BVG Trust

J. Sci. Trans. Environ. Technov. 2016, 9(3) : 125-130

Scientific Transactions in Environment and Technovation

Analysis of Big Data using Hadoop ecosystem components

K. Vijaya kumar*and M.V Srinath

Department of Computer Science Engineering, Vignan's Institute of Engineering for Women, Visakhapatnam, Andhra Pradesh, India.

Department of Master of Computer Applications, Sengamala Thayaar Educational Trust Women's College, Sundarakkottai - 614 016, Mannargudi, Tamilnadu, India.

Abstract

In enormous information world, Hadoop Distributed File System (HDFS) is extremely well known. It gives a system to putting away information in a conveyed situation furthermore has set of instruments to recover and prepare. These information set utilizing guide decrease idea. In this paper, an intensive examination has been conveyed to talk about that how enormous information investigation can be performed on information put away on Hadoop dispersed document framework utilizing Pig and Hive. Apache Pig and Hive are two undertakings which are layered on top of Hadoop, and give more elevated amount dialect to utilize Hadoop's Map Reduce library. In this paper, as a matter of first importance, the fundamental ideas of Map Reduce, Pig and Hive are presented and their performance comparison.

Keywords: framework, hadoop distributed file system, Hive, Map reduce, Pig

INTRODUCTION

In recent years, big data has rapidly developed into a hotspot that attracts great attention from academia,Industry, and even governments around the world (Cuzzocrea, 2014 and Mayer-Schonberger and Cukier 2013) .Big data is mainly collection of data sets so large and complex that it is very difficult to handle them using on-hand database management tools. The main challenges with big databases include creation, curtain, storage, sharing, search, analysis and visualization. So to manage these databases we need, "highly parallel software's". First of all, data is acquired from different sources such as social media, traditional enterprise data or sensor data etc. Flume can be used to acquire data from social media such as twitter. Then, this data can be organized using distributed file systems such as Google File System or Hadoop File System. These file systems are very efficient when number of reads are very high as compared to writes. At last, data is analyzed using map reduce so that queries can be run on this data easily and efficiently. Big data analytics is a relatively new analytics paradigm that is used to analyze datasets, which cannot be managed or processed with the currently available technologies. Currently these technologies are being adopted by a variety of industries. Big Data mining is the capability of extracting meaningful and useful information from the huge datasets and streams of data (*Wu et al, 2014*). Big data analytics is the process of uncovering hidden patterns, unknown correlations, and other useful information from the big data, that can be used to make better decisions and if managed well, it can deliver powerful insights. Novel analytic techniques are required mainly because of the volume, velocity and variety of such data produced every day (Abe, 2013). By "system monitoring" we mean the process of detecting failures, miss configurations and poor performance of the systems. Monitoring is extremely critical for large distributed systems, where the large amount of data makes it important for the system to detect failures and faulty behaviors. The amount of data generated in the modern world increase at a rate of 40% each year, which makes sorting through information insurmountable through old data methods*(Fan, et al., 2014*). In the recent years, the aviation industry has witnessed a steady increase in big data (volume, velocity, variety) that are collected and streamed from thousands of aircraft (including operational datasets and maintenance data) and test cells (that includes hundreds of sensors and their measurement). For instance, one test cell can generate 300 MB engine test data daily. For the aircraft system monitoring data alone, 5 ~ 10 MB per flight hour per aircraft are routinely collected. With over 20 TB data sources, these connected aircrafts (a part of Industrial Internet) hold the potential of providing valuable knowledge needed to maintain profitability (*Ebner, et al. , 2014*). Technically, General Electric (GE) estimates a $250M savings in engine maintenance cost using insights gained from ML and data mining (Evans, and Annunziata, 2013). Big Data processing challenges needs to be overcome before realizing these benefits. Figure 1 showing the Hadoop ecosystem.

COLLECTION OF DATA

First of all, data has to be acquired from different sources. Main sources of data are:

Traditional Organization data – it includes customer info from CRM systems, transactional ERP data or web store transactions and general ledger data.

