International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2015): 78.96 | Impact Factor (2015): 6.391

A Survey on Performance and Security of Hadoop

Swapnali A. Salunkhe[1], Amol B. Rajmane[2]

Department of Computer Science and Engineering, Ashokrao Mane Group of Institutions, Vathar, Maharashtra, India

Abstract: ***Hadoop Framework is used to process big data in parallel fashion. A big data is not only big in size but also it is in different format, different size and with different speeds. To process big data relational database management system is not a suitable one. The hadoop is a most popular framework to process big data. Hadoop framework architecture has many components like name node data, data node, job tracker and task tracker. The performance of hadoop is dependent on how these components execute. The challenge in Hadoop framework is to reduce processing time of job, but these challenges are depend upon various factors like scheduling, performance of map reduce after data encryption, resource allocation, and data encryption. Proposed research is focused on how to overcome these challenge of scheduling, resource allocation, and security. Hadoop data security is also a proposed research area i.e. to find a best suitable encryption algorithm which encrypts hadoop data without affecting hadoop performance.***

Keywords: Hadoop, Bigdata, Map Reduce, Real Time Encryption Algorithm

1. Introduction

In the current information age the requirement of data is increasing day by day. The data generated from different sources is in terabytes per day, which is called as big-data. Big-data is not just big in size, but big data have data of different variations, different sizes and at different speed. This big-data is used for many applications and business related services like business intelligence. To store and process this large amount of data we need an efficient and fault tolerant system. Google developed a file system called as Google file system to handle big data. Hadoop is based on the Google's file system. Hadoop is open source software framework to store and process this big data efficiently. It is designed in java language. HDFS (Hadoop Distributed File System) and Map Reduce are the two components of Hadoop. Map reduce is implementation of Hadoop system for cloud, map reduce is a programming model to write applications for processing big data. Hadoop is used by many organizations like Yahoo, Google, Facebook and it is maintained by Apache Foundation. Map Reduce is implemented with the help of two components: a job tracker and multiple task trackers. The job tracker is responsible to command the task trackers through two main functions i.e. map tasks and reduce tasks, the task trackers used to process data as per the commanded by job tracker. Job tracker is also in-charge of scheduling map task and reduces task to task trackers, it assigns job to the task trackers and also collects the intermediate results. Hadoop has namenode and datanode to manage and process data. Name node is node which stores file system metadata, and data node is actually store the data.

2. Hadoop System

The current Hadoop framework does not support two important features first is encryption of storing HDFS blocks and computation on such encrypted data which is a fundamental solution for secure Hadoop, and second is if same data is occurred then what should be the processing strategy. To overcome these two problems we need a principled way for the encryption process, and to minimize the time of file encryption and job execution (file decryption) and compare duplicate input data to avoid processing of same data multiple times. Input to proposed system is multiple numbers of files; the system will first encrypt files and then load at HDFS, then execute the job on data at HDFS on user request. At the time of job execution; it needs to perform decryption

Internet now generating large amount of data every day, International Data Corporation published a statistics which include the structured data on the Internet is about 32% and unstructured is 63%. Also the volume of digital content on internet grows up to more than 2.7ZB in 2012 which is up 48% from 2011 and now rocketing towards more than 12ZB by 2016. Market survey tells that big data is beneficial for productivity growth.

In commercial data analysis applications which operate on big data, Hadoop becomes important platform. In upcoming 5 years, more than 50% of big data applications will execute on Hadoop.

