Handling Unstructured Databases on Cloud Using Hadoop

Harshal Shastri¹, Ashwin Prajapati²

¹CE, HGCE, GTU, Ahmedabad, Gujarat, India
²Project Guide, Asst. Prof., CE, HGCE, GTU, Ahmedabad, Gujarat, India
Email- harshal.shastri@yahoo.in, head024@gmail.com

Abstract: Unstructured database is defined as that in which we cannot store data in form of table i.e. rows and columns. So to handle that types of data we are going to store NoSQL i.e. KV (key value) store database. Everyone can handle big or huge data using Hadoop technology by creating a Hadoop cluster on cloud and also can perform CRUD operations on data. On cloud, we can store any type of data whether it is relational or non-relational data. This provides access of data to all users as it is open source and is implemented in java.

KeyWords: Bigdata, Hadoop, oracle cloud, CRUD.

I. INTRODUCTION

Honestly speaking there is nothing new under the Sun. Integration has been a problem that we have faced for 40-plus years in IT. But there are many new ways to handle a problem. there are many new ways to handle a problem. So the role is shaped from SOA (Service Oriented Architecture) on the cloud. using hadoop technology.

II. OVERVIEW OF UNSTRUCTURED DATABASE

Unstructured data is information, in many different forms, that doesn't how to conventional data models and thus typically isn't a good fit for a mainstream relational database. One of the most common types of unstructured data is text. Unstructured text is generated and collected in a wide range of forms, including Word documents.

A. Unstructured database:

The worst NoSQL database stores unstructured data only. Coming to question which is the best NoSQL database Well it depends on the product and the scale. Most of the NoSQL databases are scalable. NoSQL are of many types like

B. Types of NoSQL database

Document based - mongodb is an example which I really love. They store data in the JSON format. Companies like Hike uses mongodb. At ChalkStreet we use mongodb for storing discussions and tracking user activity.

Key-value based - Redis is the best example. Its similar to memcache. Used for the quick data retrieval based on key. Can also be used for sending push notifications.

Column based - Like apache cassandra. Facebook uses cassandra for messages. This makes messages easy to search despite the length-of conversations.

Graph based - Neo4j is the best example. At ChalkStreet we use neo4j for relating users to their courses, articles and topics and in turn to other users. This makes identifying and learning user very effective. We can give better In this article we uses Key-value based database.

C. overview of KV database.

KV-value Database is also called KV-Store Database. It is one type of NoSQL database. In KV-store there is no Schema required for store data. values are required and access via key and store value which can be number, string, image or videos etc. It is most flexible model of NoSQL because the application has complete control over what is stored in value.

D. How types of data can be stored in a key-value database:

The key-value part refers to the fact that the database stores data as a collection of key/value pairs. This is a simple method of storing data, and it is known to scale well.

- Key: The key in a key-value pair must (or at least, should) be unique. This is the unique identifier that allows you to access the value associated with that key.
What is key-component: A key component is a Java String. Issues of comparison can be answered by examining how Java Strings are compared using your preferred encoding. Because it is a String, a key component can be anything you want it to be.

Key major: email address.
Key minor: various properties, such as the user's street address, phone number, photograph, and name.
Value: The combination of major and minor key components. So, for example, the value for an email address plus a street address might be multiple fields related to street number, street name, city, and so forth. If we can write a records components as. If file system path delimited by a ("<>") For example if you used multiple major components to identify a record, and one such record using the following major components: "Lenovo", and "HP." Another record might use "Lenovo" and "Dell". And a third might use "Acer", and "Toshiba". Then the major components for those records could be written.

III. OVERVIEW OF HADOOP

Nutch started the concept in 2002, as a framework for building horizontally scalable search engines and internet crawlers. Earlier it was considered to be as an open-source web-search engine with its source code is written in Java. In year 2006, Hadoop became a standalone technology and is now used for creating clusters on cloud that handles large amount of data. In year 2008, Hadoop broke world record for storing 1 TB of data.

A. Hadoop Architecture

Hadoop Architecture composed of two layers:
1. HDFS layer
2. Map Reduce layer

HDFS layer

HDFS cluster operates in a master slave pattern a Namenode is a master and number of Data nodes are slave. a user can buy a whole bunch of commodity hardware to run on hadoop on individual
Architecturally, it works on 5 nodes and are as follows:
1) Namenode (Primary Namenode): Name Node provides metadata storage for the shared file system. The Name Node is master of HDFS that directs the slave Data Node daemons to perform Input/output tasks. Name Node is single point of failure of Hadoop cluster. For any of the other daemons except Name node if host nodes fails, the Hadoop cluster will continue to function smoothly or can quickly restart it.
2) Data Node: to perform reading and writing HDFS blocks to real file on local file system. while reading or writing HDFS blocks and the name node tells the client which data node each block resides in A data node may communicate with other data node to replications it data blocks for redundancy. Files are distributed among the data node continually tells the Name Node to provide information about the local changes made on the disk.
3) SecondaryNameNode: Like Name Node, each cluster has one SNN(secondary name node), residing on its own machine as well. unlike Namenode

Figure 1 Hadoop Architecture
The SNN does not record any real-time changes to HDFS instead, it gets just like HDFS metadata from Name Node at intervals. SNN snapshots help minimize the downtime and loss of data.

