

HBase Testing and Performance on Cluster Mode

T.Ramakrishna¹, V.Brahmaiah², M. Chenna Keshava³

M.Tech, Dept of CSE, JNTUACE

Pulivendula, A.P. India,

ramakrushna579@gmail.com¹, brahmaiah788.820ce@gmail.com², keshava1047@gmail.com³

Abstract— HBase is a data model and Hadoop database because it is similar to Google’s big table designed to provide quick random access to very large amounts of structured data. To connect HBase and perform basic operations on HBase by using java. It is NoSQL information that runs on prime of Hadoop. It combines the measurability of Hadoop by running on the Hadoop Distributed filing system (HDFS), with period of time knowledge access as a key value store and deep analytic capabilities of Map scale back. This paper introduces HBase and describes how it organizes are manages information and then demonstrates how to set up a local HBase environment, interact with data using the HBase shell in command wrapper. It leverages the fault tolerance provided by the Hadoop Distributed filing system (HDFS). It is a contribution of the Hadoop ecosystem that provides random period of time read and write access to information within the Hadoop File System. One will store the information in HDFS either directly or through HBase. Data consumer read and accesses the information in HDFS randomly mistreatment HBase. It sits on high of the Hadoop classification system and provides browse and write access. HBase is key value store specifically it’s the same, distributed, four-dimensional, sorted map. HBase stores information within the style of a cells here cells ar sorted by a row key into one thing that appears sort of a row and cells are hold on severally the storage is distributed. Currently the bunch, within the context of databases, refers to the flexibility of many servers or instances to attach to the information. Associate in nursing instance is that the assortment of memory and processes that interacts with information, that is that the set of physical files that truly store information. In single node setup demons that is name node, secondary name node, data node, job tracker and task tracker are not visible to user in Hadoop command wrapper so as reason we introduce the cluster mode and improvement of the query performance in the HBase Cluster Environment.

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

I. INTRODUCTION

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 zettabytes in 2006, and is forecasting a tenfold growth by 2011 to 1.8 zettabytes. This flood of data is coming from many sources such as Facebook, Twitter [2], and LinkedIn, New York stock exchange. This entire thing is creating huge volume of data. This resulted in the evolution of BIG DATA [1], but has become difficult for storing and analyzing this data.



Figure 1: Big data flow

Hadoop was created by Doug Cutting, the creator of Apache Lucene, the widely used text search library. Hadoop has its origins in Apache Nutch, an open source web search engine. Hadoop was made 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 climbable system for reliably storing and processing massive 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. Map Reduce: 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 broadcast to all the nodes of the cluster as it is being loaded in. 2.HDFS(Hadoop Distributed File System)[3]: It refers to “Hadoop Distributed File System”. HDFS is designed for the storing of very large files with in the streaming data and access patterns, running on the 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 NameNode and the slave is the DataNode. 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 quantity 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 multiple writers. Files in HDFS are breached into block-sized clods, and stored as independent units called as Blocks. The default size of a block is 64MB. Replication is the most important factor observed in HDFS. Replication refers to the copying or storing of data in more than one location on a cluster (default replication is 3).

HBase could be a column-oriented info that’s Associate in Nursing ASCII text file implementation of Google’s huge Table storage design. It will manage structured and semi-structured information and has some constitutional options like measurability, versioning, compression and garbage pickup. Since its uses write-ahead work and distributed configuration, it will give fault-tolerance and fast recovery from individual server failures. HBase[5] designed on high of Hadoop / HDFS and therefore the information hold on in HBase will be manipulated exploitation Hadoop’s Map Reduce[4] capabilities.