*Corresponding Author :
email: *vijay_kollati@yahoo.co.in*

P - ISSN 0973 - 9157
E - ISSN 2393 - 9249
January to March 2016

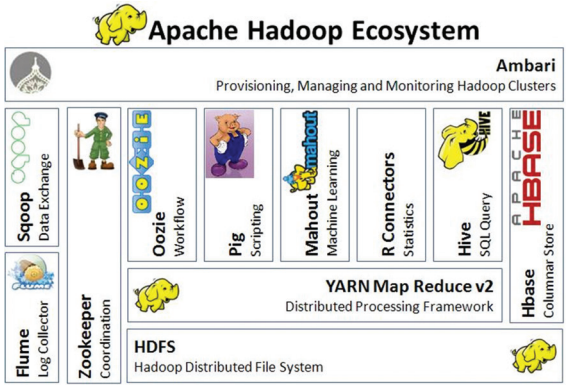

Figure1. Hadoop Ecosystem

Machine generated or sensor data – it includes Call Detail Records, smart meters, weblogs, sensors, equipment logs and trading systems data.

Social data – it includes customer feedback stream and micro blogging sites such as Twitter and social media platforms such as Face book.

Flume

Data from social media is generally acquired using flume. Flume is an open source software program which is developed by cloud era to act as a service for aggregating and moving very large amount of data around a Hadoop cluster as data is produced or shortly thereafter. Primary use case of flume is to gather log files from all machines in cluster to persist them in a centralized store such as HDFS. In it, we have to create data flows by building up chains of logical nodes and connecting them to source and sinks. For example, if you want to move data from an apache access log into HDFS then you have to create a source by tail access.log and use a logical node to route this to an HDFS sink. Most of flume deployments have three tier designs. The agent tier have flume agents collocated with sources of data which is to be moved. Collector tier consist of multiple collectors each of which collect data coming in from multiple agents and forward it on to storage tier which consist of file system like HDFS or GFS.

A Flume agent is a JVM process which has 3 components -Flume Source, Flume Channel and Flume Sink- through which events propagate after initiated at an external source. Figure 2 demonstrating the working of flume.

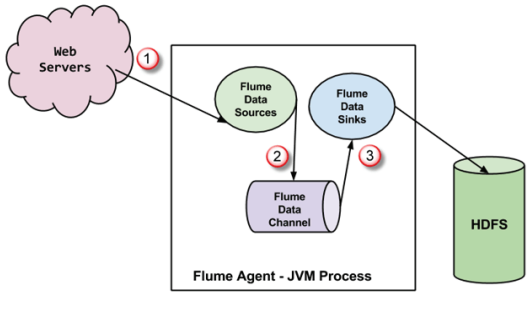

Figure 2. Working of Flume

In above diagram, the events generated by external source (Web Server) are consumed by Flume Data Source. The external source sends events to Flume source in a format that is recognized by the target source.

Flume Source receives an event and stores it into one or more channels. The channel acts as a store which keeps the event until it is consumed by the flume sink. This channel may use local file system in order to store these events.

Flume sink removes the event from channel and stores it into an external repository like e.g., HDFS. There could be multiple flume agents, in which case flume sink forwards the event to the flume source of next flume agent in the flow.

ORGANIZE DATA

After acquiring data, it has to be organizing using a distributed file system. First of all, we have to break this data into fixed size chunks so that they can store and access easily. Mainly we use GFS and HDFS file systems.

Google File System

Google Inc. developed a distributed file system for their own use which was designed for efficient and reliable access to data using large cluster of commodity hardware. It uses the approach of "Big Files", which are developed by Larry Page and Sergey Brin. Here files are divided in fixed size chunks of 64 MB. It has two types of nodes- one master node and many chunk server nodes.

Files in fixed size chunks are stored on chunk servers which are assigned a 64 bit label by master at creation time. There are at least 3 replication for every chunk but it can be more. Master node doesn't have data chunks; it keeps the metadata about chunks such as their label, their copy locations and their reading or writing processes. It also has the responsibility to replicate a chunk when its copies become less than three. Figure 5 shows the architecture of GFS is following.

