Disk Balancer System for Distributed Big Data Storage in Hadoop

Rajashekar K J [1]
Research scholar
Sri Siddhartha Academy of Higher Education
Tumakuru.

Dr. ChannakrishnaRaju [2]
Associate Professor, Dept. of CS & E
Sri Siddhartha Institute of Technology
Tumakuru

Abstract: Hadoop Distributed File System (HDFS) is a Hadoop module used to store the Big Data very effectively in parallel on Data Nodes. HDFS may not generally place data in a uniform manner across the disks because of parcel of write, delete, and disk replacement tasks. This can prompt huge slant inside a Data Node. This circumstance isn't taken care of by the current HDFS balancer, which worries about Inter, Non-Intra, and DN slant. This circumstance is dealt with by the new Intra-Data Node Balancing usefulness, which is summoned by means of the HDFS Disk Balancer. This paper explores the working and functionality of Hadoop Disk Balancer.

Key Words: Hadoop, HDFS, Big Data, Disk Balancer.

I. INTRODUCTION

Big data refers to Data sets that are so large volume of data which seems to be beyond the ability to process by any application processing software tools to store, manage, analyze and process [1]. As technology advances over time, the amount of data collected is increasing in all aspects, here the real challenge is to identifying or designing most reliable methods for extracting value from the huge amounts of scalable data.

Big data analytics involves collecting data from different sources to process large and varied data sets. It requires high performance computing power and an ability to access and integrate large and varied data sources, which tends to the execution of many discrete analytics routines. The real issue is execution of many discrete analytics and the ability to access very complex sets of data that make up larger data and model work flows [2]. Big data analytics involves data analytics technologies and techniques provide a means of analyzing data sets and drawing conclusions about them. The different difficulties and issues in adjusting and tolerating Big information innovation, with its significance in the advanced world requires versatile arrangements that support distributed computing on multiple servers, dynamic workload balancing, data integration, high availability, and job prioritization.
Hadoop is an apache open source structure written in java that permits conveyed handling of enormous datasets across bunches of PCs utilizing straightforward programming models [3].

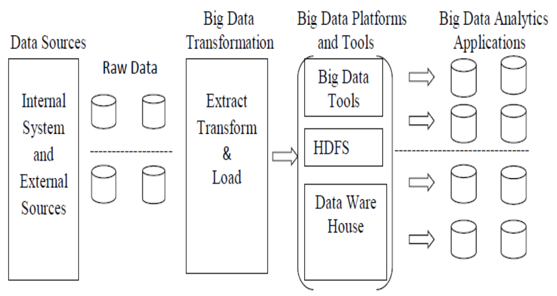

Figure1. **Conceptual Structure of Big Data Analytics.**

Big data analytics conceptual structure which uses the concept of Hadoop is depicted in Figure1. Big data analytics applications regularly incorporate data from both inside frameworks and outer sources; Hadoop serves as the primary repository for incoming streams of raw data for efficient processing of large amounts of data on large clusters in a reliable analytics process. Once the data is ready, it can be analyzed with the analytical processing software's and being stored in the Hadoop Distributed File System (HDFS) at long last data should be coordinated and designed to get great execution in conveyed applications. Apache Hadoop is created for organized datasets as well as cycle unstructured datasets [4][11].

The two main modules of Hadoop are HDFS and MapReduce. HDFS is the Hadoop Distributed File System module used to solve the problem of storing big data. This segment is intended to store extremely enormous informational indexes which gives high throughput admittance to application data and is appropriate for applications that have huge datasets. MapReduce is a programming model for processing large data set in distributed and parallel processes stored inside the HDFS. With the privilege large data investigation stages set up, the framework can expand effectiveness and improve execution. Best performance can be achieved in Big Data analytics through the use of SSD's, which provides an ability to separate data and metadata. It can also help in linear scalability of I/O, and no single point of failure, hence SSD's are the solutions to optimize I/O and throughput. They are adaptable to any workload, extremely scalable in capacity and density. HDFS may not generally place data in a uniform manner across the disks because of part of write, delete and disk replacement activities. This can prompt huge slant inside a Data Node. This circumstance isn't taken care of by the current HDFS balancer, which worries about Inter, Non-Intra, and DN slant. This circumstance is taken care of by the new Intra-Data Node Balancing usefulness, which is conjured through the HDFS Disk Balancer CLI [5][9][10].

II. BACKGROUND

Apache Hadoop is a structure for running applications on huge group worked of ware equipment. The Hadoop system straightforwardly gives applications both dependability and data movement. Hadoop actualizes a computational worldview named Map/Reduce, where the application is isolated into numerous little parts of work, every one of which might be executed or re executed on any hub in the bunch. What's more, it gives a conveyed document framework (HDFS) that stores data on the PC hubs, giving extremely high total data transfer capacity across the cluster.

