ISSN:2320-0790

COMPUSOFT
An International Journal of Advanced Computer Technology

A Quest on Hadoop

Suman Elizabeth Daniel[1], Binu A[2]

[1]Post Graduate Student, Dept. Of IT, Rajagiri School of Engineering & Technology, Kochi, sumanedan@gmail.com
[2]Asst. Professor, Dept. Of IT, Rajagiri School of Engineering & Technology, Kochi, binu_a@rajagiritech.ac.in

Abstract: Everyday quintillion bytes of data are created. About 90% of this data which are posts to social media sites, digital pictures, videos etc. are unstructured. These data is BigData and should be formatted to make it suitable for data mining and its subsequent analysis. Hadoop offers a firm platform in this regard and is designed to handle mixture of complex and structured data so that computationally extensive tasks can be performed. The paper also acquaints a brief idea on how we store and query BigData using Hadoop.

Keywords: Hadoop; BigData; Map/Reduce; Hadoop Distributed File Systems; HDFS; Job Tracker; Task Tracker; Big Data Analysis.

I. INTRODUCTION

The current era witnesses large amount of data flowing through the internet. These data can be in form of digital videos and pictures, data from sensors about climatic information, posts to social media sites etc. Such data termed as BigData are highly unstructured and unformatted making their processing and analysis a hectic task. A promising solution in this scenario is Hadoop. Hadoop is an Apache open-source project that provides support for data-intensive distributed applications. Hadoop comprises of Hadoop kernel, Map/Reduce paradigm which acts as its offline computing engine and HDFS which is Hadoop Distributed File System. Hadoop was actually created by Doug Cutting and Mike Cafarella in 2005 and named it after Doug's son's toy elephant. Hadoop is written in Java programming language and is regarded as an Apache top-level project. Hadoop has many related projects mainly Hive, HBase, Zookeeper etc.

II. ARCHITECTURE OF HADOOP

Hadoop comprises of Hadoop Common providing access to file systems supported by Hadoop. The Hadoop common package restrains the needed Java Archive [JAR] files and scripts to start and configure Hadoop. It also includes source code, documentation and a contribution section that includes the Hadoop community projects. [5]

A typical Hadoop cluster consists of a single master and several data-nodes as the workers. The master node includes the JobTracker, TaskTracker, Name-node and DataNode. The worker node acts as DataNode and TaskTracker. JobTracker is needed to assign MapReduce [computational paradigm in Hadoop] tasks to the worker nodes. A TaskTracker has the functionality to accept MapReduce and shuffle tasks from the JobTracker. The NameNode acts as the center core of Hadoop File system and keeps in account the directory tree of all files in file system and tracks of where the file data is kept. A secondary NameNode can be provided to generate snapshots of Name Node's memory structures preventing file-system corruption and reducing loss of data. DataNode stores data in the File System and a functional system would have multiple DataNode with data replicated within them. [5] A simple Hadoop cluster architecture is depicted below.

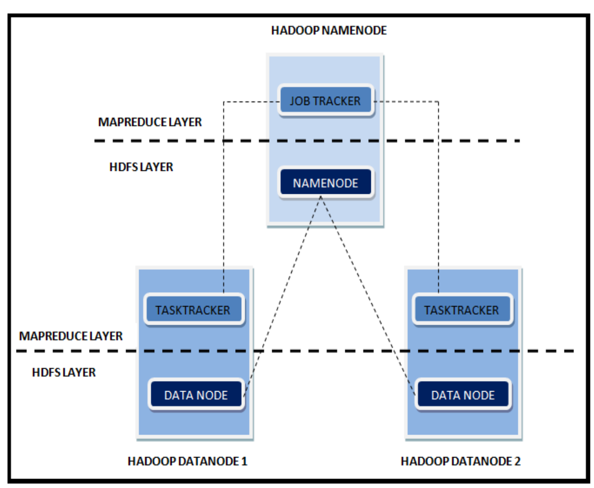

Figure I: Architecture of Hadoop

Hadoop-compatible file system should ensure location awareness for effective scheduling of work particularly the name of the rack, the network switch related to the worker node. This information is inevitable to run work on nodes, retrieve information from them and reducing backbone traffic. The Hadoop Distributed File System [HDFS] encourages replicating data and tries to keep different copies on different racks. This would largely reduce the impact of rack power outage or switch failure making the data readable even in case of such events.

