Volume 5, Issue 6, June 2015 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

A Study of Hadoop: Structure and Performance Issues

[1]Vibha Sarjolta, [2]Dr. A.J Singh
[1]Research Scholar, [2]Professor
[1, 2] Department of Computer Science
Himachal Pradesh University, Shimla, India

Abstract-Big Data is the latest buzzword that is being used to describe huge quantities of both structured and unstructured data. In order to process such huge quantities of data a software is required that does this efficiently, and this is where Hadoop steps in. Hadoop has become one of the most used frameworks when dealing with Big Data. It is used to analyze and process Big Data so it becomes important that it performs its best and the results so obtained are correct and not ambiguous. In this paper the structure of Hadoop is studied and how its different components contribute to its performance. Some performance issues affecting Hadoop are also studied in this paper.

Keywords- Big Data, Hadoop, MapReduce, HDFS (Hadoop Distributed File System), Performance

I. INTRODUCTION

With the growing digital age a large amount of data is being produced everyday, which can be in terms of the data that we find on social networking sites, online shopping stores, weather forecasts, institution records and companies database. It is easy to see that individuals are increasingly generating vast amounts of digital content from photos and videos, to tweets and documents. Besides this, the amount of data generated by machines is even greater than that generated by people. Machine logs, Radio-frequency identification readers, sensor networks, vehicle GPS traces, retail transactions—all of these contribute to the growing mountain of data [1].

According to a rough estimate of IDC (international data corporation) and Century Link in 2012, the planet had 2.75 zettabytes (ZB) worth of data. This number was likely to reach 8ZB by 2015 and forecasted to grow to 44 zettabytes by 2020. This situation has given rise to the new buzzword known as “big data” [2]. Big Data is a term that is used for any collection of data sets so large and complex that it becomes difficult to process those using traditional data processing applications. Although Big Data doesn't refer to any specific quantity, the term is often used when speaking about petabytes and exabytes of data. It is a term that is used to define huge amounts of data which can be structured, semi-structured and unstructured. Big Data has the potential to help companies improve operations and make faster, more intelligent decisions. A wide number of definitions have been put forth by various scientists, organizations to define what big data is. Big Data usually includes data sets with sizes beyond the ability of commonly used software tools to capture, curate, manage, and process data within a tolerable elapsed time. Big Data is a set of techniques and technologies that require new forms of integration to uncover large hidden values from large datasets that are diverse, complex, and of a massive scale [3].

Big Data makes it possible to achieve research results that cover a wide range of issues, and can tell us a great deal about developments in the world in many different areas [4]. In 2012, Gartner defined big data as: “Big Data is high volume, high velocity, and/or high variety information assets that require new forms of processing to enable enhanced decision making, insight discovery and process optimization." Additionally, a new V "Veracity" is added by some organizations as IBM to describe it.

The three V’s of Big Data:

- **Volume:** It refers to the huge amount of data that is present in the world today. Many factors contribute to the increase in data volume. Transaction-based data stored in companies through the years, unstructured data streaming in from social media and increasing amounts of sensor and machine-to-machine data being collected contribute to making data big in volume.
- **Velocity:** It refers to the speed at which the data is being generated. Data in today’s world is streaming in at unprecedented speed and must be dealt with in a timely manner. Reacting quickly enough to deal with data velocity is a challenge for most organizations.
- **Variety:** The data that is present today comes in all types of formats as structured, numeric data in traditional databases. Information created from line-of-business applications. Unstructured data as text documents, email, videos, audios, stock ticker data and financial transactions. Knowing which type of data is being dealt with is of great importance to the people who deal with Big Data in order to use the data in an effective manner, to bring out the desired results.

Veracity is another term used to define Big Data. It refers to the abnormalities in data. If the data that is being used by analysts is inaccurate, then the results obtained by those analyses cannot be trusted. So it is of utmost importance that the problem of veracity is considered when dealing with Big Data and dealt with.

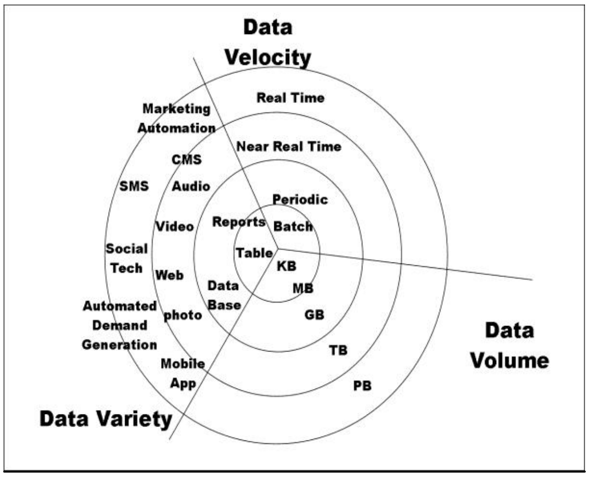

Fig 1.The three V's of Big Data [5]

Keeping in mind all the aforesaid characteristics of Big Data it becomes a huge challenge for analysts to deal with it and this has lead to the development of a number of open source platforms that have grown up specifically to handle these vast amounts of data quickly and efficiently, including Hadoop, MongoDB, Cassandra, NoSQL and more. Among all the softwares Hadoop is one such framework that is most commonly used to deal with Big Data.