Both MapReduce and the Hadoop Distributed File System are planned with the goal that hub disappointments are consequently taken care of by the structure [6][7][12]. The Hadoop Distributed File System (HDFS) is a circulated record framework intended to run on ware equipment. It has numerous likenesses with existing appropriated record frameworks. Notwithstanding, the distinctions from other conveyed document frameworks are huge. HDFS is exceptionally flaw lenient and is intended to be sent on ease equipment. HDFS gives high throughput admittance to application data and is appropriate for applications that have huge data collections. HDFS loosens up a couple of POSIX prerequisites to empower streaming admittance to record framework information. HDFS was initially worked as framework for the Apache Nutch web crawler venture. HDFS is important for the Apache Hadoop Core venture [8].

Advantages of Hadoop

Ability to store and process huge amounts of any kind of data, quickly: With information volumes and assortments continually expanding, particularly from web-based media and the Internet of Things (IoT), that is a key thought.

Computing power: Hadoop's distributed computing model processes big data fast. The additionally figuring hubs you utilize the all, the more handling power you have.

Fault tolerance: Data and application preparing are secured against equipment disappointment. On the off chance that a hub goes down, positions are naturally diverted to different hubs to ensure the appropriated processing doesn't come up short. Numerous duplicates of all information are put away naturally.

Flexibility: In contrast to customary social databases, you don't need to pre-measure data prior to putting away it. You can store as much data as you need and conclude how to utilize it later. That incorporates unstructured data like content, pictures and recordings.

Low cost: The open-source structure is free and uses item equipment to store huge amounts of data.

Scalability: You can undoubtedly develop your framework to deal with more information just by adding hubs. Little organization is required.

III. DISK BALANCER

Disk balancer is an command line tool that appropriates data equally on all disks of an data hub. This tool is not the same as Balancer which deals with cluster wide data adjusting. Data can have uneven spread between disks on a hub because of a few reasons. This can occur because of enormous measure of writes and deletes or because of disk replacements. This apparatus works against a given data hub and moves blocks starting with one disk then onto the next. Disk Balancer works by making an arrangement and proceeds to execute that arrangement on the data hub. An arrangement is a bunch of articulations that portray how much data should move between two disks. An arrangement is made out of numerous move steps. A move step has source disk, destination disk and number of bytes to move. A plan can be executed against an operational data hub. Disk balancer ought not to meddle with different disks since it chokes how much data is copied each second. Please note that disk balancer isn't empowered naturally on a cluster. To empower disk balancer, dfs.disk.balancer.enabled should be set to valid in hdfssite.xml [9]. HDFS gives a command line tool called Disk balancer. It distributes data in a uniform manner on all disks of a data node.

Need for HDFS disk Balancer

In Hadoop HDFS, Data Node distributes data blocks between the disks on the Data Node. While writing new blocks in HDFS, Data Nodes chooses volume-choosing policies (round-robin policy or available space policy) to choose disk (volume) for a block depicted in figure2.

1. **Round-Robin policy**: This spread the new blocks uniformly among the disks available. Data Node uses this policy by default.
2. **Available space policy:** This policy writes data to those disks that have more free space (by percentage).

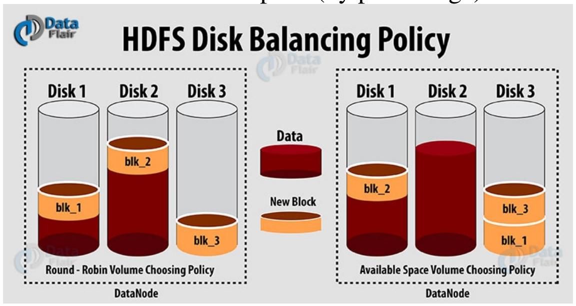

Figure2. HDFS Disk Balancing Policy

However, with round-robin policy in the long-running cluster, Data Nodes sometimes unevenly fill their storage directories (disks/volume), leading to situations where certain disks are full while others are significantly less used. This happens either because of large amounts of write and deletes or due to disk replacement. Also, if we use available-space based volume-choosing policy, at that point each new write will go to the recently added empty disk making different disks idle during the period. This will create a bottleneck on the new disk. Thus, there arises a need for Intra Data Node Balancing (even distribution of data blocks within Data Node) to address the Intra-Data Node skews (uneven distribution of blocks across disk), which occur due to disk replacement or random writes and deletes. Therefore, a tool named Disk Balancer was introduced in Hadoop 3.0 that focused on distributing data within a node.

Operation of Disk balancer

HDFS Disk Balancer in Hadoop neutralize given data hub and moves blocks starting with one disk then onto the next. Hadoop HDFS Disk balancer works by making an plan (set of statements) and playing out that plan on the data hub. Plan portrays how much data should move among two disks. Plan has many move steps. Each Move step has source disk, destination disk and various bytes to be moved. An plan can execute against an operational data hub. To empower Disk balancer dfs.disk.balancer.enabled should be set to true in hdfs-site.xml. By default, it will be disabled.