A Hadoop needs java Runtime Environment [JRE] 1.6 or higher for its configuration. The Secure Shell [ssh] is inevitable for the standard Hadoop start-up and shut down scripts.

III. MAP REDUCE PARADIGM

MapReduce is the computational paradigm adopted by Hadoop which was designed for large scale cluster-based computing architectures. It was proposed by Google to handle large-scale web search applications. This approach emerged as a promising programming approach for developing machine learning, data mining and search applications in data centers. It is advantageous on the perspective that it abstracts the programmers from the scheduling problems, parallelization, partitioning, replication and focus on developing their applications.

Map and Reduce are the data processing functions adopted by Hadoop programming model as shown in Figure II. Map tasks are parallel run on input data partitioned into fixed sized blocks to produce intermediate output as a collection of <key, value> pairs which in turn are shuffled across different reduce tasks based on <key, value> pairs. Each reduce task accept only one key at a time and then process data based on the key and provides resultant output as <key, value> pairs. [2]

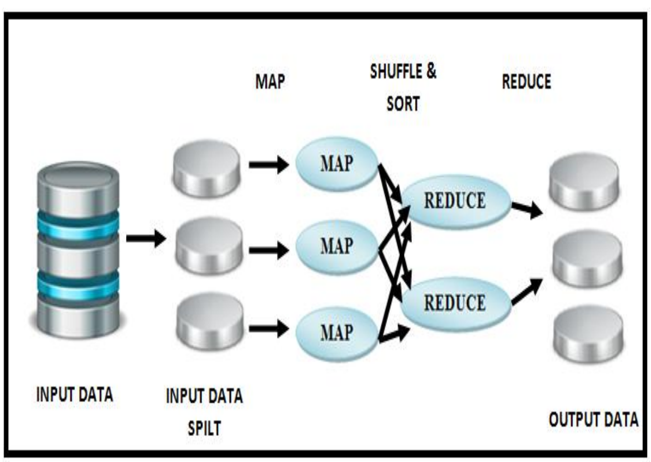

Figure II: Hadoop MapReduce

The Name-node's JobTracker accepts the job submitted from the user and split into map and reduce tasks and then assigns the tasks to Tasktrackers. They monitor the progress of Tasktrackers and when job is completed intimates it to the user. Each TaskTracker is assigned a certain number of slots which would decide how many map and reduce tasks it can perform at a specific time. The reliability and fault tolerance of MapReduce paradigm is ensured by HDFS by storing and replicating the inputs and outputs of a Hadoop job. Scheduling policies are integrated into Hadoop since jobs have to share cluster resources.

IV. HADOOP DISTRIBUTED FILE SYSTEM [HDFS]

Developed by Google, Google File System [GFS] a distributed file System was designed specifically for efficient and reliable access to data using large clusters. In this context, every file is divides to 64 MB chunks which are usually appended to or read and rarely overwritten or shrunk. [6] GFS was specifically designed and optimized to run on data centers and thereby providing extremely high data throughputs, lower latency and tolerant to server failures. The open source Hadoop Distributed File System [HDFS] was formed in inspiration from GFS and has the capability to store extremely large files across multiple machines. The reliability of the system is ensured by replication of data across multiple machines [1]. The replicas are stored in multiple compute nodes as in GFS helping in processing rapid computations. Data can be also obtained over HTTP which enables access to all content from web browser. HDFS adopts master/slave architecture. [2]

The figure III shows the Hadoop Distributed File System consisting of a single NameNode and multiple DataNodes arranged in multiple racks. The NameNode maps the data blocks to DataNodes and manages the file system operations including opening, closing and renaming files as well as directories. DataNodes does the activities such as block creation, deletion and replication of data blocks as per the instructions provided by NameNode. The NameNode also has the responsibility to maintain records of file system namespace inclusive of information like which records are created, deleted and modified by the users. A typical HDFS block size is 64 MB with replication factor 3 [second copy on local rack and third on remote rack]. [2]