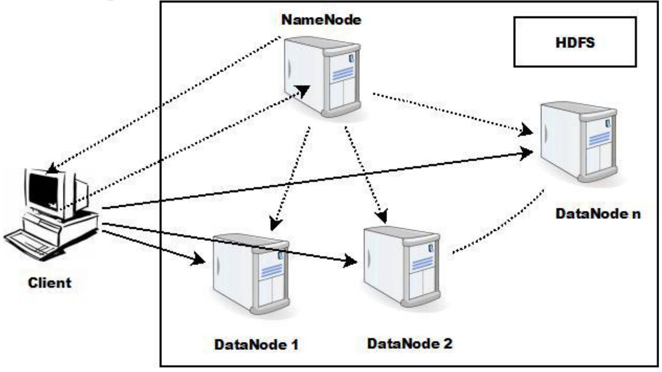


Figure 1: HDFS Architecture

File on HDFS splits into multiple blocks and replicated into multiple Data Nodes to ensure high data availability and durability to avoid failure of execution for parallel application in Hadoop environment. Originally Hadoop clusters have two types of nodes i.e. master-slave. Name Node as a master and Data Nodes are workers nodes of HDFS. Data files which are located in Hadoop are stored in Data Node which only stores data. However Name Node contains information about where the different file blocks are located but it is not persistent, when system starts block may changes one Data Node to another Data Node but it report to

Name Node or client who submit the Map Reduce job or owner of Data periodically [12]. The communication is in between Data Node and client Name Node only contains metadata.

3. Security Risks in HDFS

Hadoop uses 'whoami' and 'bash –c groups' utility of unix for individual user and groups respectively, this is the weak point because which permissions and file quota are for clients is unpredictable. There are three kinds of security violations in HDFS, unauthorized access, unauthorized modification of data and denial of service or resource.

Following are the areas where threat identify in Hadoop

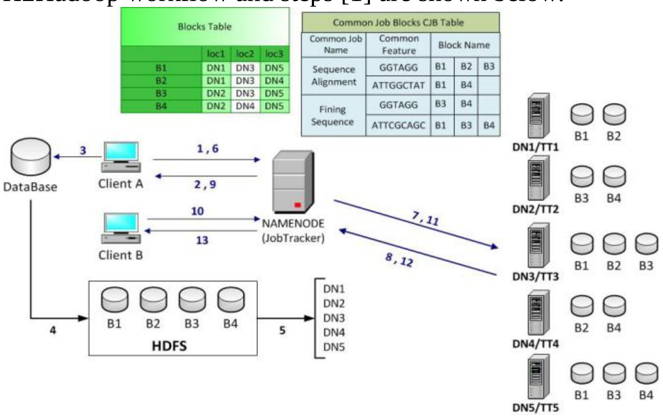
- Hadoop does enforce authentication to any user or service: unauthorized users may any HDFS cluster like owner via RPC of HTTP protocol.
- Data Node can't have any access control mechanism to protect data block: it is possible to write or modify existing data blocks to Data Node.
- An attacker can present as Hadoop service: For example, code submitted by user register itself on Map Reduce cluster as a new Task Tracker
- Super-user of system does anything without checking: User who takes control of Name Node is a super-user; it means somebody started the Name Node which has fully access on HDFS data.
- An executing Map Reduce may use the host operating system interfaces: Some time execution of Map Reduce demands access to other tasks on the host OS, access local storage for instant Map output, but both executing on the same physical node.

4. Literature Survey

Hamoud Alshammari, Jeongkyu Lee, Hassan Bajwa[1] proposed architecture related to manipulating big data that uses different parameters in the processing jobs. Author focuses on the limitations of hadoop and cloud. Study shows that the limitations are mostly because of data location, scheduling of task tracker and data tracker and resource allocation. Cloud computing requires efficient resource allocation so to improve performance. The H2Hadoop proposed by author focuses on reduction in computation cost for big data. Author also proposed architecture for efficient resource allocation. The architecture provides better solution for text data and efficient data mining approached for cloud computing. H2Hadoop provides separate control feature to name node so that name node can intelligently assign data to task trackers so without using data of whole cluster. The results of this paper show that there in reduction in CPU time, number of read operation and some other factors. The problem statement of author focuses on identifying the sequences in large unstructured data. But with single node it is time consuming and expensive. Hadoop cluster with three nodes is able to identify the sequence more efficiently. But, when we try to execute a Map Reduce job on the same cluster for more than one time, then each time it takes the same amount of time. So this study aims to solve this problem and propose a solution which will improve the time involved in the execution of Map Reduce jobs.