Hive [6] provides an SQL dialect, called Hive Query Language (abbreviated HiveQL or just HQL) for querying data stored in a Hadoop cluster. The Query Processing speed will be increasing by using Apache Hbase in multi node cluster setup

Cluster Mode is Spark applications run as freelance sets of processes on a cluster, coordinated by the Spark Context object in your main program (called the driving force program). Specifically, to run on a cluster mode, the Spark-Context will hook up with more forms of cluster managers (either Spark’s own standalone cluster manager or Mesos/YARN), that portion resources across applications. Once connected the Spark acquires executors on nodes within the cluster, that are processes that run computations and store information for your application. Next it sends your application code (defined by JAR or Python files passed to Spark-Context) to the executors. Finally, the Spark-Context sends to task for the executors to run

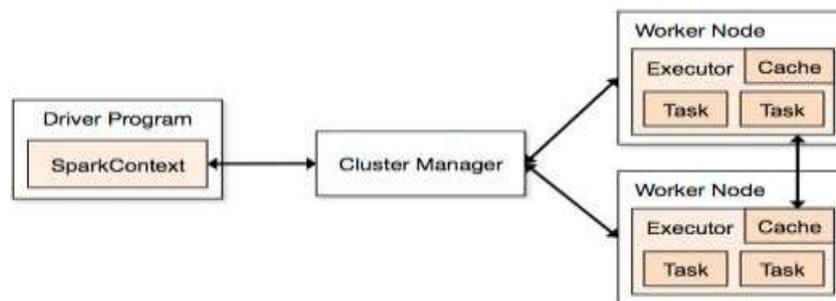


Figure 2: Cluster Manager Overview

- (1) Each application gets its own fiduciary processes that not blink for the period of the entire application and run tasks in multiple threads. This has the advantage of uninflected applications from one another, on each the planning facet (each driver schedules its own tasks) and fiduciary facet (tasks from completely different applications run in numerous JVMs). However, it additionally implies that information can not be shared across completely different Spark applications (instances of SparkContext) while not writing it to associate degree storage device system.
- (2) Spark is agnostic to the underlying cluster manager. As long because it will acquire fiduciary processes and these communicate with one another, it's comparatively straightforward to run it even on a cluster manager that conjointly supports different softwraes (e.g. Mesos/YARN).
- (3) Because the driving force schedules tasks on the cluster, it ought to be run near the employee nodes, ideally on constant native space network. If you'd prefer to send requests to the cluster remotely, it's higher to open associate degree RPC to the driving force and have it submit operations from close than to run a driver isolated from the employee nodes.

II. MOTIVATION

In this paper the performance for testing on the 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 we introduce the multi-node cluster mode set-up and improvement of query performance in the HBase cluster mode Environment.

III. METHODOLOGY APPROACHES

A. Hadoop

HDFS has master/slave design. Associate in Nursing HDFS cluster consists of one NameNode, a master server that manages the filesystem namespace and regulates access to files by shoppers. It additionally consist of secondary name node. It updates the namespace image with information log. A computation reputation by Associate in Nursing application is far additional economical if it's dead close to the info it operates on. this can be very true once the scale of the info set is big. The Belgium is that it's typically higher to migrate the computation nearer to wherever the info is found instead of moving the info to wherever the appliance is running. HDFS provides an interface for applications to maneuver themselves nearer to wherever the info is found.

The shopper opens the file it needs to scan by business open () on the FileSystem object, that for HDFS is Associate in Nursing instance of DFS. It calls the namenode, using RPC, to see the locations of the blocks for the primary few blocks within the file. For all block, the namenode returns the addresses of the datanodes that have a duplicate of that block and therefore the datanodes square measure sorted per their proximity to the shopper. The DFS returns a FSDataInputStream (an input stream that supports file seeks) to the shopper for it to scan knowledge from.

