International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 06 Issue: 03 | Mar 2019 www.irjet.net p-ISSN: 2395-0072

Performing Load Balancing between Namenodes in HDFS

Mahesh Jakkal[1], Shreyasi Goli[2], Aishwarya Dudam[3], Pooja Nilgar[4], Prof. Asiya Khan[5]

[1,2,3,4,5]*Dept. Computer Science and Engineering, BMIT Engg College, Solapur, Maharashtra, India*

Abstract - *An innovative and hottest technology that used in industries, organizations etc. is Hadoop. It most probably used to analyze the large amount of data in distributed processing manner. It just stores the data and run them on commodity hardware clusters. Hadoop provides massive storage for any kind of data, enormous processing power and the ability to handle virtually limitless concurrent tasks or jobs using Mapreduce and HDFS module. The promise of low-cost, security, high-availability storage and processing power has drawn many organizations to Hadoop. Mapreduce perform computations and processing tasks using trackers (job & task), whereas HDFS provide storing data facility using Namenode and Datanode.*

But in today's internet networking world the growth of data is tremendous and to handle them HDFS is perfect architecture, as he is having Namenode and Datanode to perform possible tasks and storing data. Our proposed architecture will help HDFS architecture perform load balancing to handle the SPOF problem nearly of Namenode.

Key Words: Hadoop, HDFS, Namenode, Datanode

1. INTRODUCTION

There are several system architectures that have been implemented for data intensive computing as well as for large-scale data analysis, such as applications having and belongs to parallel and distributed relational database management systems. But, most of the time data growth is in unstructured type form of data, that consists different and combination of so many formats. Mapreduce is a programming paradigm architecture part of Google. Now it is available in an open-source implementation called Apache Hadoop. It is used by organizations, industries like Yahoo, Facebook and other online shopping marts. Data-Intensive Computing Systems have so many approaches to parallelize the processing of data. The goal to design such platform is to provide reliability, efficiency, availability and scalability.

Hadoop is one such architecture which provides above all mentioned features in today's decade. Hadoop parallelizes data processing across many nodes computers in a cluster. It speeds up large computations and hides I/O latency. Hadoop is especially designed and well-suited to large data processing tasks like searching and indexing because it has powerful distributed file system. HDFS is big solution for enterprise that turns ugly ducking into swan.

1.1 Hadoop

Hadoop has emerged as a data mining platform and is becoming an industry standard for large data processing. Hadoop is successfully used in science and a variety of industries. Scientific applications include mathematics, high energy physics, astronomy, genetics, and oceanography. The Hadoop provides a distributed filesystem and a framework for the analysis and transformation of very large data sets using the Mapreduce paradigm. While the interface to HDFS is patterned after the UNIX filesystem, faithfulness to standards was sacrificed in favor of improved performance for the applications at hand.

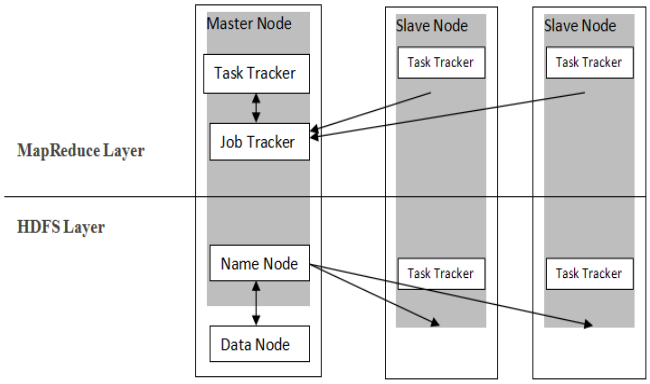

Fig -1: Hadoop System Architecture

1.2 HDFS

The Hadoop distributed file system (HDFS) has Namenode servers and data nodes. The Namenode server maintains the metadata called namespace. Namespace has information about Namenode servers, file, blocks, replica, data nodes and running jobs. HDFS is highly reliable as it replicates chunks of data to nodes in the cluster. The replica decisions are used to improve the availability of system. HDFS stores filesystem metadata and application data separately.

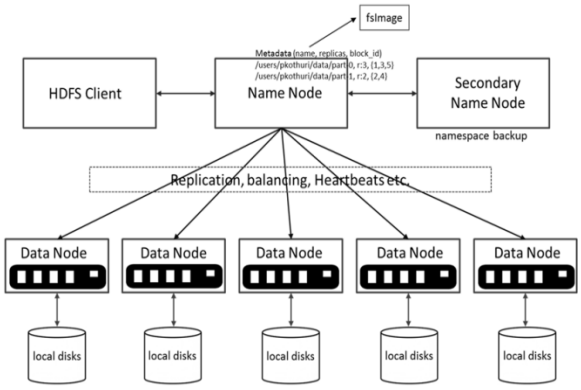

Fig -2: HDFS System Architecture

Hadoop HDFS adopt centralized metadata management solution, but if the load increases towards the centralized master i.e. Namenode, then chance off system goes down are more. It is like SPOF issue i. e. Single Point of Failure.
This issue is going covered in our proposed system that helps to reduce the ratio of failure by performing load balancing within Namenode.

