps
16676 NameNode
cluster@namenode:~$ hadoop fsck /
Warning: $HADOOP_HOME is deprecated.

FSCK started by cluster from /127.0.0.1 for path / at Tue Mar 10 21:49:17 IST 20
15
Status: HEALTHY
 Total size:    0 B
 Total dirs:    0
 Total files:   0 (Files currently being written: 2)
 Total blocks (validated):      0
 Minimally replicated blocks:   0
 Over-replicated blocks:        0
 Under-replicated blocks:       0
 Mis-replicated blocks:         0
 Default replication factor:    1
 Average block replication:     0.0
 Corrupt blocks:                0
 Missing replicas:              0
 Number of data-nodes:          0
 Number of racks:               0
FSCK ended at Tue Mar 10 21:49:17 IST 2015 in 7 milliseconds


The filesystem under path '/' is HEALTHY
cluster@namenode:~$
```

Figure 11: Multi-node cluster set-up health checkup

6. Future work and Conclusion

6.1 Future work

It proposed an approach which considers are done the multi-node cluster set-up in Hbase mode. After forming cluster the all nodes working independently and also the RAM competence will be increased. so, the performance testing Hbase will do on multi-node cluster set-up. These multi-node will apply in the future oozie and pig.

6.2 Conclusion

As we know today's world is becoming a narrower. We get reaction of people for particular products, events, issues very fast on web. Automatic sentiment analysis is very useful to identify and predict current and future trends, product reviews, people judgment for social issues, effect of some precise event on people so data will be generated day to day so, the data will be storing concepts introduced by google. The concepts are Hadoop and it,s eco-system. In these eco-system developed by some others. These eco-system providing Hbase. so we are concentrating on performing the multi-node cluster set-up by using eco-system and HDFS.

References

[1] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, and A. H. Byers, "Big Data: The Next Frontier For Innovation, Competition, And Productivity", May 2011.

[2] A. Pak and P. Parouek, "Twitter as a corpus for sentiment analysis and opinion mining," in Proceedings of LREC, vol. 2010.

[3] T. White, "The Hadoop Distributed Filesystem," Hadoop: The Definitive Guide, pp. 41-73, GravensteinHighwaNorth, Sebastopol: O'Reilly Media, Inc., 2010.J.

[4] J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters," Communications of the ACM, Vol. 51, Iss. 1, pp. 107-113, January 2008.

[5] Apache HBase, http://hbase.apache.org/.

[6] (Online Resource) Hive (Available on:http://hive.apache.org/).

[7] S. Nishimura, S. Das, D. Agrawal, and A. Abbadi, "Mdhbase: A scalable multi-dimensional data infrastructure for location aware services," in Mobile Data Management (MDM), 2011 12th IEEE International Conference on, vol. 1. IEEE, 2011, pp. 7–16.

[8] K. Muthukkaruppan, "HBase @ FacebookThe Technology Behind Messages," April 2012. [Online]. Available: http://qconlondon.com/dl/qcon-london-2011/

slides/KannanMuthukkaruppan HBaseFacebook.pdf

[9] ZHANG Huaqiang, "Relational Database and NoSQL database Computer Knowledge and Technology, vol. 7, No. 20, Jul.2011

[10] Rabi Prasad Padhy, Manas Ranjan Patra and Suresh Chanadra , "RDBMS to NoSQL: Reviewing Some Next-Generation Non-re.

[11] HAN Junyi, "Considerations of Application NoSQL Database Solutions," Technology Information, No. 043, 2011, pp. 130.

[12] http://hadoop.apache.org/(Online resource)

[13] https://www.mapr.com/products/apache-hadoop(online source)

[14] http://www.cloudera.com/content/cloudera/en/search.html.hbas

[15] http://hortonworks.com/

Author Profile

Nataraja Sekhar.G was born in Andhra Pradesh, India. He received the B.Tech Degree in Computer Science and Engineering from Jawaharlal Nehru Technological University Anantapur branch, India in 2013 and M.Tech Degree in also same branch and University. His research interests are in the area of Java and Big Data Analytics.

Smt. S.Jessica Saritha M.tech.,[Ph.D] is currently working as an Assistant Professor in Department of CSE JNTUA College of Engineering, Pulivendula ,Andhra Pradesh India Her Research interests are Data mining and distributed computing.

Penchalaiah.C was born in AndhraPradesh, India. He received the B.Tech Degree in Computer Science and Engineering from Jawaharlal Nehru Technological University Anantapur branch, India in 2011 and M.Tech Degree in also same branch and University. His research interests are in the area of Semantic - Web and BigData Analytics.