INTERNATIONAL JOURNAL OF PURE AND APPLIED RESEARCH IN ENGINEERING AND TECHNOLOGY

A PATH FOR HORIZING YOUR INNOVATIVE WORK

A COMPREHENSIVE VIEW OF HADOOP

ER. AMRINDER KAUR

Assistant Professor, Department of Computer Science & Engineering, University Institute of Engineering & Technology, Kurukshetra University, Haryana.

Accepted Date: 24/09/2015; Published Date: 01/10/2015

Abstract: - As data is evolving day by day with tremendous amount like exabytes ($2^{18)}$, at high speed with various varieties like financial data, weather forecasting, social media, email and list go on. This vast amount of data is not warehoused on one site and therefore, a need of a framework required that dispense the data across multiple clusters and provide distributed computing to answer the queries. Hadoop is the solution of above discussed points. Hadoop is open source and fault tolerant. Hadoop provide high available services to the cluster of computers.

Keywords: Hadoop, map reduce, client-server, task manager, job tracker

Corresponding Author: ER. AMRINDER KAUR

Access Online On:

www.ijpret.com

How to Cite This Article:

Er. Amrinder Kaur, IJPRET, 2015; Volume 4 (2): 122-128

PAPER-QR CODE

1. INTRODUCTION

Hadoop is free ware; its programming is based on java framework. It supports distributed computing environment in which processing of large data set is to be done. Hadoop is batch oriented where jobs are queued and then executed, and processing of jobs may take minutes or hours. The basic storage mechanism in Hadoop is Hadoop Distributed File System (HDFS) [1] The MapReduce framework is proposed by Google.

The framework is responsible of everything else such as parallelization, fail-over etc. With Hadoop's distributed file system, mapreduce framework read and writes its data. Usually, Hadoop MapReduce uses the distributed file system of hadoop known as HDFS, is the open source complement of the Google File System (GFS). Therefore, Hadoop MapReduce job's input and output performance strongly depends on HDFS.

2. ARCHITECTURE

Hadoop is open source framework this is composed of hadoop distributed file system and map reduce engine. Hadoop is scalable fault-tolerant distributed system for processing of data and forage. Hadoop provide framework for analysis and transformation of extreme data using map reduce paradigm data storage. Hadoop provide framework for analysis and transformation of extreme data using map reduce paradigm.[3][5]

2.1. Hadoop Distributed File System

HDFS is responsible for managing the data or files which is present on different clusters. Meta data of a file and application data which is needed during job are stored separately. Meta data is stored on name node; a dedicated server. Application data are stored on data node; other servers. These servers are connected to each other & communication done with the help of TCP based protocols. In HDFS Raid structure is used to provide data durability. For reliability file content is duplicated on multiple data nodes [2][3][6]. HDFS structure is shown in figure below

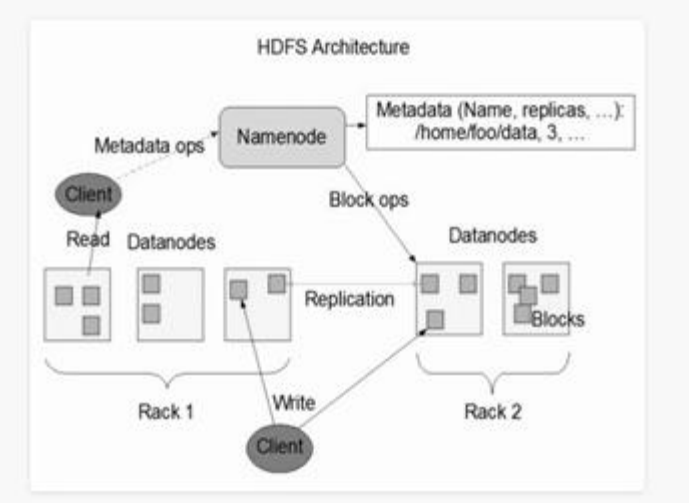

Figure: HDFS's Architecture

2.1.1. Name Node

Responsibility of name node is for maintaining the directory structure of HDFS. It is also known as namespace. On namenode inode is used to represent directory and files and it also record attributes like permissions, access times, modification etc. Namenode only maintains mapping between blocks of file and the datanodes on which blocks are stored. Each block of the file is replicated on multiple datanode. For performing any operation on file (like open close, delete, rename) initially client contact the namenode. For example to perform a read operation on a file, a client first contact the namenode for the location of datanode and after that read the data from the closest datanode to the client. And when clients want to write the data onto a file it requests the namenode to nominate three datanodes which contains the block replicas. And writing is performed in a pipeline fashion. Currently a single name node is nominated for each cluster. But datanodes can be hundreds or thousands and may execute multiple tasks concurrently.[2][3][6]