Hadoop Distributed File System

Hadoop distributed file system is a distributed, scalable and portable file system which is written in java (ibm.com. IBM. Retrieved 2014). All machines which support java can run it. In it, every cluster has a single name node and many demands. A data node has many blocks of same size except last block which have different size. It does communication using TCP/IP layer but clients uses RPC to communicate with each other. Every file in HDFS has a size of 64 MB or multiple of 64 MB. Reliability is due to replication of data. At

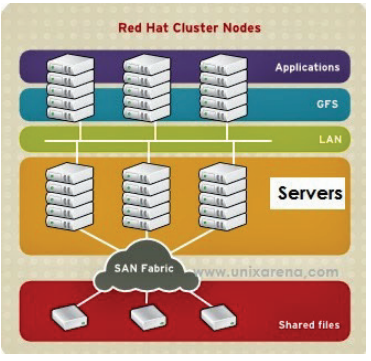

Figure 3 GFS

least 3 copies of every data node are present. Data nodes can communicate with each other to rebalance data or copy data or to keep high replication of data. HDFS has high availability by allowing name node to be manually failed over to backup in case of failure. Now days, automatic failover is also developing. It also uses a secondary name node which continuously takes the snapshots of primary name node so that it can be active when failure of primary node occurs. Data awareness between task tracker and job tracker is an advantage. Job tracker can schedule map reduce job to task tracker efficiently due to this data awareness. The core of Apache Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part called Map Reduce. Hadoop splits files into large blocks and distributes them across nodes in a cluster. To process data, Hadoop transfers packaged code for nodes to process in parallel based on the data that needs to be processed. This approach takes advantage of data locality (Russom, 2011) — nodes manipulating the data they have access to— to allow the dataset to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel file system where computation and data are distributed via high-speed networking (Sanjay Ghemawat, Howard Gobioff, and Shun-Tak Leung, 2003). The base Apache Hadoop framework is composed of the following modules: Hadoop Common – contains libraries and utilities needed by other Hadoop modules;

Hadoop Distributed File System (HDFS) – a distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster;

Hadoop YARN – a resource-management platform responsible for managing computing resources in clusters and using them for scheduling of users' applications; (Murty, 2012) .

Hadoop Map Reduce – an implementation of the Map Reduce programming model for large scale data processing.

The term Hadoop has come to refer not just to the base modules above, but also to the ecosystem, or collection of additional software packages that can be installed on top of or alongside Hadoop, such as Apache Pig, Apache Hive, Apache HBase, Apache Phoenix, Apache Spark, Apache Zookeeper, Cloud era Impala, Apache Flume, Apache Sqoop, Apache Oozie, Apache Storm. (Sanjeev Dhawan and Sanjay Rathee, 2013)

Apache Hadoop's Map Reduce and HDFS components were inspired by Google papers on their Map Reduce and Google File System .

The Hadoop framework itself is mostly written in the Java programming language, with some native code in C and command line utilities written as shell scripts. Though Map Reduce Java code is common, any programming language can be used with "Hadoop Streaming" to implement the "map" and "reduce" parts of the user's program (John Wiley and Sons, 2014).Other projects in the Hadoop ecosystem expose richer user interfaces. Figure 4 showing the HDFS.

ANALYZE DATA

After organizing data, it has to be analyzing to get fast and efficient results when a query is made. Map reducer's are mainly used to analyze data. Map reducer, Pig and Hive are very efficient for this purpose.

Setup for Analysis

An analysis is performed on a big database of 8 lakh records using Pig, Hive and Map Reduce. For this purpose, we install Hadoop, Pig, Hive on cloud era. And analysis time is calculated for each (Hadoop.apache.org. Retrieved 2013) . Figure 5 shows the temperature datasets present in the HDFS. From the temperature datasets we are generating the maximum temperature from the year 1900 to 2014.

Map Reduce

Hadoop Map Reduce is a software framework for easily writing applications which process vast amounts of

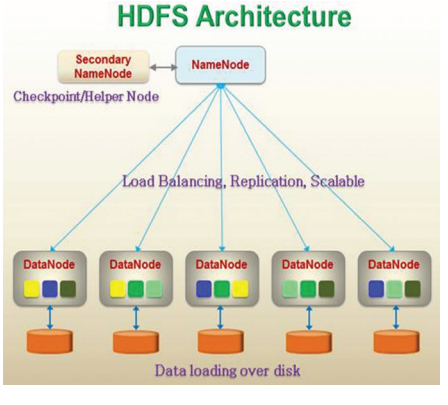

Figure 4 HDFS

data (multi-terabyte data-sets) in-parallel on large clusters (thousands of nodes) of commodity hardware in a reliable, fault-tolerant manner.