The **plan** command will generate the plan for the specified DataNode and can be run against a given DataNode. With some additional options hdfs.diskbalancer plan command allows users to have control over output process and execution of a plan. The plan will be executed with **execute** command against the DataNode. The JSON document contains the generated plan (nodename.plan.json). The existing status of the HDFS disk balancer from a DataNode will be achieved by **query** command. The **cancel** command cancels the running plan. The detailed report of the specified DataNodes can be taken from the **report** command.

Functions of HDFS Disk Balancer

HDFS Disk balancer supports two major functions i.e, reporting and balancing.

Data Spread Report

In order to define a way to measure which machines in the cluster suffer from the uneven data distribution, the HDFS disk balancer defines the HDFS Volume Data Density metric and the Node Data Density metric. HDFS Volume data density metric allows us to compare how well the data is spread across different volumes of a given node. The Node data density metric allows comparison between nodes.

Volume Data Density

Volume data density metric figures how much data exists on a node and what should be the ideal storage on every volume. The ideal storage rate for every device is equivalent to the absolute data put away on that node partitioned by the complete disk limit on that node for every storage type.

Node Data Density or Inter-Node Data Density

After calculating volume data density, one can calculate Node Data Density, which is the sum of all absolute values of volume data density. This allows comparing nodes that need our attention in a given cluster. Lower node Data Density values indicate better spread, and higher values indicate more skewed data distribution.

Reports

When we have volume Data Density and node Data Density, we can locate the top 20 nodes in the cluster that slanted data distribution, or we can get the volume Data Density for a given node.

Disk balancing

Once we know that a certain node needs balancing, we compute or read the current volume Data Density. With this information, we can easily decide which volumes are over-provisioned and which are under-provisioned. With this to move data from one disk to another in the Data Node, we would add a protocol-based RPC similar to the one used by the balancer. Thus, allowing the user to replace disks without worrying about decommissioning a node. Commands are supported by HDFS Disk Balancer to perform the task.

Plan

The plan command generates the plan for the specified Data Node. This command can be run against a given Data Node. There are some additional options that can be used with hdfs diskbalancer plan command which allows the users to have control over the output and execution of a plan.

IV. CONCLUSION

The disk Balancer is a tool that distributes data blocks between the volumes of the Data Node. Thus, it addresses Intra-Node skew. Also, the disk balancer moves data from one volume to another within nodes while nodes are alive, so users can replace disks without having to worry about decommissioning a node. HDFS Disk Balancer support some of the commands like plan for generating a plan, execute for executing the plan against Data Node, query for querying the status of disk balancer, cancel for cancelling the plan, etc.

REFERENCES

[1]. Stephen Kaisler, Frank Armour, J. Alberto Espinosa, William Money, "Big Data: Issues and Challenges Moving Forward", IEEE, 46th Hawaii International Conference on System Sciences, 2013.

[2]. Michael Minelli, Michele chambers, Ambiga Dhiraj, " BIG DATA BIG ANALYTICS", Emerging Business Intelligence and Trends for Today's Businesses, WILEY CIO Series.

[3]. Avita Katal, Mohammad Wazid, R H Goudar, "Big Data: Issues, Challenges, Tools and Good Practices", 978-1-4799-0192-0/13/$31.00 ©2013 IEEE Network.

[4]. S. Chandra and D. Motwani, "An Approach to Enhance the Performance of Hadoop MapReduce Framework for Big Data,"2016 International Conference on Micro-Electronics and Telecommunication Engineering (ICMETE), GHAZIABAD, India, 2016, pp. 178-182. doi: 10.1109/ICMETE.2016.64.

[5]. https://data-flair.training/blogs/hadoop-hdfs-diskbalancer/.

[6]. https://hadoop.apache.org/docs/r3.0.0/hadoop-project dist/hadoop/

[7]. https://hadoop.apache.org/docs/current/hadoop-project-dist/hadoop-hdfs/HDFSDiskbalancer.html

[8]. "Data Balancing Scheme for Multi-node Heterogeneous Hadoop Cluster", International journal of Science and Research (IJSR), Volume 6, Issue 3 March 2017.

[9]. "A Load-Balancing Algorithm for Hadoop Distributed File System", International Conference on Network based information systems, IEEE, September 2015.

[10]. Hadoop Technology and Architecture series 1, 2, and 3, EMC2 Education series.

[11]. T. White, "Hadoop: The Definitive Guide" in, O'Reilly Media, Yahoo! Press, June 2009.

[12]. J. Dean and S. Ghemawat, "MapReduce: Simplified Data Processing on Large Clusters", Proc. of the 6th Symposium on Operating Systems Design and Implementation San Francisco CA, Dec. 2004.