2. Proposed System Architecture

Proposed system will be able to solve the SPOF issue of Namenode nearly by using multiple namenodes. Proposed system architecture consist interconnected machines of Clients, Namenodes and Datanodes.

Here, multiple Namenodes are connected to each other and they are having their respective Datanodes to perform I/O operations.

When Clients sends request to Namenode, it checks the entry of that respective request in the namespace, if present it continue with response. Response contains information about the Datanodes. Clients contact to Datanode as per getting response from Namenode. But if entry does not exit then Namenode create and give the response to the client. Datanode performs the I/O operations request given by client and also send the heartbeats to the Namenode.

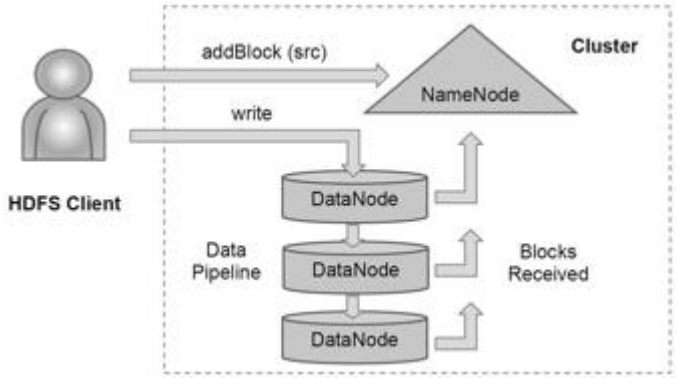

Fig -3: Creating new file of Client

During this process if one Namenode goes down, the other Namenode helps him by balancing his load using Chord system. It provide list to their respective nodes to each other like mirroring concept. So when system starts it just keep the mirror copies of Namenode to another Namenode n vice versa. It will help system to cover up with client request n response execution properly within no time.

Software requirements: Linux OS, Hadoop 2.7.3, jdk 1.6
Hardware requirements: System 32/64bit, HDD–10 GB, RAM-2 GB

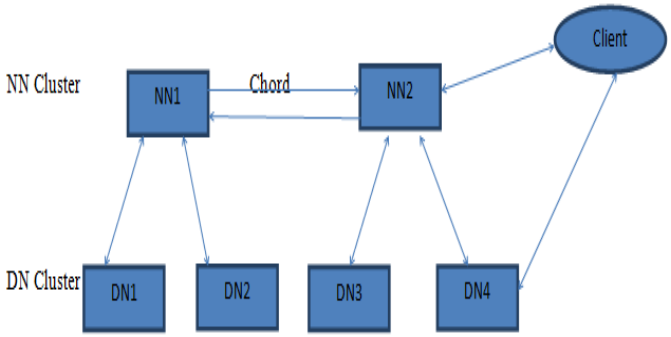

Fig -4: Proposed Architecture of HDFS System

3. CONCLUSION

The proposed architecture will utilize multiple namenodes to result in good scalability and availability of Namenodes without any downtime. Also it this project work shows the naming flexibility of namespace and load balancing too.

REFERENCES

1. Apache™ Hadoop®. Hadoop documentation, http://hadoop.apache.org/,(2014)February11.
2. J. Cui, T. S. Li and H. X. Lan,"Design and Development of the mass data storage platform based on Hadoop", Journal of Computer Research and Development, vol. 49, (2012), pp. 12-18.
3. Towards a scalable HDFS architecture, Farag Azzedin IEEE 2013.
4. Load rebalancing for Distributed File System in Clouds, IEEE transactions on Parallel and distributed system.
5. Chord: A Scalable Peer-to-peer Lookup Protocol for Internet Applications.

AUTHORS

Mr. Mahesh Jakkul
Studying Engineering in Bhramhadevdada Mane Institute of Technology, as BE (CSE) student. Interest area are in Technical languages and new techs.

Ms. Shreyasi Goli
Studying Engineering in Bhramhadevdada Mane Institute of Technology, as BE (CSE) student. Interest area in Technical languages.

Ms. Aishwarya Dudam
Studying Engineering in Bhramhadevdada Mane Institute of Technology, as BE (CSE) student. Interest area in Technical languages.

Ms. Pooja Nilgar
Studying Engineering in Bhramhadevdada Mane Institute of Technology, as BE (CSE) student. Interest area in Technical languages.

Prof. Ms. Asiya Khan
Working in Bhramhadevdada Mane Institute of Technology, as Assistant Professor for CSE department. Interested areas are Database, Image processing etc.