II. HADOOP

The Apache Hadoop project [6] develops open-source software for reliable, scalable, distributed computing. The Apache Hadoop software library is a framework that uses simple programming models for the distributed processing of large data sets across clusters of computers. It is designed to scale up from single servers to thousands of machines, each offering local computation and storage. Rather than rely on hardware to deliver high-availability, the library itself is designed to detect and handle failures at the application layer, so delivering a highly-available service on top of a cluster of computers, each of which may be prone to failures. Apache Hadoop is a free, Java-based programming framework that supports the processing of large data sets in a distributed computing environment. Hadoop allows running applications on systems with thousands of nodes with thousands of terabytes of data.

Structure of Hadoop

Although Hadoop is best known for MapReduce and its distributed file system (HDFS, originally named NDFS), the term is also used for a family of related projects that fall under the umbrella of infrastructure for distributed computing and large-scale data processing. Some of these are Apache Hive, Pig, Avro, Zookeeper, Sqoop, Oozie and more [1].

Hadoop is specially designed for two core concepts: HDFS and MapReduce. Both are related to distributed computation. Hadoop architecture is primarily a distributed master slave architecture that consists of a single master and many slaves. The architecture consists of the Hadoop Distributed File System (HDFS) for storage and MapReduce for computational capabilities. The functions of Hadoop in the architecture are data partitioning and parallel computation of large datasets. Its storage and computational capabilities scale with the addition of hosts to a Hadoop cluster, and can reach volume sizes in the petabytes on clusters with thousands of hosts [7].

- **HDFS:** HDFS is a distributed, scalable, and portable file-system written in Java for the Hadoop framework. It is a UNIX-based data storage layer of Hadoop. HDFS stores very large files across multiple machines. HDFS is derived from the concepts of Google file system. A HDFS cluster primarily consists of a NameNode that manages the file system metadata and DataNodes that store the actual data. An important characteristic of Hadoop is the partitioning of data and computation across many (thousands of) hosts, and the execution of application computations in parallel, close to their data [5]. On HDFS, data files are replicated as sequences of blocks in the cluster. When adding commodity servers a Hadoop cluster scales computation capacity, storage capacity, and I/O bandwidth. HDFS can be accessed from applications in many different ways.

  Features of HDFS [8]
- Hadoop, including HDFS, is well suited for distributed storage and distributed processing using commodity hardware.
- It is fault tolerant, scalable, and extremely simple to expand.
- HDFS is highly configurable with a default configuration well suited for many installations. Most of the time, configuration needs to be tuned only for very large clusters.
- Hadoop is written in Java and is supported on all major platforms.
- Hadoop supports shell-like commands to interact with HDFS directly.
- The NameNode and DataNodes have built in web servers that make it easy to check current status of the cluster.
- **MapReduce:** MapReduce is a programming model that is used for processing large datasets distributed on a large cluster. MapReduce is the heart of Hadoop. Its programming paradigm allows performing massive data processing across thousands of servers configured with Hadoop clusters. This is derived from Google MapReduce [5].

Hadoop MapReduce is a software framework for writing applications easily, which process large amounts of data (multi Terabyte datasets) in parallel on large clusters consisting of thousands of nodes of commodity hardware in a reliable, fault-tolerant manner. The MapReduce paradigm is divided into two phases, Map and Reduce that mainly deal with key and value pairs of data. The Map and Reduce task run sequentially in a cluster; the output of the Map phase becomes the input for the Reduce phase. These phases are explained as below:

- **Map phase**: The datasets are first divided and then assigned to the task tracker to perform the Map phase. The data functional operation will be performed over the data, emitting the mapped key and value pairs as the output of the Map phase.
- **Reduce phase**: The master node then collects the answers to all the sub problems and combines them in some way to form the output which is the answer to the problem it was originally trying to solve.