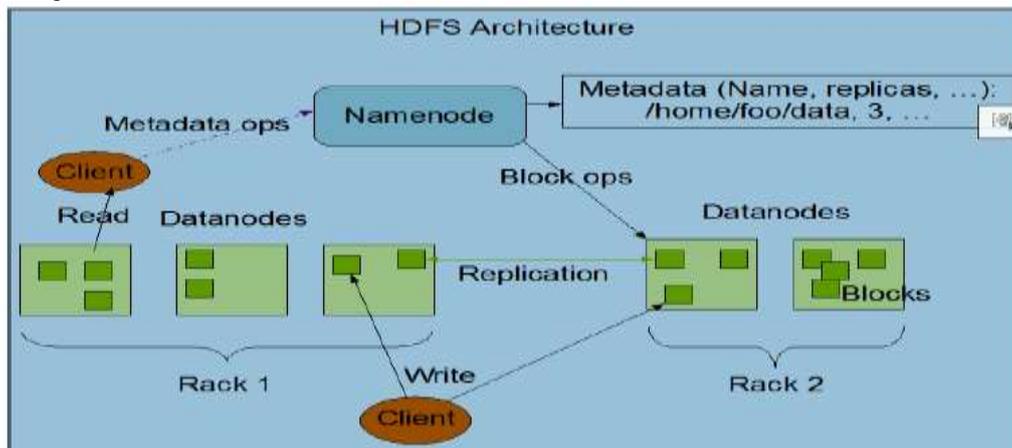


Figure 3: HDFS Structure

B. NoSQL (Not only SQL)

Directory of NoSQL [8][10] databases with basic information on the individual data stores.

1) ACID

All RDBMS [9] will follow the ACID Properties:

- a) *Acid*: (an acronym for Atomicity Consistency Isolation Durability) it is a concept of Database Professionals generally look for when evaluating databases and application architectures. For a reliable database all this four attributes should be achieved.
- b) *Atomicity*: it is an all-or-none proposition.(roll back is there).
- c) *Consistency*: ensures that a transaction never leaves your database in a half-finished state. (two people are Quanta ting the same thing).
- d) *Isolation*: keeps transactions separated from each other until they're finished.
- e) *Durability*: guarantees that the database will keep track of pending 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	2. Schema less

3. Vertically Scalable(increase the RAM or HD)	3. Horizontally Scalable (increase the nodes (computers)).
4.Joins are Supported	4. Joins are not supported.
5.Primary Key is there	5.Primary Key is there
6.CRUD allowed	6.CRUD allowed
7.Store Small Data size(<=1TB)	7.Store the large Amount of Data(>1TB)

2) 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.

3) 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

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

b) Availability: It will checking is there or not.

c) 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) Hbase multi-node cluster

A distributed data store that can scale horizontally to 1,000s of commodity servers and pet bytes of indexed storage. Designed to operate on top of the Hadoop distributed file system (HDFS)[11] or Kosmos File System (KFS, aka Cloudstore) for scalability, fault tolerance, and high availability. It is Table-like in data structure, multi-dimensional map, High scalability, High availability and performance. Hbase[7] is a Bigtable clone and open source, good community and promise 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.