4) **Job tracker**: Job Tracker determines which files to process, assigns nodes to tasks and monitors all tasks. If any task fails, the job tracker will automatically start the task again, probably on a different node for a limited number of retries. There is only one job tracker daemon per Hadoop cluster. It is also called the heart of Hadoop.

5) **Task tracker**: Task Tracker manages tasks executing on each slave node. Each task is assigned by the job tracker. Task tracker constantly keeps communicating with the job tracker. Failure in receiving a pulsation from a task tracker has crashed and resubmits the corresponding tasks to other nodes in the cluster.

**Map Reduce function**

Map-Reduce is a parallel data processing model introduced by Google dividing computation in two functions, Map and Reduce. In Map-Reduce programming model, a MapReduce job consists of a map function, a reduce function. When a function is called, the below steps of actions take place. Map-Reduce will first divide the data into N partitions with size varies from 16MB to 64MB. Then it will start many programs on a cluster of different machines.

The main steps Map-Reduce would take to process the data would be:
- Get the input data.
- Split the data into separate blocks.
- Assign the blocks to Map tasks.
- Sort the output of the Map tasks.
- Reduce the sorted data using the Reduce tasks.

**IV. HADOOP CLUSTER**

Hadoop cluster is a special type of computational cluster designed for storing and analyzing vast amounts of structured and unstructured data.

**A. Core Component of Hadoop Cluster:**
1) Client
2) Master
3) Slave

**Client:**
It is neither master nor slave, rather play a role of loading the data into the cluster, submit MapReduce jobs describing how the data should be processed and then retrieve the data to see the response after job completion.

**Masters:**
Master consists three components: Namenode, Secondary Namenode, Job tracker.
Name node: Name node does not stores the files metadata. In later section we will see it is actually the Datanode which stores the file.

Secondary Name node: Secondary Namenode is not the backup or high arability node for SecondaryNameNode is not backup rather it doesn’t job of housekeeping . in the case of Namenode failed, saved metadata built easily.

Slaves:
- Slave nodes are the majority of machines in Hadoop Cluster and are responsible to
  - Store the data
  - Process the computation

V. HADOOP CLUSTER ON ORACLE CLOUD USING NOSQL DATABASE

A. Starting a Cluster

Whenever we are going to start a hadoop cluster, we have to run a following command:

```
s$ bin/hadoop
```

When we run this command this will display the documentation for hadoop script.

A Hadoop cluster can be started in following three modes:

a. Standalone (Local) mode: A Hadoop cluster can be started in a local (standalone mode). Local mode is defined as that in which hadoop is configured to run as a single Java process. That is why Local mode or Standalone mode is also called as a non-distributed mode.

b. Pseudo-Distributed mode: Pseudo-Distributed mode is one in which Hadoop is considered to be running on a single node like the Local mode but with one difference that each daemon of Hadoop runs Java process separately.

c. Fully-Distributed mode: Setting up of information on multiple nodes by configuring multiple clusters is said to be in a Fully-Distributed mode. Fully-Distributed is considered to be the opposite of local mode.

VI. RUNNING HADOOP CLUSTER

- To run all the nodes discussed in chapter 6 to start a cluster we need a platform that must be UNIX based (For e.g.: Ubuntu, Cygwin, etc.)

- Commands To Run On Cygwin (Cygwin is a platform that provides UNIX like environment on windows) TO START CLUSTER

  A. Start the NameNode with the following command:

    ```
    >cd hadoop-0.20.1+169.127
    >bin/hadoop namenode
    ```

  B. Start the Secondarynode with the following command:

    ```
    >cd hadoop-0.20.1+169.127
    >bin/hadoop secondarynamenode
    ```

  C. Start the datanode:

    ```
    >cd hadoop-0.20.1+169.127
    >bin/hadoop datanode
    ```

  D. Start the jobtracker

    ```
    >cd hadoop-0.20.1+169.127
    >bin/hadoop jobtracker
    ```

  E. Start the tasktracker

    ```
    >cd hadoop-0.20.1+169.127
    >bin/hadoop tasktracker
    ```

- After running hadoop cluster connecting with KV store database

VII. CONCLUSION

In this we can handle big or huge data using Hadoop technology by creating a Hadoop cluster on cloud and also can perform CRUD operations on data. On cloud, we can store any type of data whether it is relational or non-relational data. This provides access of data to all users as it is open sourced and is implemented in java.

REFERENCES


[6] Big Data using KV-Store databases ijarcsse