2.1.2. Data Node

Datanode is used to hold the block replica and these block replica is represented by two files in the local host native file system. First file contains the data itself and second file contains metadata of block. Except name node all other node will act as a datanode. Each node hold file blocks on the behalf of local or remote host. On the request of name node blocks are created or destroyed on data nodes. Name node is responsible for validating and processing requests from clients. Clients communicate directly with datanode for data in order to read or write data at

HDFS block level. In startup phase datanode connects to namenode and perform handshake. Handshake is done to verify the namespace ID and software version of datanode. If ID doesn't match with namenode then datanode automatically shuts down. Each datanode sends heartbeat signal within few seconds and if it fails to send these signals then it will be considered as out of service and namenode find other datanodes for block replica. [2][3][6]

2.2. Map Reduce

Hadoop Map Reduce framework is one of the popular implementation of map reduce framework which is proposed by google. It become popular because it is easy to use, scalable and fault tolerent. It is used for processing big data in industry and academia also. It consist of two functions i.e. map and reduce. Both these functions are user defined. The map function take input in the form of (k,v) where k refers to key and v refers to value. After that map function is applied on each pair of (k,v). After that it will generate intermediate key value pair which is showed as (k', v'). Iteratively on each intermediate key value pair reduces function is called and after that reduce function merge all the intermediate values on the basis of a single key. [4][5]

Map Reduce Architecture is shown in figure below.

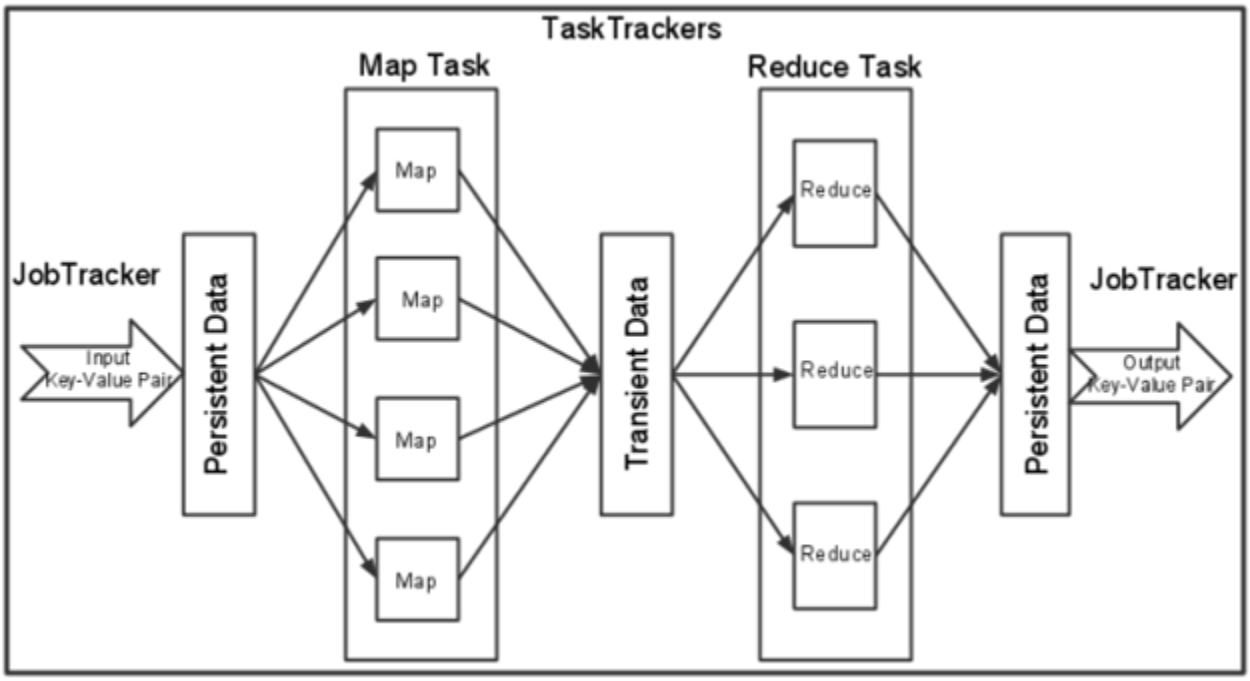

Figure: Map Reduce Architecture

2.2.1. Job Tracker

Job tracker is responsible for maintaining the list of the processing resources available in the clusters. A Job tracker run on master node and it is responsible for distributing the different map reduce jobs into the cluster. When there is a request for job then it schedules the job and assigns the job to the task tracker running on the datanodes.