Author [1] defined some terminologies to show the solution. First terminology is CJBT i.e. Common Job Block table. It is a look table designed specifically for name node to identify data. Second terminology is CJN i.e. Common Job Name. It is a shared name which must be used by each user while submitting a job in order to get benefits of proposed architecture. The third terminology is CF i.e. Common Features shows the shared data between jobs. It is used to identify the data nodes having common data features. The next terminology is BN i.e. Block Name or Block ID used to show the location of shared features. It helps name node to directly assign jobs to data node.

H2Hadoop workflow and steps [1] are shown below.



Blocks Table	loc1	loc2	loc3
B1	DN1	DN3	DN5
B2	DN1	DN3	DN4
B3	DN2	DN3	DN5
B4	DN2	DN4	DN5

Common Job Blocks CJB Table

Common Job Name	Common Feature	Block Name		
Sequence Alignment	GGTAGG	B1	B2	B3
	ATTGGCTAT	B1	B4	
Fining Sequence	GGTAGG	B3	B4	
	ATTCGCAGC	B1	B3	B4

Figure 2: H2Hadoop Workflow [1]

Map Reduce workflow in H2Hadoop has been explained in figure 1 as follows:

Step 1 to Step 8: remain in the same workflow as native Hadoop except results from the first 7 steps are stored in the CJBT.

Step 9: Job Tracker sends the result to Client "A". In this step, Name Node keeps the names of the blocks that produced the results in the local lookup table (CJBT) by the Common Job Name (Job1) that has common feature as explained above.

Step 10: Client "B" sends a new Map Reduce job "Job2" to the Job Tracker with the same common job name and same common feature or super-sequence of "Job1".

Step 11: Job Tracker sends "job2" to Task Trackers who hold the blocks, which have the first result of the Map Reduce "Job1" (DN2, DN4, DN5). In this step, the Job Tracker starts with checking the CJBT first to find if it is a new job which has the same common name and common features of any previous ones or not – In this case yes. Then the Job Tracker sends "Job2" only to TT2, TT4 and TT5. We may assume here that the lookup table will be updated with more details OR just remain as is because every time we have a new job that may carry the same name of "Job1".

Step 12: Task Trackers execute the tasks and send the results back to the Job Tracker.

Step 13: Job Tracker sends the final result to Client "B".

Weijia Xu*, Wei Luo, Nicholas Woodward [2] evaluated cost of importing large scale data into hadoop cluster. Author proposed detailed evaluation and implementation for importing large scale data into hadoop. They also proposed method for improving performance in hadoop for importing large scale data.

Herodotos [3] proposed a performance model for improvement of hadoop performance.

Mohammad Hammoud and Majd F. Sakr [4] proposed locality aware for reducing and improving the map reduce performance. They uses network locations and size of reducers in order of network traffic for improving Map Reduce performance .For locality aware technique avoids scheduling delay ,poor system utilization and low degree of parallelism.

Min Chen · Shiwen Mao · Yunhao Liu [05] discussed the several challenges occurred during development of big data applications. The challenges include Data representation, redundancy reduction and Data Compression, Data lifecycle management, Analytical mechanism, Data confidentiality, Energy management, Expendability and scalability, Cooperation. They also mentioned relationship between cloud computing and big data

Jeffrey Dean and Sanjay Ghemawat [6] describes how map reduce job runs on large clusters of commodity machines and is highly scalable.

Jinshuang Yan, Xiaoliang Yang, RongGu, Chunfeng Yuan, and Yihua Huang [7] proposed parallel computing framework for solving the problem of data intensive applications. In this paper reduce the time cost of initialization and termination stage, pull model is replaced by push model with task assignment mechanism and message communication mechanism between task tracker and job. They also analyzed and identified two critical limitations of Map Reduce execution mechanism and that are achieved by implementing new job setup/cleanup tasks. In this paper author improved hadoop performance by using job scheduling and job parameter optimization level. The author implemented SHadoop framework that achieve stable performance improvement by around 25% benchmarks without losing scalability and speedup.

Songchang Jin, Shuqiang Yang, Xiang Zhu, and Hong Yin [8] proposed a new encryption technology known as fully homomorphic encryption technology and authentication agent technology for a file system. This method ensures the reliability and security form three levels of hardware, data and users operations. It offers variety of access control rules for data stored in hadoop file system.