A Map Reduce *job* usually splits the input data-set into independent chunks which are processed by the *map tasks* in a completely parallel manner. The framework sorts the outputs of the maps, which are then input to

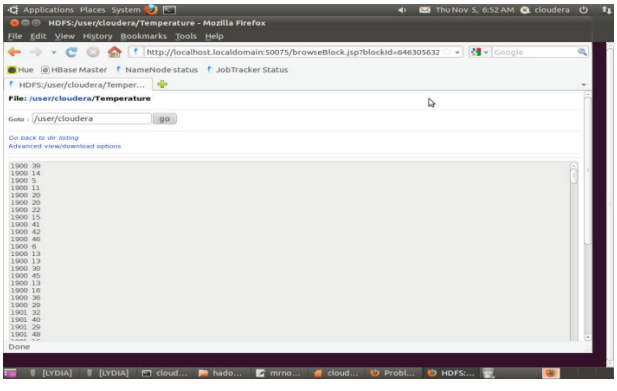

Figure 5. Temperature dataset in the HDFS

the *reduce tasks*. Typically both the input and the output of the job are stored in a file-system. The framework takes care of scheduling tasks, monitoring them and re-executes the failed tasks.

Typically the compute nodes and the storage nodes are the same, that is, the Map Reduce framework and the Hadoop Distributed File System are running on the same set of nodes. This configuration allows the framework to effectively schedule tasks on the nodes where data is already present, resulting in very high aggregate bandwidth across the cluster.

The Map Reduce framework consists of a single master Job Tracker and one slave Task Tracker per cluster-node. The master is responsible for scheduling the jobs' component tasks on the slaves, monitoring them and re-executing the failed tasks. The slaves execute the tasks as directed by the master.

Minimally, applications specify the input/output locations and supply *map* and *reduce* functions via implementations of appropriate interfaces and/or abstract-classes. These, and other job parameters, comprise the *job configuration*. The Hadoop *job client* then submits the job (jar/executable, etc.) and configuration to the Job Tracker which then assumes the responsibility of distributing the software/configuration to the slaves, scheduling tasks and monitoring them, providing status and diagnostic information to the job-client. Figure 6 demonstrating the execution of Map Reduce and Figure 8 showing the status report of job tracker and 100% completion of map and reduce jobs.

Pig

Pig is an Apache Software Foundation project which was initially developed at Yahoo Research in 2006. Pig has a language and an execution environment. Pig Latin which is a dataflow language is used by Pig (Ashish Thusoo, et al. ,2013). Pig Latin is a type of language in which you program by connecting things together. Pig

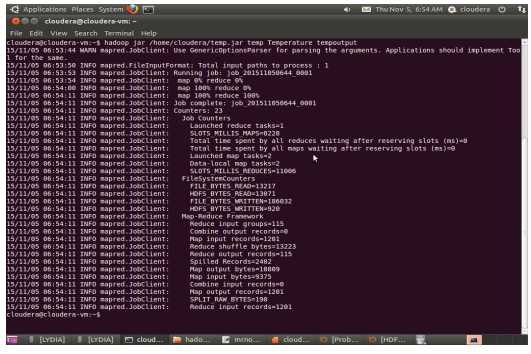

Figure 6. Execution of Map Reduce

Figure 7. Output screens, showing the maximum temperature from the year 1900 to 2014

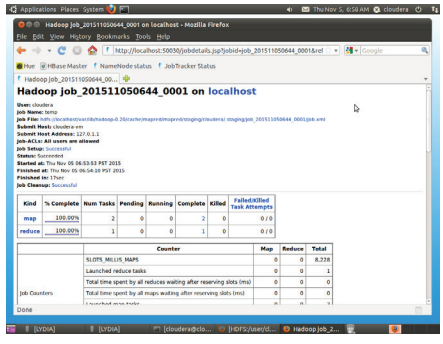

Figure 8. Showing the status of job tracker and 100% completion of map and reduce jobs.

can handle complex data structure, even those who have levels of nesting. It has two types of execution environment local and distributed environment. Local environment is used for testing when distributed environment cannot be deployed. Pig Latin program is

collection of statements. A statement can be an operation or command. Here is a program in Pig Latin to analyze a database. Figure 9 shows the pig's grunt shell and Figure 10 shows the Loading of temperature datasets to temp relation to perform the temperature analysis with less time than by writing map reduce code.

