ISSN 2277-2685
IJESR/April. 2017/ Vol-7/Issue-2/1-6
Keerthigan M[3], Dr Kalaikumarano T[4] , Dr Karthika S[5] *et. al.,* / **International Journal of Engineering & Science Research**

Analysis of Hadoop's MapReduce Algorithm

Keerthigan M[3], Dr Kalaikumarano T[4] , Dr Karthika S[5]
[*1,2,3] Student, Dept of Computer Science and Engineering, SNS College of Technology, INDIA
[4] Professor and Head, Dept.of Computer Science and Engineering, SNS College of Technology, INDIA
[5] Professor and Dean, Dept.of Computer Science and Engineering, SNS College of Technology, INDIA
, Blocks, racks, hdfs, datanode, namenode.

ABSTRACT
In any engineering field the data associated with knowledge is important one for taking decisions for solving problems in the current system development. In current scenario Organization are supposed to work with huge amount of data. Based on those data analysis, predictions, manipulations are made. Replicas of blocks are created to improve the Redundancy in distributed system. A deep understanding is needed inorder to remove some drawbacks in these fields.

I. INTRODUCTION

More than just a petabyte of data, "big data" is an unstructured collection of information. Data in the form of logs is an example of unstructured data. Big Data may only consist of digital information; no physical rows, columns, or other data structures are allowed. The greater the data set, the more difficult it is to analyze it. In order to do business analysis on their data, certain instruments are used. Unstructured large data should primarily be used for making predictions, conducting analyses, fulfilling requirements, etc. Volume, velocity, and variety are the three characteristics that make up big data. The massive data set will be handled by the high-powered computer. However, there are limits to how much data can be processed before the computer slows down.

All types of businesses may now easily collect, store, and analyze massive amounts of data. The ultimate purpose of such data collection is to get useful insights from the data to enhance the relevant organization's business operations via the use of statistical and data mining technologies. Market basket analysis, fraud detection, consumer profiling, medicine, agriculture, and many more fields have all consecutively updated their enormous data mining demands. Companies and government agencies have been utilizing statistical approaches to analyze data for the benefit of customers and society long before data mining became popular. Data mining is a modern method of statistical analysis that builds on existing methods. Finding previously unknown links and patterns in the data is a primary goal of data mining. When data is mined for correlational patterns, some of these findings may be made. When consumers are given access to the data, they are more likely to make these kinds of unexpected findings. The entire potential of data is being held back, however, by the rising impediments that privacy and confidentiality concerns provide. Data stored for the sole purpose of analysis is often kept in a restricted area, with access limited to a select group of experts. This obviously reduces the data's value and goes against the very point of collecting them. In this respect, numerical data are crucial. They provide the greatest risk but also the highest rewards. They are the most dangerous since they

likely to be virtually entirely unique, and an attacker armed with numerical data may rapidly jeopardize the privacy and security of critical

material. They are the most useful since so much of what we learn about running a company can be reduced to numbers. This is why

II. BIG DATA

2.1. More than just a petabyte of data, "big data" is an unstructured collection of information. Data in the form of logs is an example of unstructured data. Big Data may only consist of digital information; no physical rows, columns, or other data structures are allowed. The greater the data set, the more difficult it is to analyze it. In order to do business analysis on their data, certain instruments are used. Unstructured large data should primarily be used for making predictions, conducting analyses, fulfilling requirements, etc. Volume, velocity, and variety are the three characteristics that make up big data. The massive data set will be handled by the high-powered computer. However, there are limits to how much data can be processed before the computer slows down.

2.2. ANALYSIS ON BIGDATA

Analyzing massive amounts of data, which may include both static and dynamic streams of information, is known as "big data analytics." Unstructured data from sensors, devices, third parties, Web applications, and social media, much of it sourced in real time on a large scale, is at the heart of big data analytics. These details of business operations and customer interactions rarely make it into a data warehouse or standard report. Companies may analyze big data using sophisticated analytics methods like predictive analytics, data mining, statistics, and natural language processing to learn about their present situation and forecast how customers' behaviors will change in the future. Hadoop and MapReduce are two examples of new approaches to dealing with large data that may be used as an alternative to conventional data warehousing.

