Scientific Journal of Impact Factor (SJIF): 3.134

e-ISSN (O): 2348-4470
p-ISSN (P): 2348-6406

International Journal of Advance Engineering and Research Development

Volume 3, Issue 1, January -2016

A SURVEY ON HADOOP HBASE SYSTEM

Nikita Bhojwani[1] ,Asst Prof. Vatsal Shah[2]

[1]B.V.M. Engineering College, V.V. Nagar, Gujarat, India
[2]B.V.M. Engineering College, V.V. Nagar, Gujarat, India

Abstract: *In this paper, we will discuss that how HBase works and how it is better than RDBMS. Actually HBase is a distributed column oriented database in which the data can be fetched randomly. Apparently, it provides a very high performance, low latency access to single rows from billions of records and manages fault tolerance. Access patterns are well known and simple. It sits on top of HDFS(hadoop distributed file system). It deals with enormous tables. It stores data in indexed HDFS files for faster lookups. Though, hbase is a column oriented database but the data in it are sorted by row.It also includes the architecture comprising of master server, region, region servers and zookeeper. Moreover,HBase contains a shell in which we can communicate with HBase. There are some general commands for communication as well as for certain operations. Also, there is an admin API in which communication is done with HBase and manages HBase. It is extremely useful for heavy read- write and to access the randomly large data. Companies such as facebook, Yahoo etc use HBase internally.*

Keywords*: Hbase, HDFS, Zookeper, Hadoop, Database, Communication.*

I. INTRODUCTION

1.1. Hbase0

Hbase is a distributed column oriented database built on top of HDFS(hadoop distributed file system). It is an apache open source project whose objective is to provide storage for the hadoop distributed computing[2]. Data is logically organized into tables, rows and columns[2]. It is horizontally scalable. HBase is the data model that is similar to Google's big table designed to provide quick random access to enormous amount of structured data. It influences the fault tolerance provided by HDFS[1]. A data row in HBase is a sortable row key and a variable number of columns, which are further grouped into sets called column families[3]. Each data cell in HBase can contain multiple versions of the same data which are indexed by timestamps.It is part of the hadoop ecosystem that provides the random real read/write access to data in hadoop file system[1][7][8]. One can accumulate the data in HDFS either directly or through Hbase. Data consumer reads/access the data in HDFS randomly using HBase[2]. HBase sits on top of the Hadoop file system and provides read and write access [3].

HBase is very different from traditional relational databases like SQL, MySQL, PostgreSQL, Oracle etc in how it's architected and the features that it provides to the applications using it[5][6] . HBase trades off some of these features for scalability and a flexible schema[5][6]. This also translates into HBase having a very different data model. Designing HBase tables is a different ballgame as compared to relational database systems[5][6].

1.1.1.Data Storage Model[4]:-

- Column-oriented database(column families)
- Table consist of rows, each of which has a primary key(row key).
- Each row may have any number of columns.
- Table schema only defines column families(column family can have any number of columns)
- Each cell value has a timestamp.

1.1.2.HDFS AND HBASE[2]

HDFS	HBASE
HDFS is a distributed file system suitable for storing large files	HBASE is a database build on top of HDFS
HDFS does not support fast individual record lookup	HBASE provides fast lookups for larger tables.
It provides high latency batch processing ; no concept of batch processing	It provides low latency access to single rows from billions of records.
It provides only sequential access of data.	Hbase internally uses hash tables and provides random access and it stores the data in indexed HDFS files for faster lookups.

Table 1. HDFS AND HBASE

1.1.3. Storage mechanism in Hbase

HBase is a **column-oriented database** and the tables in it are sorted by row. The table schema defines only column families, which are the key value pairs[1]. A table have multiple column families and each column family can have any number of columns. Subsequent column values are stored contiguously on the disk. Each cell value of the table has a timestamp[1]. In short, in an HBase:

- Table is a collection of rows.
- Row is a collection of column families.
- Column family is a collection of columns.
- Column is a collection of key value pairs.

1.2. Column Oriented and Row Oriented[1]

Column – oriented databases are those that store data tables as sections of columns of data, rather than as rows of data[1]. Shortly, they will have column families

Row- Oriented Database	Column Oriented Database
It is suitable for online transaction process(OLTP).	It is suitable for online analytical processing(OLAP).
Such databases are designed for small number of rows and columns.	Column-oriented databases are designed for huge tables.

Table 2.Column Oriented and Row Oriented

1.3. HBASE AND RDBMS[2]

HBASE	RDBMS
Hbase is schema-less, it doesn't have the concept of fixed columns schema; defines only column families.	An RDBMS is governed by its schema, which describes the whole structure of tables.
It is built for wide tables. Hbase is horizontally scalable.	It is thin and built for small tables. Hard to scale.
There are no transactions in Hbase.	RDBMS is transactional.
It has de-normalized data.	It will have normalized data.
It is good for structured as well as semi-structured data,	It is good for structured data.