- Initially client node submits its job to job tracker
- Then job tracker is incharge of determining the location of data in datanode.
- After locating the datanode its corresponding task tracker node is located which is nearest to the data or which have available slots.
- Job is then assigned to task tracker node.
- Tasktracker nodes are monitored continuously and if they don't respond with heartbeat signals then they are considered as failed and the job is scheduled on some other task tracker.
- Due to some reason if job fails then task tracker notify the jobtracker . Then job tracker will decide whether to submit the job somewhere else or to restart the job on same task tracker node.
- And on the completion of job, jobtracker updates its status about a particular job. And then client node asks the job tracker for information. The diagram below show the cluster setup in the network. [7]

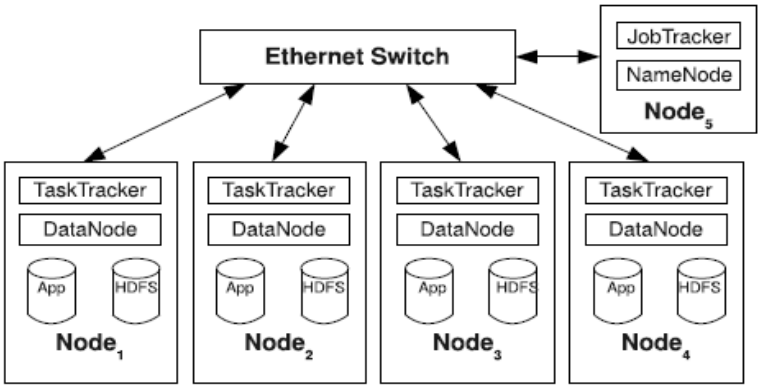

Figure: Cluster Setup in the network

2.2.2. Task Tracker

Duty of task tracker is to execute the job that is assigned by jobtracker node and then report the status of jobs back to job tracker.

In the cluster, task tracker daemon runs on every slave node. Therefore processing of data and its storage is also done by tasktracker. Map, reduce and shuffle operations are performed by task tracker and these operations are assigned by job tracker.

Tasktracker maintain a set of slots. Some slots are alloted for map tasks and some are alloted for reduce tasks. When jobtracker want to schedule task, then it first determine the available empty slot on the server that contain needed data in the datanode. And if empty slot is not available then jobtracker look for another empty slot in the same rack.

Meanwhile of processing tasktracker generate heartbeat signal in few minutes for jobtracker to assure that tasktracker is alive and performing its job. This heartbeat is also useful in determining the number of available slots. After completion of job, tasktracker report back to jobtracker with the status of the job.[7]

3. APPLICATIONS OF HADOOP

➔ Hadoop is used to analyze the risks which are life threatening for mankind

➔ It is used to identify the security breaches by analyzing the warning signs

➔ Hadoop is used to understand the perception of people about company or organization by analyzing their social media conversations.

➔ By analyzing sales data based on various factors like weather, days, weekends etc., it will help to understand when to sell which products.

➔ With the help of log files which are generated by software contains very useful data. By analyzing these log files one can find security breaches and usage statistics

➔ It is used in various fields like politics, data storage, financial services, health care, telecoms, human science, travel etc.[8],[9]

4. CONCLUSION

Big data is data which is accumulating from different sources and with different varieties like social media, sensor's data, emails etc. on tremendous speed. Today's data's volume range in petabytes but in future it will range from few exabytes to thousands of exabytes. To handle such a volume of data an efficient tool is needed which is able to analyze and mine some useful knowledge from such a large volume of data. Hadoop is the answer of these needs raised due to big data. Hadoop is applicable in all the fields of life like health, science, telecoms, data storage etc. therefore it can be able to answer the different questions raised in different fields.

5. REFERENCES

1. Revolution Analytics White Paper," *Advanced 'Big Data' Analytics with R and Hadoop"*, 2011.

2. Konstantin Shvachko etal. , "*The Hadoop Distributed File System"*, IEEE, 2010

3. Hadoop. http://hadoop.apache.org, September, 2015.

4. Jens Dittrich et al., *"Efficient Big Data Processing in Hadoop MapReduce"*, The 38th International Conference on Very Large Data Bases, Istanbul, Turkey. Proceedings of the VLDB Endowment, Vol. 5, No. 12, August 27th 31 st, 2012.

5. Harshawardhan S. Bhosale etal. , *"A Review Paper on Big Data and Hadoop"*, International Journal of Scientific and Research Publications, Volume 4, Issue 10, October 2014.

6. HDFS (Hadoop distributed filesystem) Architecture, September 2015,http://hadoop.apache.org/common/docs/current/hdfs design.html.

7. Hadoop Map Reduce framework, September, 2015,https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html

8. web_link1:http://www.mrc-productivity.com/blog/2015/06/7-real-life-use-cases-of-hadoop/

9. web_link2: http://hadoopilluminated.com/hadoop_book/Hadoop_Use_Cases.html