Big Data analytics, illuminating hidden patterns for improving all points of contact with customers. Analytics are swiftly produced, adapted, and shared across business teams using personal workspaces and self-service templates.

2.3. HADOOP

- Hadoop is an ecosystem of programs. Hadoop's primary goal is to facilitate the operation of applications on large data sets. Hadoop is a free software project that is licensed by the Apache Software Foundation. It has to be able to efficiently process a large amount of data (Volume), data that is arriving quickly (Velocity), and data that is diverse in kind. Hadoop partitions the massive data collection. As an added bonus, it dissects the calculation into manageable chunks. When all of the calculations have been completed, the results are added together. There are two primary parts of Hadoop, and they are

- MapReduce
- File system

The name of the storage system there is HDFS. The goal of the many tools and projects that make up Hadoop is to carry out the work at hand. Hadoop relies on a decentralized system to function. Hadoop is a LINUX-based suite of tools, which is why it can run on so many inexpensive machines. And eventually we'll refer to all these machines as "slaves."

III. HDFS

3.1. HDFS is a distributed file system that runs on commodity hardware clusters and is optimized for storing and serving extremely big files with streaming data access patterns. Conditions such as those with • Low latency data access • Many tiny files,

3.2. • Multiple authors making unannounced changes to the same file

3.3. Hadoop is only used for reading out massive data sets at one go, hence the data retrieved cannot be stored.

3.4. HDFS COMPONENTS:

- Namenode
- Datanode

Namenode

The Namenode is the masternode where all job tracking procedures take place. A Namenode is a trusted computer that monitors and manages the blocks that exist on Datanodes. Nothing is stored in its memory. The Namenode is a costly node with several copies of every machine.

Datanodes

3.5. Datanodes serve as Hadoop's indentured servants. Each machine offers octal storage, and they have been deployed. It is your job to fulfill the client's read/write requests. Data node connected to the task tracker, where procedures for tracking tasks are performed. To let you know they're still there, Task Tracker will sometimes "heartbeat" you.

3.6. Blocks and Racks are where much of our attention is directed in the current HDFS Architecture. DataNodes are shelved in racks for convenient access. Multiple racks are used so that replication may be used. DataNode receives the duplicate information each time an operation is executed because of replication. Therefore, some storage is required to keep track of information that has previously been retrieved.

3.7. RACK AWARENESS:

The Namenode chooses which DataNode will receive the first data block. DataNode clients will attempt to locate themselves there if possible. If it is already on the same rack, it will select a random position. There are two other nodes in the cluster where data is replicated to automatically. Each of the three Datanodes in the pipeline is connected to one another through a pipeline. The second copy of the block, or DataNode, is placed on a rack at random from among the other nodes. The purpose of this is to add safety nets. In the same rack as the second copy, the third replica goes in the randomly selected node. From the second DataNode, the information is sent to the third through a pipe.

The third DataNode sends a packet of acknowledgment to the second DataNode, which in turn sends a packet to the first DataNode to verify that the write was successful. And then the client, all the way from the first dataNode! In this way, the client is given confirmation that the blocks were received without incident. In this situation, the procedure is repeated for each of the file's blocks.

To divide into two blocks, second and third. It's worth noting that there are duplicates of every block on at least two shelves. Now that it has finished composing the dataNode pipeline and obtained confirmations, the client notifies the NameNode. Before replying, the NameNode will make sure there is at least some replication of the blocks.