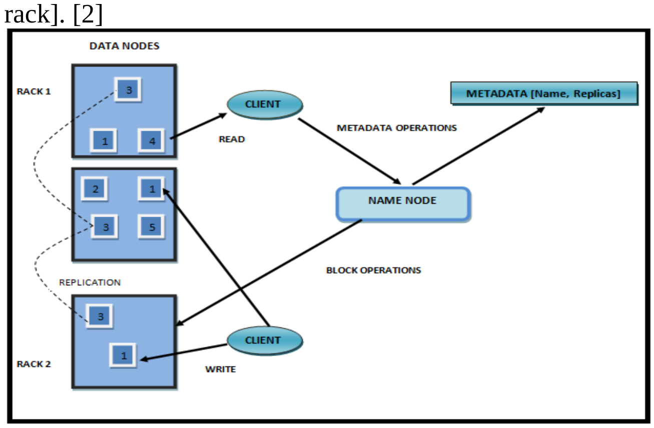

Figure III: Hadoop Distributed File System

V. Scheduling Mechanisms In Hadoop

Hadoop uses FIFO as the default scheduling mechanism. But later it was realized that the concept of job priority should be added with paved way for the inclusion of Fair scheduler and Capacity scheduler. A significant work was done by Facebook and Yahoo in this regard. The detailed description of different schedulers is discussed below:

A. Default FIFO Scheduler: The jobs are executed in order of their submission. The FIFO scheduler operates using a FIFO queue. Whenever a user submits a job, the job gets partitioned into individual tasks and then loaded into the queue. The jobs are assigned to free slots based on their availability in TaskTracker nodes. Since every job utilizes the whole cluster, each job has to wait for their turn for accomplishment of task. [2]

B. Fair Scheduler: Actually developed by Facebook the fair scheduler provides every user a fair share of cluster capacity over time. The jobs are assigned to pools and each pool assigned with a minimum number of Map and Reduce slots. Fair scheduler supports preemption and administrators can also enforce priority settings on certain pools. The shorter jobs are allocated sufficient resources to finish quickly while larger jobs are ensured to not to starve for resources. [3]

***C. Capacity Scheduler*:** Capacity scheduler was actually developed by Yahoo and is intended for the scenario where the number of users is large and fair allocation of computational resources should be ensured. Here the jobs are allocated based on the submitting user to the queues with configurable number of Map and Reduce slots i.e. the cluster capacity is shared among the users and not among the jobs. [4]

VI. Advantages Of Hadoop

Hadoop has various unique strengths which makes it advantageous. Hadoop offers a computing solution that is

A. Scalable: New nodes can be added without the need to change data formats.

B. Cost-Effectiveness: Brings out massive parallel computing and thereby offering sizable decrease in cost per terabyte of storage.

C. Flexible: It is schema-less and can any type of data structured or not from any number of sources.

D. Fault Tolerant: When a node is lost the system redirects to another location of data and continues processing.

E. Reliable: Data is replicated across multiple nodes and RAID storage is not needed. [7]

VII. Big Data Analysis Using Hadoop

This section elucidates the mechanism of how BigData is stored and queried using Hadoop from its HDFS. Big Data is the unstructured data that arises as data from social feeds, Images, data about world events, documents, web logs, GIS [Geographic Information System] data, audit logs etc. All these data are made easier to be stored and queried by using the Big Data engine powered by Hadoop.

The background Hadoop architecture consists of a master node which acts as the Name Node and Job Tracker and the data nodes which are the slaves. The immense amount of unstructured data is chopped and stored in the HDFS [Hadoop Distributed File System]. These data are replicated and scattered over numerous nodes which is the part of the Hadoop cluster which ensures fault tolerance in the system. The single Name Node stores the metadata and the different data nodes stores the data blocks. Both offers local storage and computation.

Whenever there is need for analysis of a specific data the query is submitted via an application interface from the client machine which has Hadoop installed in it. The submitted query is acquired by the master node which uses the *Map* process to assign the sub job to the slave nodes. The sub job is then executed in parallel in each of the nodes. On completion of the job the result is returned back to the master node. It then aggregates the result using the *Reduce* process which is the actual query result. This result is procured by the client.

The following figure depicts the entire mechanism of how data is stored in the HDFS and whenever there is a query from the client how the wanted data is acquired by the client.

