RESEARCHERID
THOMSON REUTERS

[Annapoorani* *et al.*, 7(2): February, 2018]
IC™ Value: 3.00

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

IJESRT
INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

IMPROVING PERFORMANCE OF DATA IN HADOOP CLUSTERS USING DYNAMIC DATA REPLICA PLACEMENT: A SURVEY

S. Annapoorani [*1], Dr. B. Srinivasan [2]
[*] Computer Science, Gobi Arts & Science College, India

DOI: 10.5281/zenodo.1165754

ABSTRACT

Big data refers to various forms of large information sets that require special computational platforms in order to be analyzed. Research on big data emerged in the 1970s but has seen an explosion of publications since 2008. The Apache Hadoop software library based framework gives permissions to distribute huge amount of datasets processing across clusters of computers using easy programmer models. In this paper, we discuss the architecture of Hadoop, survey paper of various data replication placement strategies and propose an approach for the improvement of data replica placement and suggest an implementation of proposed algorithm with various MapReduce applications for improving performance of data in Hadoop clusters with respect to execution time and number of nodes in Hadoop platform

KEYWORDS: Apache Hadoop, HDFS, MapReduce, Data Replication Placement, MapReduce applications

I. INTRODUCTION

Hadoop, well known as Apache Hadoop, is an open-source software platform for process large amount of data. It is scalable and distributed computing of large volumes of data. It provides rapid, high performance and cost-effective analysis of structured and unstructured data generated on digital platforms and within the enterprise. It is used in almost all departments and sectors today. Hadoop is a distributed file system, which lets to store and handle massive amount of data on a cloud machines, handling data redundancy. The primary benefit is that since data is stored in several nodes, it is better to process it in distributed manner. Each node can process the data stored on it instead of spending time in moving it over the network. The performance of Hadoop depends on various factors, such as amount and frequency of CPU cores, RAM capacity, throughput of storages, data flows intensity, Network bandwidth etc [1]. Hadoop is a popular cloud computing platform based on HDFS and MapReduce.

Hadoop Architecture:

In Figure 1,

- **Hadoop Common** – This includes Java libraries and utilities which provide those Java files which are essential to start Hadoop.
- **Task Tracker** – It is a node which is used to accept the tasks such as shuffle and MapReduce form job tracker.
- **Job Tracker** – It is a service provider which runs MapReduce jobs on cluster.
- **NameNode** – It is a node where Hadoop stores all file location information (data stored location) in Hadoop distributed file system. Files and directories are represented on the NameNode by inodes [2].
- **DataNode** – The data is stored in the Hadoop distributed file system. Each block replica on a DataNode is represented by two files in the local system. The NameNode does not directly call DataNodes. It uses replies to heartbeats to send instructions to datanodes

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

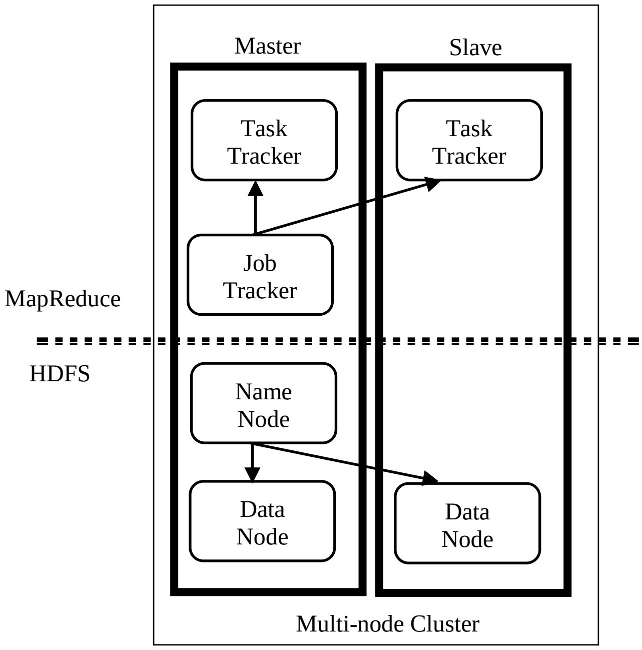

Figure 1. Architecture of Hadoop

Hadoop Distributed File System (HDFS):