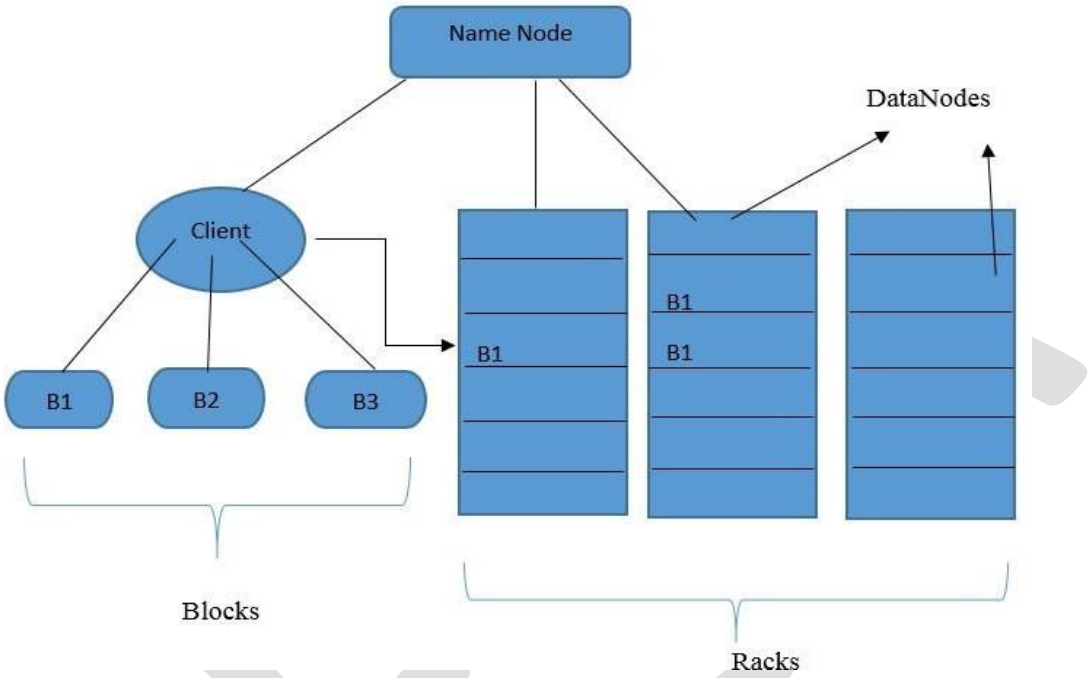

Fig. 1 Existing Replica allocation on Racks

In this approach, if the primary DataNode fails for whatever reason, only the backup in the second rack will be usable. This also lengthens the total time spent fetching. In order to get the DataNode, the client must go to different racks. Two identical items could be found in the shelf unit. Any equivalent will do for the customer. Once this is complete, the data and results of the operation may be transferred to the client from the secondary rack. For optimal performance and data transmission, more bandwidth is required. This is a potential downside of the HDFS design.

IV. MAPREDUCE ALGORITHM

4.1. MAPREDUCE:

4.2. Core to Hadoop is the MapReduce framework. It's just a paradigm for creating programs, and it may be written in whatever language you choose. Languages like Java, Python, etc., are widely used in practice. The data is processed locally on the DataNode through the map. The Map creates a key-value pair that may then be used in the data processing.

4.3. This particular key-value combination expedites the process and produces the desired results. Input splits are used to break the file into manageable chunks. Different DataNodes must be assigned to each input split. Each input divides need a Map to accomplish. As a corollary, the number of maps equals the number of forks. Data processing tasks are carried out by map, while the amount of map programs is minimized by reduction. The efficiency will grow as a result of the cutting. Although the number of input divides is equal to the number of Maps, the size reduction is not. The correct reduction is achieved by the use of predetermined algorithms. While there are many variations on the MapReduce framework, the Google MapReduce Algorithm has gained the greatest traction. Other common MapReduce algorithms include minimal MapReduce, sorting,

searching, BFS, and TF-IDF.

4.4.