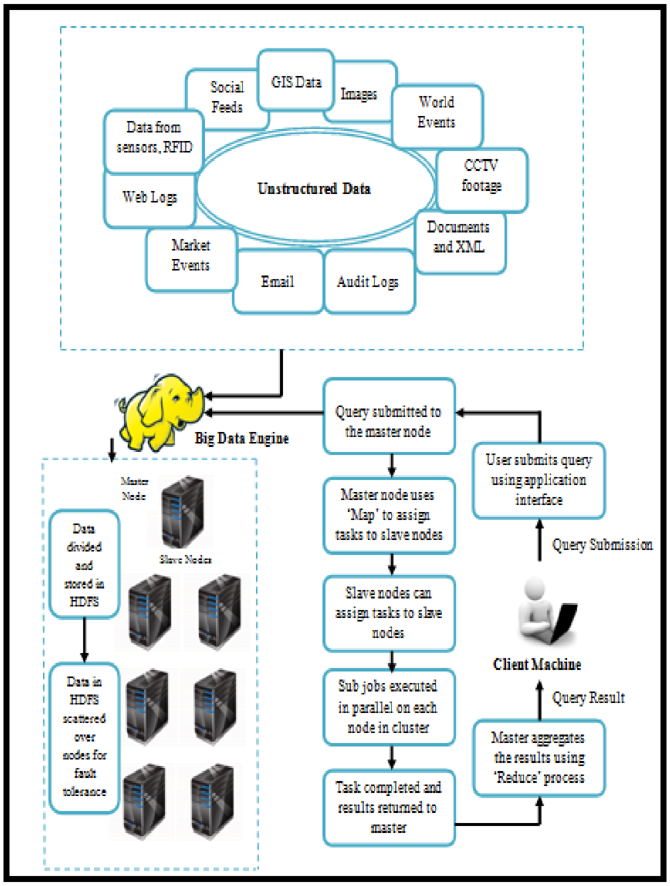

Figure IV: Storing and Querying of Big Data using Hadoop

VIII. Applications

Hadoop outperforms any other existing technique in BigData perspective which has paved way for its widespread applications. It is extensively used by Amazon to build product search indices. Yahoo has the biggest Hadoop cluster with about 20,000 computers serving the purpose of research in web search. AOL has successfully adopted Hadoop for running application that analysis behavioral pattern of their uses and offer targeted services. Facebook uses Hadoop for storing copies of their internal logs and dimension data sources and also as a source for reporting and analytics.

IX. Challenges Faced By Hadoop

Hadoop as an emerging research area has its own challenges. Some of the prominent challenges faced by Hadoop in adopted organizations are:

A. Universal Data Access: Hadoop need to process and store a variety of data from diverse data sources and combining and processing all relevant data is a challenging task.

B. Data parsing and exchange: Ability to derive meanings and make sense out of it across all data types is a major challenge.

C. Managing Metadata: Metadata management and data auditability is lacking in Hadoop due to which outcomes of projects are suspect and suffers from inconsistency and poor visibility.

D. Data quality and data governance: Organizations keep some data in Hadoop for storage or experimental tasks that does not require high quality data while some use for end-user reporting and analytics. In such a scenario it is found difficult to trust the underlying data.

E. Mixed Workload management*:* Hadoop finds difficulty to manage mixed workloads according to user Service-Level Agreements [SLA].

F. Resource optimization and reuse: Hadoop needs a framework to reuse and standardize data integration tasks.

G. Interoperability with rest of Architecture: A major challenge faced by Hadoop is rationalizing Hadoop and integrate it as part of extended environment.

X. Conclusion

Hadoop is an upcoming solution for processing and analysis of BigData. It has all the pertinent capabilities for such a scenario. But as an emerging research area it also encounters challenges which should be curbed at the latest. The organizations adopting Hadoop should implement a comprehensive, open and unified data integration platform to take the complete advantage of Hadoop.

XI. Acknowledgement

We are greatly indebted to the college management and the faculty members for providing necessary facilities and hardware along with timely guidance and suggestions for implementing this work.

XII. References

[1] Jeffrey Shafer, Scott Rixner, and Alan L. Cox IEEE ISPASS, 2010. The Hadoop Distributed File System: Balancing portability and Performance.

[2] B.Thirumala Rao, Dr. L.S.S Reddy. International Journal of Computer Applications - ISSN 0975-8887, Volume 2, No. 9, November 2011. A Survey on improved Scheduling in Hadoop MapReduce in cloud Environments

[3] Hadoop's Fair Scheduler - http://hadoop.apache.org/common/docs/ r0.20.2/fair_scheduler.html

[4] Hadoop's Capacity Scheduler: http://hadoop.apache.org/core/docs/ current/capacity_scheduler.html.

[5] Apache Hadoop. http://hadoop.apache.org.

[6] Hadoop Distributed File System, http://hadoop.apache.org/hdfs

Hadoop in Practice – Ebook on Hadoop