HDFS is a distributed, scalable and portable file system written in Java for the Hadoop framework [2]. Each node in a Hadoop has one single datanode. A HDFS cluster consists of cluster of datanodes. HDFS is designed for storing large number of files over multiple machines. It achieves reliability by replicating the data across multiple hosts [2]. HDFS system stores files redundantly across cluster nodes for security and availability. To store a file HDFS splits it into blocks and replicates those according to a replication factor [3]. HDFS provides high throughput, and is suitable for large data sets (Figure 2)

In HDFS, the block placement plays a major role of improving performance of data. When a new block is created, the first block of replica placed in the first location allotted for the block. The other block of replica placed randomly on different nodes by using rack. For a HDFS read, name node provides the data nodes that are closer to the client. This would helps to improving write performance.

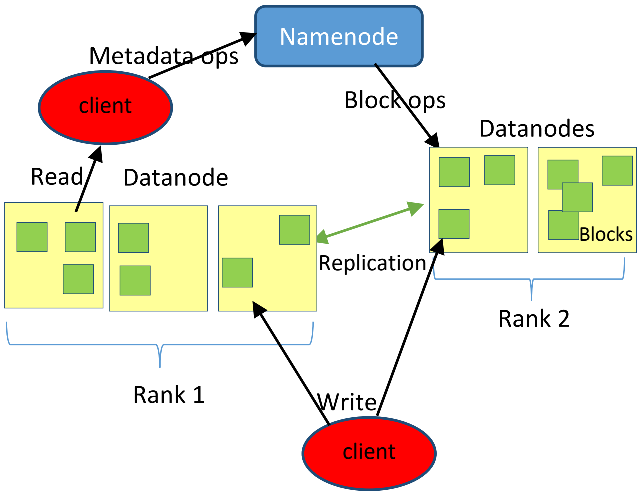

Figure 2. HDFS Architecture

ISSN: 2277-9655
Impact Factor: 5.164
CODEN: IJESS7

MapReduce:
Hadoop runs the applications on the basis of MapReduce where the data is processed in parallel and accomplish the entire statistical analysis on large amount of data. It is mainly a data processing component of Hadoop. It is a programming model for processing large number of data sets. It contains the task of data processing and distributes the particular tasks across the nodes. It consists of two phases: One is to Map converts a typical dataset into another set of data where individual elements are divided into key/value pairs. Next is to Reduce task takes the output files from a map considering as an input and then integrate the data tuples into a smaller set of tuples. Always it is been executed after the map job is done.

II. LITERATURE SURVEY

Aseema Sultana [4]: HDFS runs on computers with clusters that extend across a number of racks. In fact, when a file has a replica factor 3, HDFS's Placement policy is to put one replica on one node in the local rack, another on a node in a remote rack, the last on a different node in the same remote rack. This policy cuts the inter-rack write traffic which generally improves write performance.

Dipayan Dev, Ripon Patgiri [5]: The main concern of HDFS block placement policy is minimization of write cost, and maximization of data reliability, scalability and increase the overall bandwidth of the cluster. After the generation of new block, HDFS searches for a location places the first replica on that node, the 2nd and 3rd replicas are stored similarly on two different nodes in a different rack, and the rest are placed on random nodes. HDFS provides a restriction that, more than one replica cannot be stored at one node and more than one replica cannot be stored at one node and more than two replicas cannot be stored in the same rack.

Ch. Bhaskar Vishnu Vardhan and Pallav Kumar Baruah [3]: discussed about initial data placement algorithm starts by evenly dividing the large input file into number of blocks. Then based on the performance capability of the nodes the input file fragments are assigned to nodes in the cluster. Nodes that are having high compute capability are expected to process and store more file fragments compared to nodes with low compute capability. The initial block placement mechanism is to distribute the data blocks to all the nodes in the heterogeneous cluster based on the performance of each node.

Patel Neha M, Patel Narendra M, Mosin I Hasan, Shah Parth D, Patel Mayur M [6]: proposed that HDFS location policy which determines DataNodes to place block replication. HDFS stores files across one or more blocks, so for each block the NameNode is consulted to determine which DataNodes will hold the data for the block. When determining what DataNodes should be used for a block the NameNode first attempts to pick the local node, if the client is running on a DataNode in the order of local disk and Rack-local nodes.

