International Journal of Engineering Research ISSN:2319-6890)(online),2347-5013(print)
Volume No.5 Issue: Special 4, pp: 790-991 20 May 2016

Preserving Privacy and Security of Big Data Using Hadoop Framework

Shushma. R[1] , Mr.Vinayaka. S.P[2]

1. Student M.tech, Dept. of Computer Science, SJM Institute of Technology Chitradurga, Karnataka.
E-mail: shushma.ramesh@gmail.com,
2. Asst.Prof, Department of Information Science and Engineering, Cambridge Institute of Technology, Bangalore.
Email: vinayaka.ise@citech.edu.in

***Abstract*—Ensuring Privacy and Security of Big Data acquiring more significance since all the new technologies started to depend on Big Data. In this paper, we mainly discussing Hadoop and difficulty in preserving the privacy as well as security of Big Data. The objective of this paper is to present a Hadoop technology that ensures the privacy and security of the data stored on the cloud system. Since Hadoop has emerged as a popular tool for Big data implementation, the paper deals with architecture of Hadoop along with its details of its components.**

Keywords-Big data; Privacy and security; Hadoop; HDFS; Map Reduce ;Cloud storage system; Encryption; Name node; Data node;

I. INTRODUCTION

Big data is the availability of a large amount of data which becomes difficult to store, process and mine using a traditional database primarily because of the data available is large, complex, unstructured and rapidly changing. Preserving Privacy and security of this information is getting more prominence and priority. As Big Data constitutes both structured and unstructured data, implementing security mechanisms on this data is a challenge [1]. Big Data is difficult to work with using most relational database management systems, desktop statistics and visualization packages since it requires massive parallel software running on tens, hundreds or even thousands of servers. The concept of Big data was first embraced by online firms like Google, eBay, Facebook, Linkdin etc.

II. CHALLENGES OF BIG DATA

Big data has three challenges, i.e. the 4 V's: Volume, Velocity, Variety ,and Veracity.

A. Volume

It is the data at rest. Usually, Terabytes and Exabyte's of existing data to process for accurate analysis. The largeness of the data available made it a challenge as it was neither possible nor efficient to handle such a large volume of data using traditional databases.

B. Velocity

It is the data in motion [2]. Usually, streaming data has a few milliseconds to a few seconds for a response. Even a few seconds is too late for some time critical applications. Maintaining such response times along with ensuring privacy and security becomes difficult to achieve. As compared to the earlier versions, where data was available in one or two forms (possibly text and tables), the current versions would mean data being available additionally in the form of pictures, videos, tweets etc.,

C. Variety

It is data in many forms. Structured, Unstructured, text and multimedia, etc., various forms of data present a challenge for achieving the privacy and security of Big Data. Different implementations of security mechanisms are necessary for different forms of data. These are going to inhibit the scalability and performance of this system.

D. Veracity

The data which is incomplete. Data is undetermined due to inconsistency, doubts, ambiguities, latency, deception and model approximations.

III. HADOOP COMPONENTS

Hadoop implements a computational paradigm named Map/Reduce, where the application is divided into many small fragments of work, each of which may be executed or re-executed on any node in the cluster. In addition, it provides a distributed file system (HDFS) that stores data on the compute nodes, providing very high aggregate bandwidth across the that node failures are automatically handled by the framework. Hadoop Common is a set of utilities that support the other Hadoop subprojects.

A. Map Reduce

MapReduce [1] is the main important part of Hadoop, It has a parallel computing framework and has responsibilities like parallelization, fault tolerance, data distribution and load balancing. The purpose of implementing MapReduce is to process and generate large data sets. The computation of MapReduce takes a set of input key/value pairs and generates a set of output key/value pairs. The computation of generating the set of output key/value pairs is divided into two functions: Map function and Reduce function..

B. Working

- The JobTracker [4] receives the MapReduce jobs submitted by the user/client. The information about

the Map, Combine and Reduce functions, input and output path is contained by the client/user program.

- The JobTracker is responsible for executing the jobs in a first come first serve order by placing the pending jobs in a queue. From the input path specified, the JobTracker determines the splits and assigns different Map and Reduce tasks to each TaskTracker. The location of the data is determined by the JobTracker by contacting the NameNode.

- The TaskTracker is preconfigured on how many slots it can support, i.e. how many tasks it can handle simultaneously. The JobTracker looks for an empty slot on the TaskTracker to assign a task. If no slot is empty it assigns the task to a TaskTracker in the same rack.

- A local environment is created by the TaskTracker when a task is assigned to it. To run the task, the TaskTracker needs the resources and it copies the required files from the distributed cache to the TaskTracker's file system. TaskTracker should report the progress about the Map and Reduce tasks back to the JobTracker

- When the Map tasks are completed by the TaskTracker it reports to the JobTracker and JobTracker specifies the selected TaskTrackers to execute the Reduce phase.

- The failure of TaskTracker when a running job crashes is prevented by the spawning of separate Java Virtual Machine (JVM) process. When the job crashes the JVM process crashes, but not the TaskTracker.