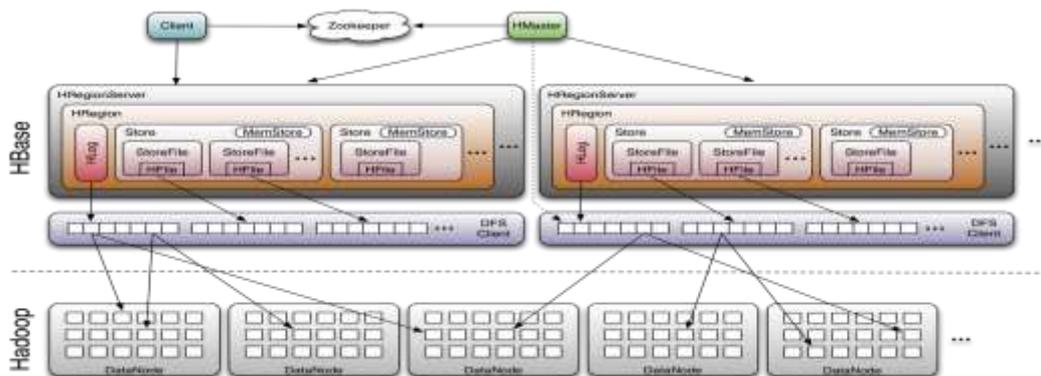


Figure 4: Hbase cluster

IV IMPLEMENT AND RESULTS OF CLUSTER MODE

A. HBase Testing

The report here will describe the HBase may be a column-oriented info that's Associate in Nursing yank customary Code for information Interchange file implementation of Google's large Table storage vogue. it'll manage structured and semi-structured info and has some intrinsic selections like quality, variant, compression and pickup. Then it's used write-ahead work and therefore the distributed configuration, it'll be provide fault-tolerance and therefore the short recovery from individual server failures. The HBase designed for high in Hadoop / HDFS then the data confine HBase unit of measure generally manipulated pattern Hadoop's MapReduce capabilities. That needed steps unit of measure fitting a locality a single-node Hadoop cluster [14] backed by Hadoop. The distributed system, running on Ubuntu package. Hadoop may be the framework written in Java for running applications on large clusters of entity hardware and incorporates selections identical as those of the Google system (GFS) and of the MapReduce computing paradigm. Hadoop's HDFS may be a extremely fault-tolerant distributed filing the system, rather just like the Hadoop unit of measure typically, designed for deployed on reasonable hardware. It providing more output access to application info and is suitable for applications that have large info sets.

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

- a) *DataNode*: A DataNode stores data in the Hadoop File System. A functional file system has more than one DataNode, with the data replicated across them.
- b) *NameNode*: The NameNode is the centrepiece of an HDFS file system. It keeps the directory of all files in the file system, tracks where across the cluster in the file data is kept. It does not store the data of these file itself.
- c) *Jobtracker*: The Jobtracker is the service within hadoop that farms out MapReduce [12] to specific nodes in the cluster, ideally the nodes that have the data, or at least are in the same rack.
- d) *TaskTracker*: A TaskTracker is a node in the cluster that accepts tasks- Map, Reduce and Shuffle operatons – from a Job Tracker.
- e) *Secondary Namenode*: Secondary Namenode whole purpose is to have a checkpoint in HDFS. It is just a helper node for namenode. 1.1.2

Prerequisites: Java 6 JDK

Hadoop requires a working Java 1.6 installation.

Update the source list

```
user@ubuntu:~$ sudo apt-get update
```

B. cluster mode

Cluster analysis or agglomeration is that the task of clustering a group of objects in such some way that objects within the same group (called a cluster) area unit a lot of similar (in some sense or another) to every aside from to those in alternative teams (clusters).

While Infinispan is especially attention-grabbing in clustered mode, it additionally offers a awfully capable native mode, wherever it acts as an easy and in-memory knowledge cache almost like JBoss Cache and EHCACHE. However why would one use a adjacency cache instead of a map? Caches provide plenty of options over and on top of an easy map, together with write-through and write-behind caching to persist knowledge, eviction of entries to stop running out of memory, support for end entries. Infinispan, specifically, is constructed around a superior, read-biased knowledge instrumentality that uses trendy techniques like MVCC lockup - that buys you non-blocking and thread-safe reads even once co-occurring writes area unit going down. Infinispan additionally cause significant use of compare-and-swap and alternative lock-free algorithms, creating it ideal for high-throughput, multi-CPU and multi-core environments. Further, Infinispan's Cache API extends the JDK's Concurrent Map - creating motion from a map to Infinispan trivial.