Jian Tan, Shicong Meng, Xiaoqiao Meng, Li Zhang [09] data locality is difficult for large scale hadoop clusters. Proposed solution is based on greedy approach i.e. reduce task is placed close to the majority of intermediate data already generated. The side effect is that, in presence of job arrivals and departures, assigning the ReduceTasks of the current job to the nodes with the lowest fetching cost can prevent a subsequent job with even better match of data locality from being launched on the already taken slots.

Changqing Ji , Yu Li , Wenming Qiu , Uchechukwu Awada , Keqiu Li [10] proposed systematic flow of big data using cloud computing. They discussed issues like cloud storage and computing architecture. Proposed system shows big data processing technique and cloud data management. In this paper they introduce problems on cloud computing platform, cloud architecture, cloud database and data storage scheme.

Ahmed H. Omari, Basil M. Al-Kasasbeh [11] proposed encryption algorithm to provide security for real time applications. Authors proposed new cryptographic technique for improving time of encryption and decryption algorithm.

Current Hadoop Framework does not support storing metadata of previous jobs; it ignores the location of Data Node with sub-sequence and reads data from all Data Nodes for every new job. So author proposes new architecture i.e. H2Hadoop. [11]

Xuhui Liu1, Jizhong Han, Yunqin Zhong, Chengde Han [12] says HDFS is designed to handle large files but it suffers with performance when large amount of small file are provides as input to HDFS. The proposed solution is to combine small files to large one with building index of files to keep track of small files. And the preliminary experiments show that this method improves performance.

5. Conclusion

The study of various research articles focuses to improve Hadoop performance with different methods and algorithms are suggested. Different factors related to performance are requires improvement. The hadoop data security is also area where there is no method suggested. So there is need to improve and test the performance with security.

References

[1] Hamoud Alshammari, Jeongkyu Lee and Hassan Bajwa "H2Hadoop: Improving Hadoop Performance using the Metadata of Related Jobs" IEEE TRANSACTIONS ON Cloud Computing, manuscript ID TCC-2015-11-0399.
[2] Weijia Xu*, Wei Luo, Nicholas Woodward "Analysis and Optimization of Data Import with Hadoop" 2012 IEEE 26th International Parallel and Distributed Processing Symposium Workshops & PhD Forum.
[3] Herodotos "Hadoop Performance Models" 6 Jun 2011.
[4] Mohammad Hammoud and Majd F. Sakr "Locality-Aware Reduce Task Scheduling for Map Reduce" 2011 Third IEEE International Conference on Coud Computing Technology and Science
[5] Min Chen, Shiwen Mao, Yunhao Liu "Big Data: A Survey" Springer Science+Business Media New York 2014.

[6] Jeffrey Dean and Sanjay Ghemawat “Map Reduce: Simplified Data Processing on Large Clusters” Google Inc.

[7] Jinshuang Yan, Xiaoliang Yang, Rong Gu, Chunfeng Yuan, and Yihua Huang “Performance Optimization for Short Map Reduce Job Execution in Hadoop” 2012 Second International Conference on Cloud and Green Computing

[8] Songchang Jin, Shuqiang Yang, Xiang Zhu, and Hong Yin ―Design of a Trusted File System Based on Hadoop 2013

[9] Jian Tan, Shicong Meng, Xiaoqiao Meng, Li Zhang, Improving ReduceTask Data Locality for Sequential Map Reduce Jobs, 2013 Proceedings IEEE INFOCOM

[10] Changqing Ji , Yu Li , Wenming Qiu , Uchechukwu Awada , Keqiu Li “Big Data Processing in Cloud Computing Environments ”, 2012 International Symposium on Pervasive Systems, Algorithms and Networks.

[11] Ahmed H. Omari , Basil M. Al-Kasasbeh” A New Cryptographic Algorithm for the Real Time Applications”, Proceedings of the 7th WSEAS International Conference on INFORMATION SECURITY and PRIVACY (ISP '08)

[12] Xuhui Liu1, Jizhong Han, Yunqin Zhong, Chengde Han, “Implementing WebGIS on Hadoop: A Case Study of Improving Small File I/O Performance on HDFS”