International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2017): 7.296

Hadoop Performance Improvement using Metadata and Securing with Oauth Token

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

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

Abstract: ***Initially Hadoop was designed without performance and security aspects. It was just used to process big data in parallel fashion. But now a day's user needs big data with high speed and with security features. Hadoop has some limitations while executing the job. These limitations are reduces the efficiency of hadoop and increases the job execution time. It is mostly because of the job processing method of current hadoop system, scheduling and resource allocation. The proposed system replaces the current job processing method by using Oauth token and Real time encryption algorithm. Proposed system matches a new job with previously executed jobs. If a match found then same results are displayed to user and if not then it will execute new job. If a matching rate is high then execution time will automatically decreases. The proposed system also focuses on security constraints of current hadoop system. Current hadoop system secures data while uploading and downloading it from system. Data in hadoop system is secured with OAuth authorization token token with AES algorithm for encryption and decryption. Proposed system authorizes hadoop user and encrypts data while uploading and downloading data from hadoop system. The encryption time of data is also less so it does not affect the performance of hadoop. So the proposed system decreases the execution time by metadata matching technique and secures with real time encryption algorithm.***

Keywords: Hadoop, Bigdata, Map Reduce, OAuth token, Real Time Encryption Algorithm.

1. Introduction

Hadoop was originally designed from Google File System. It uses highly scalable distribute programming which made this more popular. It is basically used for data intensive applications and real time analytics. 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. 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.

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 features 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 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.

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 known as 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. 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 approach 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 is reduction in CPU time, number of read operation and some other factors.

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] describe 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 the algorithm 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 drawback in this technology is, 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 introduced 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] described the 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. The preliminary experiments show that this method improves performance.

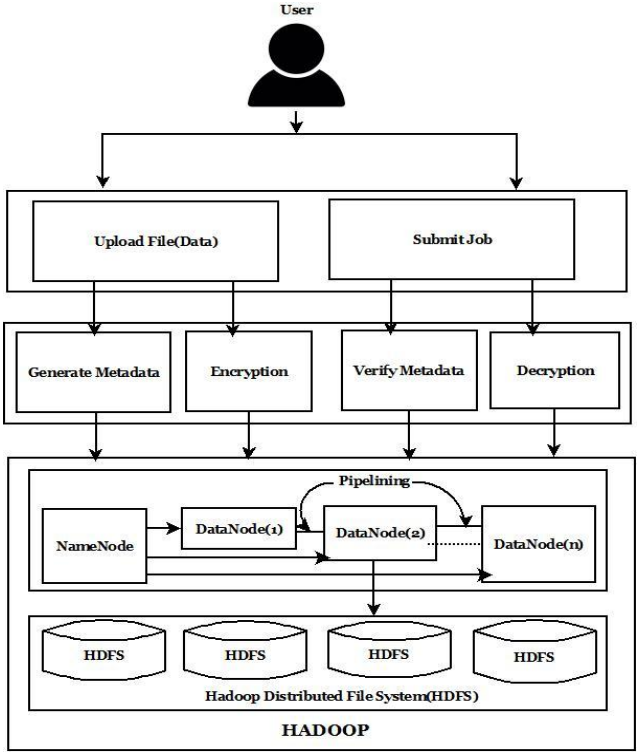

Figure 1: System Architecture

4. Proposed System

The proposed system architecture shows the overall process of system. The user first starts the hadoop system and logins to the system and provides n number of documents as input to HDFS. But before writing it sends it to map reduce programming model. This model checks the submitted job with metadata repository. If a match found then no need to execute the user job and then directly displays the result to hadoop user. It is because the job submitted by user is already executed by some other user and that results are stored in repository. This reduces job execution time for duplicate jobs.

In case of no match the job will be executed by map reduce system. Map-Reduce programming model perform data encryption, similarly it also perform decryption when user read data from HDFS. Hadoop also record each activity of current logged in users as audit log. OAuth provide authentication token and authorization token which are used for user verification and encryption/decryption algorithm (with AES) respectively.

A. Duplicate job identification

To identify duplicate file a file reader is used and some random contents of data from that file and taken as input file and these contents are matched with metadata of file which is already on HDFS. If file duplication found then new file is not uploaded to HDFS instead the same file is used file job execution. To encrypt the data the encryption and decryption algorithm is given below.

Steps:
Input: User File and Map reduce job.
Output: Results of that job.

1. Start
2. Get the input file x from user.
3. Pick random contents from input file x[random]
4. Match the random contents with the metadata m from metadata repository.
5. if(x[random]==m)
   then display the result of same metadata file
   else
   execute the job with input file x
6. Display the result of newly executed job and store it in metadata repository for future reference
7. Exit

B. OAuth 2.0

It is an open authentication protocol which helps to rake over the problems of conventional client server model. In the conventional client-server model, the client requests to an access protected resource on the server by authenticating itself using the resource owner's passport. In order to give third-party applications access to restricted resources, the resource owner verifies its authorization with the third-party.

Hadoop user registered with system using OAuth server generates two types of token. First token is used as authentication token and second is for authorization

Roles in OAuth:-

1) Resource Owner: - A user who stores data into system.
2) Resource Server: - A Server that holds all resources e.g. name node in hadoop.
3) Authorization Server: - An application which requests for data process.
4) Client: - A Server which grants access for resources after authentication.

C. User Login

OAuth is an Open Authentication Protocol used to verify a user authenticity. Before OAuth Kerberos was used for primary authentication in Hadoop with SASL/GSSAPI to mutually authenticate users, their applications, and Hadoop services over the RPC connections. Hadoop also supports .irrecoverable authentication for HTTP Web Consoles meaning that implementers of web applications and web consoles could implement their own authentication mechanism for HTTP connections [13]. There are many data flows in current Hadoop authentication. It is easy and more secure than three steps techniques by Kerberos. We propose OAuth 2.0 authentication protocol to verify user, it provide two different types of tokens for authentication as well as authorization.