Hive

Hive is a technology which is developed by Face book and which turns Hadoop into a data warehouse complete with an extension of sql for querying. Hive SQL which is a declarative language is used by Hive. In pig Latin, we have to describe the dataflow but in Hive results must be describe. Hive itself find out a dataflow to get those results. Hive must have a schema but it can be more than one.

Hive must be configured before use. It can be configured in three different ways:

By editing a file hive-site.xml,

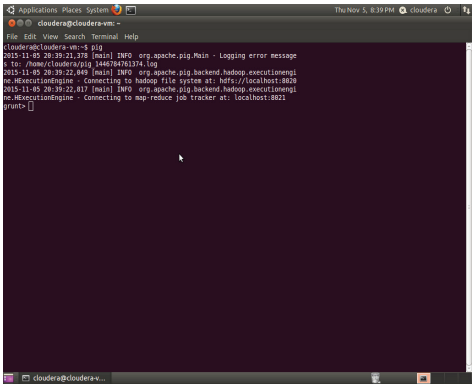

Figure 9. Pig's grunt shell

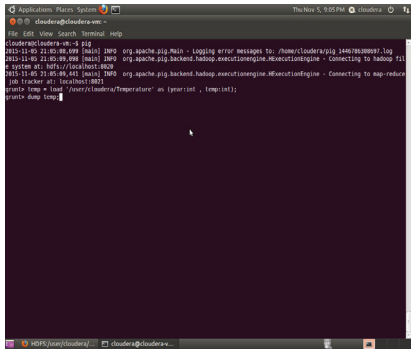

Figure 10. Loading of temperature datasets to temp relation

By hive conf option in Hive command shell

By using set command.

Here we have a database with more than eight lakh records which is analyzed by using Hive to get a temperature.

Creating Database

Create database temp;

Use temp;

Create table for storing temp records

Create table temps(year INT, year INT)

Load the data into the table

LOAD DATA LOCAL INPATH '/home/cloudera/hive/data/Temperature.csv'

OVERWRITE INTO TABLE temps;

Describing metadata or schema of the table

Describe temps

Selecting data

*Select * from temps;*

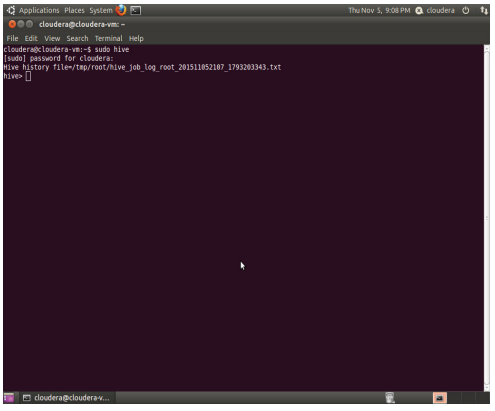

Figure 11. Hive Screen

To perform the analysis on temperature data sets, we also used hive along with pig and map reduce coding, using hive we performed the analysis by using above HIVE Query Language commands. Therefore the research paper, demonstrates that, to perform the quick analysis HIVE and PIG can be used than map reduce, where we need to write lengthy coding, which is absent in the hive and pig.

CONCLUSION

It is not possible to handle big data using traditional database management systems like relational databases. So we use some highly parallel software to handle big databases. Some components are also used to handle them. Firstly we have to acquire data from different sources which can be easily done by using components like Flume. Flume can directly fetch tweets from websites like twitter and store them in Bid databases. Then we can organize them using distributed file system like GFS HDFS. At last they can be analyzed for faster access and query response. After analysis