Table 3HBASE AND RDBMS

1.3.1. Features of Hbase

- HBase is linearly scalable.
- It has automatic failure support.
- It provides consistent read and write files.
- It integrates with hadoop both as a source and a destination.
- It has easy java API for client.
- It provides data replication across clusters.

1.3.2. Where to use Hbase

- Apache Hbase is used to have random, real time read/write access to big data[2].
- It tracts very large tables on top of clusters of commodity hardware.
- Apache Hbase is a non- relational database modeled after Google's bigtable. Bigtable is a compressed, high performance and proprietary data storage system built on Google file system, chubby lock service SS table (log-structured storage like level DB) and a few other Google technologies. It acts upon Google file system, likewise apache Hbase works on top of hadoop and HDFS[1].

II. BACKGROUND THEORY

In Hbase, tables are split into regions and are served by the region servers. Regions are vertically divided by column families into "Stores". Stores are saved as files in HDFS.

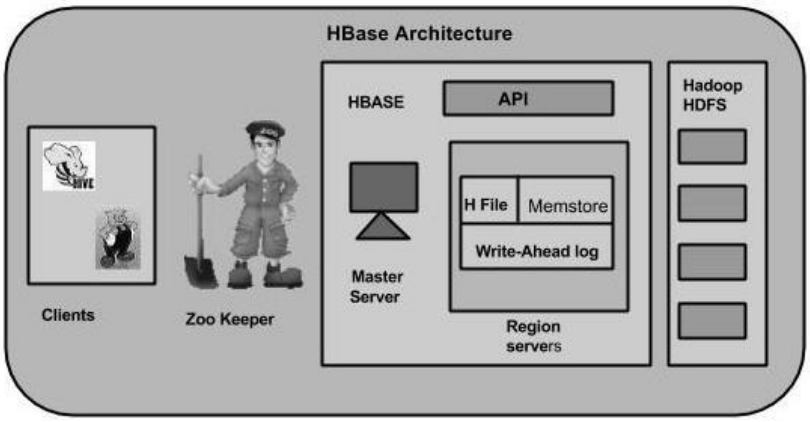

Figure 1. HBase Architecture[1]

Hbase has three components: the client, library, a master server and region servers. Regions servers can be added or removed as per requirement.

Master Server

The master server-

- Itassigns regions to the region servers and takes the help of apache. Zookeeper for this task[5].
- Zookeeper manages everything. It tackles load balancing of the regions across region servers. It unloads the busy servers and shifts the regions to less occupied servers.
- Sustains the state of the cluster by negotiating the load balancing.
- It is responsible for schema changes and other metadata operations such as creation of tables and column families.

2.1. Regions

- Regions are nothing but tables are split up and spread across the region servers.
  Region server:
  The region servers have regions that –
- Interact with the client and tackle data-related operations.
- Manages read and write request for all the regions under it.
- Decides the size of the region by following the region size thresholds[5].

When we take a deeper look into the region server, it contain regions and stores as shown below:

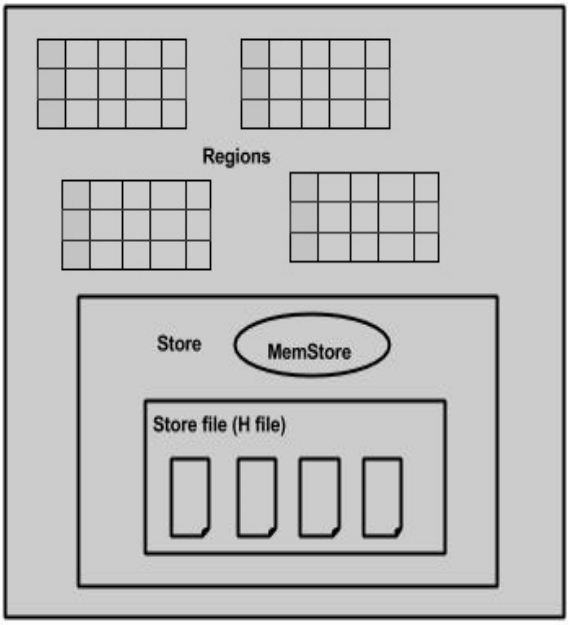

Figure 2. Region server[1]

As shown in figure, the store contains memory store and hfiles. Memstore is just like a cache memory. Anything that is entered into the hbase is stored here initially[1]. Later, the data is transferred and saved in Hfiles as blocks and the memstore is flushed[1].

2.2. Zookeeper

- Zoo keeper instance is started along with the HBase and keep track of all region servers in HBase. Information such as how many region servers are there and which region servers are holding from which data node to which data node one specialty is it keeps track of smaller data sets which Hadoop does not do[5].
- It decreases the overhead on top of Hadoop by avoiding maintaining the Meta. The HMaster by actually contacting Zoo keeper it gets the details of
- Zookeeper is an open-source project that provides services like maintaining configuration information, naming, providing distributed synchronization etc[5].
- Zookeeper has ephemeral nodes representing different region servers. Master servers use these nodes to discover available servers[1].
- In addition to availability, the nodes are also used to track server failures or network partitions.
- Clients communicate with region servers via zookeeper.
- In pseudo and standard modes, Hbase itself will take care of zookeeper[1].