D. User Authentication Token

OAuth 2.0 authorization server gives the unique 64-bit token to each registered user. When user access files from distributed storage then token is used to verify. For user authentication OAuth server is responsible.

E. User Authorization Token

Proposed system provides User authorization token when user registers with authorization server. After successful registration with system, authorization token is issued to user. Token which is provided by system is 64-bit and used

for providing data security and privacy amongst different users of Hadoop. It is unique for each user and it also grants access to user who is accessing files or executing jobs. OAuth 2.0 token has a refresh mechanism or expiry technique which adds more security.

F. File Encryption

The encryption can be done in two ways. First, when file is stored in Hadoop, the complete file can be encrypted first and then stored in Hadoop. In this approach, the data blocks in each Data Node can't be decrypted until we put all the blocks back and create the entire encrypted file. Second, by applying encryption to data blocks once they are loaded in Hadoop system [14]. Proposed system encrypts all files before storing it in HDFS. The simple authentication and security layer (SASL) framework is used for encrypting the data in Hadoop. Hadoop supports encryption capability for various channels like RPC, HTTP, and Data Transfer Protocol for data in motion. HDFS supports AES, OS level encryption for data at rest. We have proposed ASE-OAuth algorithm to protect data over HDFS, it include ASE and OAuth 2.0 authorization token.

G. Job Execution and Decryption Module

User accesses files from HDFS, for that they submit Map Reduce job. But before Map Reduce Process the data, first the data is decrypted by every Mapper and then by reducer. The decryption is reverse of the encryption process i.e. AES and OAuth authorization token.

H. Audit Log

Hadoop stores sensitive information and security of this information is important for organizations. So to meet these security requirements, we need to audit the entire Hadoop system on a periodic basis. Hadoop does not provide built-in audit logging, so we can use audit logs activity recording tools. Scribe and LogStash are open source tools that integrate into most big data environments. We propose audit record model which records each activity did by every logged user including super-user.

5. Experimental Setup and Results

Proposed system has two different encryption techniques first does encryption using AES and second Real time encryption using OAuth token . The hadoop job takes input as encrypted data and execute job, we have observed that 23.0490 seconds was taken for running a Word Count Map Reduce job for unencrypted HDFS for size of 10MB test file while 83.2780 seconds for the encrypted HDFS with AES and 54.2360 seconds for encrypted HDFS with Real-time encryption algorithm (RTEA).

Table 1 shows the file encryption Comparison between AES and the Real Time Encryption Algorithm. The result of data uploads of plain file and encrypted file shown in following figures in terms of graphs

Table 1: Comparison between AES and Real Time Algorithm

Data (MB)	Encryption Type	Encrypted Data (MB)	Time Required for Execution (in sec)	Time required to upload to HDFS (in sec)
1 MB	AES	1.8720	25.9190	1.8110
1 MB	RTEA	1.0709	12.1510	1.5260
10 MB	AES	20.1113	297.0780	2.0210
10 MB	RTEA	10.7053	130.5600	1.8129

Table 2 shows The job execution Comparison between AES encryption and the Real Time Encryption Algorithm.

Table 2: Comparisons of Job execution of AES and Real Time Algorithm

Data (MB)	Encryption Type	Encrypted Data (MB)	Time Consumed Execution (in sec)
1 MB	AES	1.8720	26.0399
1 MB	RTEA	1.0709	21.9099
10 MB	AES	20.1113	84.1120
10 MB	RTEA	10.7053	54.1289

Figure 2 shows the comparison of file encryption time between AES algorithm and Real time algorithm. The real time algorithm requires less file encryption time as compare to AES algorithm.

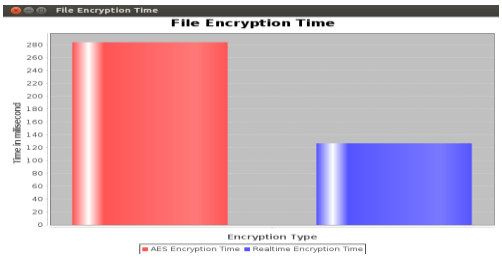

Figure 2: Comparison of File Encryption Time

Figure 3 shows the comparison between Normal File execution time, Encrypted File execution time and Real time Encrypted execution time.

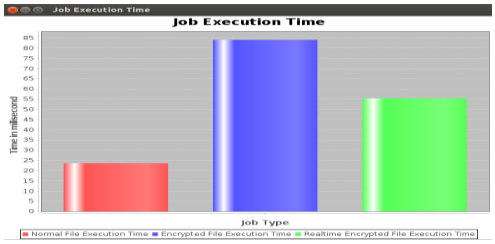

Figure 3: Comparison of Job Execution

6. Conclusion

In Hadoop data comes from various resources. Here time is important factor. Less the execution time more the benefit. 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, 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
[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 Reduce Task 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 Web GIS on Hadoop: A Case Study of Improving Small File I/O Performance on HDFS"
[13] Big Data Security: The Evolution of Hadoops Security Model Posted by Kevin T. Smith on Aug 14, 2013
[14] Seonyoung Park and Youngseok Lee, Secure Hadoop with Encrypted HDFS, Springer-Verlag Berlin Heidelberg in 2013
[15] Ke Liu and Beijing Univ OAuth Based Authentication and Authorization in Open Telco API .IEEE International Conference on Communication Systems and Network Technologies in 2012