Richa Jain, Amit Saxena, Manish Manoriya [7]: To achieve the simplest I/O performance, one might create replicas of input file of a Hadoop application that every node in an exceedingly Hadoop Cluster contains a native copy of the input file. Such a data replication minimizes the data transfer among slow and quick nodes within the cluster throughout the execution of the Hadoop application.

III. PROPOSED METHODOLOGY

We proposed an approach to overcome the above by suggesting an improvement in the Hadoop default Data Placement Policy. The proposed replica block placement policy consists of two major objectives which has been incorporated into a single algorithm. First, the new dynamic replica placement algorithm distributes blocks across all the DataNodes in the cluster evenly. Second, nodes having higher I/O efficiency would handle more of data.

By implementing the proposed technique for data replica placement is highly expected to increase the overall performance of the cluster.

IV. MAPREDUCE APPLICATIONS

Many Hadoop benchmarks can provide insight into cluster performance. The best benchmarks are always those that reflect real application performance. The two benchmarks are Terasort and TestDFSIO, which provides a good Hadoop installation, is operating and can be compared with public data published for other Hadoop systems. The results should not be taken as a single indicator for system-wide performance on all applications.

Running the Terasort Test

The Terasort benchmark sorts a specified amount of randomly generated data. This benchmark provides combines testing of the HDFS and MapReduce layers of a Hadoop cluster. A full Terasort benchmark run consists of the following three steps

1. Generating the input data via Teragen program
2. Running the actual Terasort benchmark on the input data
3. Validating the sorted output via the Teravalidate program

Running the TestDFSIO Benchmark

Hadoop also includes an HDFS benchmark application called TestDFSIO. It is a read and write test for HDFS. The file size and number of files are specified as command line arguments. A TestDFSIO benchmark run consists of the following three steps

1. Run TestDFSIO in write mode and create data
2. Run TestDFSIO in read mode
3. Clean up the TestDFSIO data

V. CONCLUSION

In this paper, we discussed about Hadoop with the components of HDFS and MapReduce. Data Replication is used to improve the performance of data in HDFS that can increase write throughput with the affect of performance of data. This paper shows that the study on improving performance of data in Hadoop Clusters using data replica placement. The proposed approach provides a significant performance enhancement over the HDFS Dynamic Data Replica Placement Strategy and also discusses the benchmark applications such as Terasort and TestDFSIO.

VI. REFERENCES

[1] Anton Spivak and Denis Nasonov "Data Preloading and Data Placement for MapReduce Performance Inproving" Procedia Computer Science 101, 2016, Pages 379 – 387

[2] Mahesh Maurya, Sunita Mahajan "Performance analysis of MapReduce Programs on Hadoop Cluster" IEEE World Congress on Information and Communication technologies,2012

[3] Ch. Bhaskar Vishnu Vardhan and Pallav Kumar Baruah, " Improving the Performance of Heterogeneous Hadoop Cluster" Fourth International Conference on parallel, Distributed and Grid Computing (PDGC), 2016

[4] Aseema Sultana "Unraveling the Data Structures of Big Data, the HDFS Architecture and Importance of Data Replication in HDFS" International Research Journal of Engineering and Technology (IRJET), Volume:05 Issue :01, Jan 2018

[5] Dipayan Dev, Ripon Patgiri "Performance Evaluation of HDFS in Big Data Management", International Comference on High Performance Computing and Applications (ICHPCA), 2014

[6] Patel Neha M, Patel Narendra M, Mosin I Hasan, Shah Parth D, Patel Mayur M, "Improving HDFS Write Performance Using Efficient Replica Placement", 5th International Conference – confluence The Next Generation Information Technology Summit, 2014

[7] Richa Jain, Amit Saxena, Manish Manoriya, "Analysis of Dynamic Data Placement Strategy for Heterogeneous Hadoop Cluster", International Journal of Emerging Trends and Technology in Computer Science, Volume 4, Issue 4, July – Aug 2015