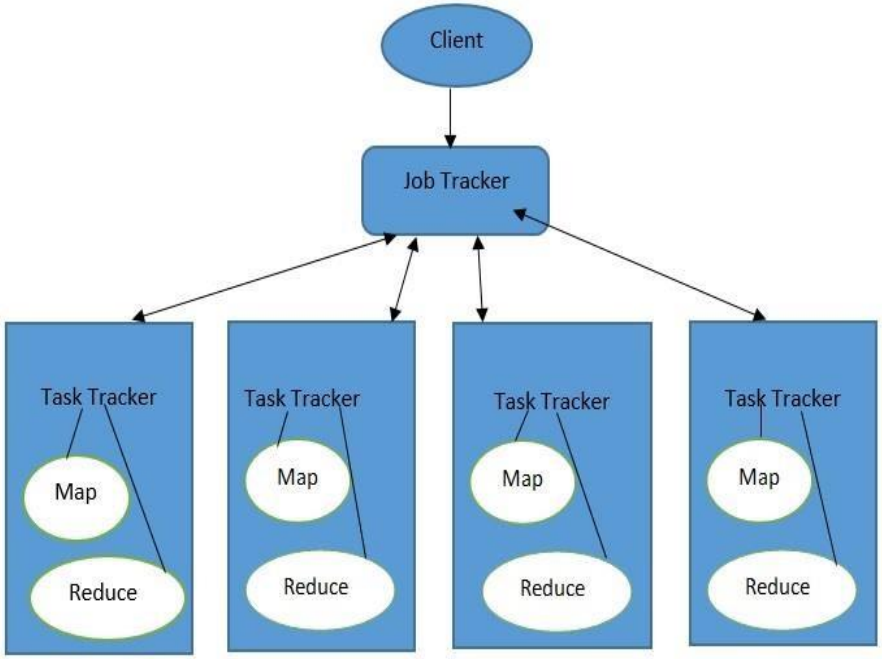

Fig. 2 Map Reduce inside the Task Tracker

4.5. ALGORITHMS FOR MAPREDUCE

- Sorting
- Searching
- TF-IDF
- BFS
- PageRank
- More advanced algorithms

4.6. MAPREDUCE ALGORITHM BY GOOGLE:

MapReduce Jobs Tend to be very short, code-wise Identity Reducer is very common "Utility" jobs can be composed Represent a data flow, more so than a procedure

Sort Algorithm Takes advantage of reducer properties: (key, value) pairs are processed in order by key; reducers are themselves ordered. Mapper: Identity function for value

(k, v) _ (v, _)

_ Reducer:

Identity function (k', _) -> (k', "")

Sort: The Trick
_ (key, value) pairs from mappers are sent to a particular reducer based on hash(key)
_ Must pick the hash function for your data such that k1 < K2 => hash (k1) < hash (k2)

V. CONCULSION

VI. Basic procedures on how a system would interact with other parts were also explored in depth, making this a comprehensive look at the fundamentals of big data. By analyzing the material efficiently, it became clear that the map reduce method in Hadoop was functioning as intended. To solve the problem of numerical secrecy in Big Data, any methods in the future may be provided by which a Well-equipped Block racking could be done, and some technologies could be deployed.

VII. REFERENCES

Perturbing non-normal confidential variables: The copula technique. [1] Sarathy R., K. Muralidhar, and R. Parsa. The latest issue of Management Science has pp. 1613–1627.
According to [2] "A Theoretical Basis for Perturbation Methods," by K. Muralidhar and R. Sarathy (Statistics and Computing, volume 13, pages 329–335).

[1] K. Muralidhar and R. Sarathy, "Data shuffling - A new masking approach for numerical data,"

Management Science, vol. 52, pp. 658-670, 2006.

[2] L. T. Willenborg and T. D. Waal, Elements of statistical disclosure control. New York: Springer, 2001.

[3] R. Nelsen, "An introduction to Copulas," New York: Springer, 2007

[4] K. Muralidhar, R. Parsa, and R. Sarathy, "A general additive data perturbation method for database security," Management Science, vol. 45, pp. 1399-1415, 1999.

[5] K. Muralidhar, R. Sarathy, and R. Parsa., "An improved security requirement for data perturbation with implications for e-commerce," Decision Sciences, vol. 32, pp. 683-698, 2001.

[8]. Hadoop, Applications powered by Hadoop: http://wiki.apache.org/hadoop/ PoweredB

[9]. Presentation by Randal E. Bryant, Presented in conjunction with the 2007 Federated Computing Research Conference, http://www.cs.cmu.edu/~bryant/ presentations/DISC-FCRC07.ppt.

[10]. L. Barroso, J. Dean, and U. Holzle, Web search for a planet: The Google cluster architecture, IEEE Micro, 23(2), 2003, pp. 22_28.

[11].MapReduce in Wikipedia, http://en.wikipedia.org/wiki/MapReduce (accessed September2009).

[12].Hadoop in Wikipedia, http://en.wikipedia.org/wiki/Hadoop (accessed September 2009).