2.2.1. HBase Shell

HBase contains a shell using which you can communicate with Hbase.
Hbase uses the hadoop file system to store its data. It will have a master server and region server. The data storage will be in the form of regions.
These regions will be split up and stored in region servers[1].
The master server manages these region servers and all these tasks take place on HDFS. Given below are some of the commands supported by Hbase shell.

2.2.2 General Commands

- Status: Provides the status of Hbase, for example the number of servers.
- Version: Provides the version of Hbase being used.
- Table-help: Provides help for table – reference commands.
- Whoami: Provides information about the user.

2.3. Data Definition Language

These are the commands that operate on the table in Hbase.

- create: Creates a table.
- list: Lists all the tables in HBase.
- disable: Disables a table.
- is_disabled: Verifies whether a table is disabled.
- enable: Enables a table.
- is_enabled: Verifies whether a table is enabled.
- describe: Provides the description of a table.
- alter: Alters a table.
- exists: Verifies whether a table exists.
- drop: Drops a table from HBase.
- drop_all: Drops the tables matching the 'regex' given in the command.
- Java Admin API: Prior to all the above commands, Java provides an Admin API to achieve DDL functionalities through programming[1]. Under org.apache.hadoop.hbase.client package, HBaseAdmin and HTableDescriptor are the two important classes in this package that provide DDL functionalities[1].

2.4. Data Manipulation Language

- put: Puts a cell value at a specified column in a specified row in a particular table.
- get: Fetches the contents of row or a cell.
- delete: Deletes a cell value in a table.
- deleteall: Deletes all the cells in a given row.
- scan: Scans and returns the table data.
- count: Counts and returns the number of rows in a table.
- truncate: Disables, drops, and recreates a specified table.
- Java client API: Prior to all the above commands, Java provides a client API to achieve DML functionalities, CRUD (Create Retrieve Update Delete) operations and more through programming, under org.apache.hadoop.hbase.client package. HTable Put and Get are the important classes in this package[1].

APIHBase is written in java, therefore it provides java API to communicate with HBase. Java API is the fastest way to communicate with HBase. Given below is the referenced java Admin API that covers the tasks used to manage tables.

Class HBaseAdmin:HBaseAdmin is a class representing the admin. This class belongs to the org.apache.hadoop.hbase.client package[1]. Using this class, you can perform the tasks of an administrator. You can get the instance of admin using connection.getAdmin() method.

Hbase Applications

- It is used whenever there is a need to write heavy applications.
- Hbase is used whenever we need to provide fast random access to available data.
- Companies such as Facebook, twitter, yahoo, and adobe use HBase internally.

III. CONCLUSION

In this paper we conclude that hbase can access heavy applications. It is used to provide fast random access to available data with very low latency. Also zookeeper manages everything in hbase. However it is also observed that hbase doesn't support joins. It stores limited real time queries and limited sorting. It is difficult to store large binary data. Hbase requirements are very expensive because it needs at least 5 servers(zookeeper, region server, name nodes, data nodes, etc).However it deals with large data sets ability to aggregate and analyze billion of rows with a very short span of time.It deals with structured and semi-structured data, however work should be done regarding its cost and sorting issues.

IV. REFERENCES

[1]. Tutorials point, "apache Hbase", 2014.
[2]. WPI, Mohamed Eltabakh, "CS525: Special Topics in DBs Large- Scale Data Management, Spring 2013.
[3]. Priyanka Raichand, Rinkle Rani Aggarwal, "A Short Survey [3]of Data Compression Techniques for column oriented databases", Journal of Global Research in Computer Science, Volume 4, No.7, July 2013.
[4]. Gkavresis Giorgos -1470, " Introduction to HBase".
[5]. R.saraswathay, p.priyadarshini, p.sandeepa, "HBase cloud research architecture for large scale image processing", "international journal of advanced research in computer science and software engineering, volume 4, issue12, december 2014.
[6]. Amandeep Khurana, "Introduction to HB ase Schema Design", OCTOBER 2012

[7]. S. Babu and H. Herodotou, "Massively Parallel Databases and MapReduce Systems", published in Foundations and Trends in Databases Vol. 5, No. 1 (2012) 1–104, DOI: 10.1561/1900000036
[8]. Fay Chang, Jeffrey Dean, Sanjay Ghemawat, Wilson C. Hsieh, Deborah A. Wallach, Mike Burrows, Tushar Chandra, Andrew Fikes, Robert E. Gruber, "Bigtable: A Distributed Storage System for Structured Data", OSDI 2006