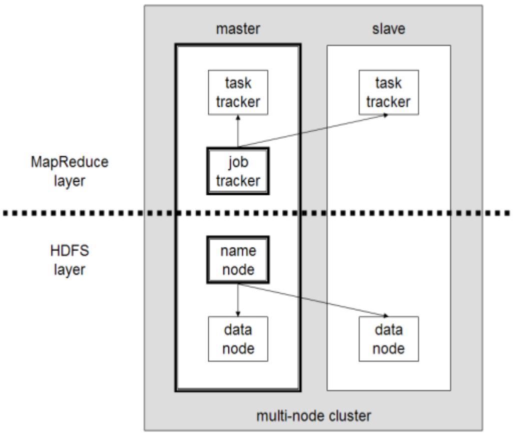

Fig 3.Hadoop Architecture [9]

The architecture of Hadoop consists of a Hadoop Distributed File System (HDFS) layer, that is used for the storage of data and a data processing and execution model called MapReduce, both of which are necessary for storing and processing large volumes of data.

Both HDFS and MapReduce are based on master slave architecture. A small Hadoop cluster includes a single master and multiple worker nodes. The master node consists of a JobTracker, TaskTracker, NameNode and DataNode. A slave or worker node acts as both a DataNode and TaskTracker, though it is possible to have data-only worker nodes and compute-only worker nodes. The Hadoop data processing includes several tasks that help achieve the final output from an input dataset. These tasks are as follows [5]:

1. Preloading data in HDFS.
2. Running MapReduce by calling Driver.
3. Reading of input data by the Mappers, which results in the splitting of the data execution of the Mapper custom logic and the generation of intermediate key-value pairs
4. Executing Combiner and the shuffle phase to optimize the overall Hadoop MapReduce process.
5. Sorting and providing of intermediate key-value pairs to the Reduce phase. The Reduce phase is then executed. Reducers take these partitioned key value pairs and aggregate them based on Reducer logic.
6. The final output data is stored at HDFS.

III. FACTORS AFFECTING PERFORMANCE OF HADOOP

When an analysis is being conducted on Big Data it is of utmost importance that the data being dealt with is accurate and does not have any abnormalities. There are numerous factors that affect the performance of Hadoop such as hardware and software when handling huge amounts of data. Both the main components of Hadoop, that is, HDFS and MapReduce play a major role in its performance Hadoop and the results that are generated. Below are some of the factors that affect the performance of Hadoop cluster:

1. **Hardware:** When selecting hardware, care should be taken to configure it keeping in mind the nodes to be stored. Using fast machines, having processing speed as a function of the number of cores available helps in improving performance.RAM requirements for the NameNode increase in proportion to the total number of data blocks in the cluster, and extra RAM on the NameNode accommodates the future growth of the cluster. Disk speed affects the degree of throughput that can be achieved, and the number of disks per node affects the cluster’s ability to “scale up,” which in this case means the ability to add storage to individual nodes in the system [10].
2. **MapReduce:** Tuning the number of map tasks and reduce tasks for a particular job in the workload is another way that performance can be optimized. If the mappers are running only for a few seconds then fewer mappers can be used for longer periods [1].Also performance depends on the number of reducers used which should be slightly less than the number of reduce slots in the cluster to improve performance. This allows the reducers to finish in one wave and fully utilizes the cluster during the reduce phase. MapReduce job performance can also be affected by the number of nodes in the Hadoop cluster and the available resources of all the nodes to run map and reduce tasks.
3. **HDFS:** The number of reading and writing operations performed on the nodes also affects the performance of Hadoop. The performance of HDFS also depends on whether the work is being performed on big or small dataset.

4. **Shuffle tweaks:** The MapReduce shuffle also helps to alter performance as it maintains a balance between the map and reduce functions. If adequate amount of memory is allocated to map and reduce functions then the shuffle can also be allocated enough memory to operate thereby improving performance. Therefore, a trade off needs to be carried out when allocating memory to tasks in MapReduce.

IV. PERFORMANCE ISSUES

Hadoop offers distributed storage, superior scalability and has proved to be one of the most useful frameworks for handling Big Data. But there are certain issues present that hamper its performance. Some of the performance issues observed in Hadoop are as follows:

1. **Hadoop on small data:** Although big data is not exclusively made for big businesses, not all big data platforms are suited for small data needs. Unfortunately, Hadoop happens to be one of them. Due to the high capacity design of Hadoop Distributed File System or HDFS it lacks the ability to efficiently support the random reading of small files. As a result, it does not perform well with small quantities of data.
2. **MapReduce inefficiencies:** MapReduce programs are not guaranteed to be fast. In particular the partition function and the amount of data written by the Map function can have a large impact on the performance. Additional modules such as the Combiner function can help to reduce the amount of data written to disk, and transmitted over the network. The sort-merge algorithm used in MapReduce programming model is not always the most efficient algorithm for performing certain kinds of analytical tasks, especially for those which do not care about the order of intermediate keys [11].
3. **Open Source nature of Hadoop:** Being an open source platform makes Hadoop vulnerable to stability issues due to the contributions of the many developers who continue to work on the project. These issues may affect performance and to tackle such issues latest stable versions should be used. Also this problem can be solved when they are run under a third-party vendor equipped to handle such problems.
4. **Multiple copies of already big data:** Because HDFS was built without the notion of efficiency; it results in multiple copies of the data. At a minimum, there are generally three copies of the data. And because of the need for data locality in maintaining performance, we very often see six copies of the data required and that's for data that's already "big" by definition [12].
5. **Expensive crash recovery:** For processes that complete fast, and where the data fits into main memory of a single machine or a small cluster, using a MapReduce framework usually is not effective: since these frameworks are designed to recover from the loss of whole nodes during the computation, they write interim results to distributed storage. This crash recovery is expensive, and only pays off when the computation involves many computers and a long runtime of the computation - a task that completes in seconds can just be restarted in the case of an error; and the likelihood of at least one machine failing grows quickly with the cluster size. On such problems, implementations keeping all data in memory and simply restarting a computation on node failures, or - when the data is small enough - non-distributed solutions will often be faster than a MapReduce system [13].
6. **Uncertainty:** Hadoop can usually ensure that a data job completes, but it is unable to guarantee when the job will be completed. Hadoop jobs often take longer to run than anticipated, making it risky to depend on the job output in production applications. When a critical production job is running, other, lower-priority jobs can sometimes swallow up the cluster's hardware resources, like disk and network, creating serious resource contentions that ultimately can result in critical production jobs failing to complete safely and on time.

V. CONCLUSION AND FUTURE SCOPE

Hadoop being open-source software is easily available and one of the most used framework for handling Big Data. When dealing with data using Hadoop certain things need to be studied carefully so that the results obtained are accurate and easy to analyze. In this paper the structure and some of the performance issues of Hadoop are studied. Hadoop consists of HDFS and MapReduce, its two main components, which play a major part in its performance. Limitations of these two components cause performance degradation of Hadoop as a whole. Through this paper it is found that although Hadoop is the main software used for analyzing huge amounts of data it faces certain problems when handling small amounts of data.

Through this paper an attempt is made to bring out the issues related to Hadoop's performance in a theoretical manner. Hadoop is evolving so there is scope for future work, in which its performance can be enhanced by altering variables as system latency, memory settings, Input/output bandwidth, job parallelization that will result in improved level of performance.

ACKNOWLEDGEMENT

I would like to thank the almighty, for bestowing his grace upon me and also extend my heart filled gratitude to my supervisor, Dr. A.J Singh, Professor, Computer Science Department, Himachal Pradesh University for his continuous guidance and encouragement. I would also like to thank our university for providing me with this opportunity to carry out this research work.

REFERENCES

[1] Tom White O Reilly, yahoo press, "Hadoop the definitive guide", 3rd edition, 2012.

[2] Harish Chauhan and Jerald Murphy, "Harnessing Hadoop: Understanding the Big Data Processing Options for Optimizing Analytical Workloads", Cognizant 20-20 Insights, Nov 2013.

[3] Ibrahim A.T. Hashem, Ibrar Yaqoob, Nor B. Anuar, Salimah Mokhtar, Abdullah Gani, Samee Ullah Khan, "The rise of "big data" on cloud computing: Review and open research issues", Information Systems, Volume 47, January 2015, Pages 98-115,ISSN 0306-4379.

[4] Big Data, for better or worse, http://www.sciencedaily.com/releases/2013/05/130522085217.htm, Accessed on 3 Jun, 2015.

[5] Vignesh Prajapati, "Big Data Analytics with R and Hadoop", Packt Publishing, 2013.

[6] Apache Hadoop, http://hadoop.apache.org/, Accessed on 3 Jun, 2015.

[7] Alex Holmes, "Hadoop in practice", Manning Publications, Second edition, 2014.

[8] HDFS user Guide, https://hadoop.apache.org/docs/stable/hadoop-project-dist/hadoop-hdfs/HdfsUserGuide.html, Accessed on 3 Jun, 2015.

[9] Apache Hadoop Wikipedia, http://en.wikipedia.org/wiki/Apache_Hadoop, Accessed on 3 Jun, 2015.

[10] Dirk deRoos, Paul Zikopoulos, Bruce Brown, Rafael Coss, Roman B. Melnyk, "Hadoop for Dummies" ,2014.

[11] Dawei Jiang,Beng Chin Ooi,Lei Shi,Sai Wu,"The Performance of MapReduce: An In-depth Study", Proceedings of the VLDB Endowment, Vol. 3, No. 1,2010.

[12] Paraccel, "Hadoop's Limitations for Big Data Analytics", white paper, 2012.

[13] Ullman, J.D , "Designing good MapReduce algorithms", XRDS: Crossroads, the ACM Magazine for Students (Association for Computing Machinery) 19:30.doi:10.1145/2331042.2331053,2012.