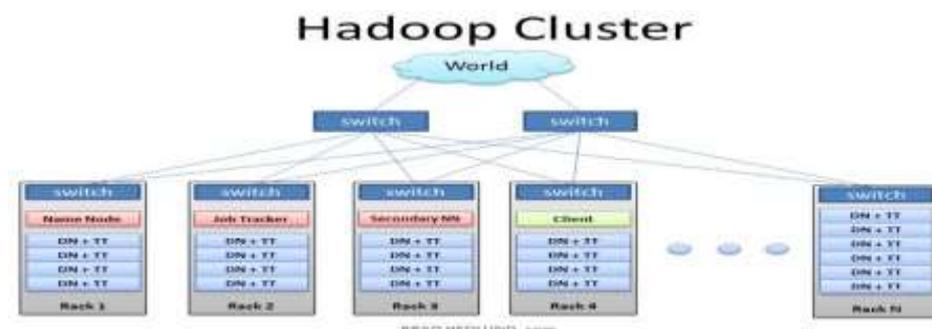


Figure 5: Hadoop cluster



Figure 6. Configure the core-site.xml

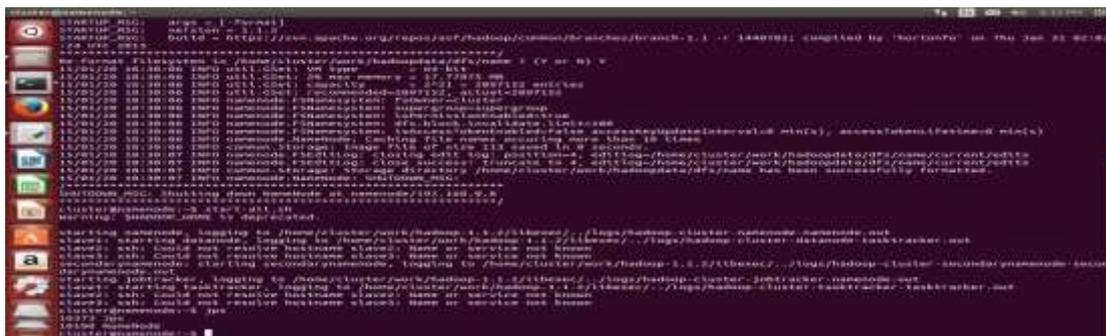


Figure 7. Setting the namenode



Figure 8. Assign the jobtracker

Running Hadoop on Ubuntu Linux (Multi-Node Cluster) within Hbase.

Hadoop can build a multi-node cluster merge multiple single-node clusters into one multi-node cluster among which one Ubuntu box can become the selected master however additionally act as a slave , and therefore the different box can become solely a slave.

Prerequisites

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

```
user@ubuntu:~$ bin/stop-all.sh
```

Networking

The easiest way is to put in the both machines 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 every machine in the network. Here we are creating a cluster of two machines, one is master cluster@master:\$ sudo gedit /etc/hosts. Add the following lines for two nodes cluster 172.16.2.191 master (IP address of the master node) 172.16.2.231 slave1 (IP address of the slave node) SSH access. The hduser user on the master (cluster@master) must be able

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 its eco-system. In these eco-system developed by some others. These eco-system providing Hbase. So we are concentrating on Hbase testing and performing on cluster mode by using eco-system and HDFS. After forming cluster the all nodes working separately and also the RAM capacity will be increased. So, the performance testing HBase will do on multi-node cluster setup. With the explosive growth of network data, so NoSQL database have been widely used and have become a substitute for traditional databases here MSOL databases can be used in combination with relational databases. The above performance test results are not enough they are still in the growth stage. This multi-node cluster mode will apply in the future oozie and pig. Nowadays

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 (Availabl on:<http://hive.apache.org/>).
- [7] K. Muthukkaruppan, "HBase @ FacebookThe Technology Behind Messages," April 2012. [Online].Available: <http://qconlondon.com/dl/qcon-london-2011/slides/KannanMuthukkaruppanHBaseFacebook.pdf>
- [8] ZHANG Huaqiang, "Relational Database and NoSQL database Computer Knowledge and Technology, vol. 7, No. 20, Jul.2011
- [9] Rabi Prasad Padhy, Manas Ranjan Patra and Suresh Chanadra, "RDBMS to NoSQL: Reviewing Some Next-Generation Non-re.
- [10] HAN Junyi, "Considerations of Application NoSQL Database Solutions," Technology Information, No. 043, 2011, pp. 130.
- [11] <http://hadoop.apache.org/>(Online resource)
- [12] <https://www.mapr.com/products/apache-hadoop> (online source)