- JobTracker monitors the status of TaskTracker. The TaskTracker should send a heartbeat to the JobTracker every few minutes to update its status to the JobTracker. If the TaskTracker fails to send a heartbeat in the specified time, the JobTracker assumes it crashed and assigns the task to a different TaskTracker. The TaskTracker is responsible to report about the task to the JobTracker. If the TaskTracker reports and failure of task to the JobTracker, it is responsible for handling the failure by assigning the task to a different TaskTracker, skipping the task or by updating the TaskTracker as unreliable.

- On completion of the job the JobTracker updates its status and the client/user can pull the information from JobTracker.

C. Hadoop Distributed File System (HDFS)

Hadoop uses a block structured distributed file system for storing large volumes of data (terabytes and petabytes) called Hadoop Distributed File System [5]. All the individual files in HDFS are divided into fixed size blocks. A cluster of machines with storage capacity are used for the storage of these blocks. A DataNode is the individual machine in the cluster. The distribution of data among the Data Node is handled by the HDFS. It is responsible for adding and removing nodes from the cluster and also for DataNodes recovery. To support fault tolerance HDFS implements replication of blocks and Heartbeats. Snapshots are used for check pointing to recover from failures.

Major components of HDFS [6] include NameNode, DataNode and BackUpNode.

Name Node:

NameNode acts as the master of the system and is responsible for the management of blocks on the DataNodes. It runs on high quality software, as it's responsible for the storage of meta-data. It is the single entry point for a failure happening in Hadoop cluster. The replication factor, specified by the application is stored in the NameNode.

Data Node:

DataNodes acts as slaves and it is deployed on each machine in the cluster. DataNode stores the actual data. Its responsibility is to process the requests of client for read and write on the blocks. DataNodes hold only part of the overall data and responsible for processing the part of data it holds. The creation, deletion and replication of the blocks are handled by the DataNode when instructed by the NameNode.

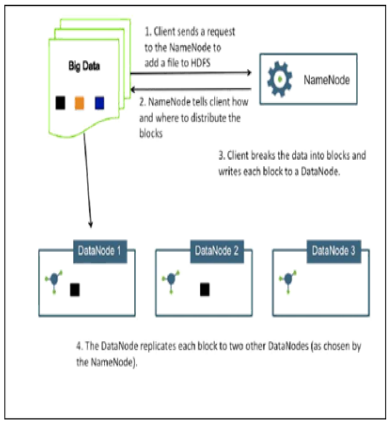

Fig.1Hadoop Architecture

ISSN:2319-6890)(online),2347-5013(print)
20 May 2016

BackUpNode:

BackUpNode is responsible for check pointing. If a failure occurs, it handles it either by rolling back or restarting using the saved checkpoint.Figure 1 clearly shows the working of the HDFS system.

IV.HADOOP KEY FEATURES

The following key features has made Hadoop very distinctive and attractive. Hadoop has several features and they are: [8].

A. Accesible:

Hadoop runs on a large clusters of commodity machines or on cloud computing services such as Amazon's Elastic Compute Cloud(EC2).

B. Robust

As Hadoop is intended to run on commodity hardware, it is architected with the assumption of frequent hardware malfunction.it can gracefully handle most such failures.

C. Scalable

Hadoop scales linearly to handle larger data by adding more nodes to the cluster.

D. Simple

Hadoop allows users to quickly write efficient parallel code. Hadoop's accessibility and simplicity give it an edge over writing and running large distributed program.

V. SECURITY MECHANISMS

Providing Privacy and security to the Big Data stored on distributed cloud storage became the critical issue of the current industry [9]. In order to overcome the delays in storing Big Data on distributed cloud storage, we use public storage as a solution. However using public storage will make the data vulnerable to transmission and storage. Therefore, there is a need for a security algorithm provides tradeoffs during time delay, security strength and storage risks with flexible key based encryption techniques.

A. Existing system

Privacy and security[10-11] of Big Data has become a important concern as every piece of technology is generating huge volumes of data to process. Researchers have worked to find a solution for this problem and few solutions to the problem have been found. Recently CSA (Cloud Security Alliance) members have done advanced research on Big Data expands through streaming cloud technology, traditional security mechanisms tailored to securing small-scale, static data on firewalled and semi isolated networks are in-adequate.

They have used cryptography and access control to protect the distributed data computations. They addressed to provide the security and privacy which are.

First one is formalizing that covers the data from cyber-attacks or losing the data. Second one is Finding the tractable solution for Analysis based on threat model. Third one is Implementing the solution using existing infra-structure. Figure 2 shows the architecture of the existing work.

B. Proposed Solution

There are privacy and security issues for the Hadoop Framework as the NameNode and the DataNode have the entire control over the data. The user has no control over the data and so there is a need for establishing a trust [12] between the user and the NameNode, i.e. we are authenticating the user, so that not everybody can access the data. To make the system more secure, we need to implement randomized encryption techniques on the data. Map Reduce does the processing and implementation of the random encryption techniques. Map Reduce is going to break the encryption/decryption process and speed up the process so as to ensure that the performance or scalability of the system is not affected. Multiple encryption techniques are implemented with an assumption that if the hacker manages to compromise certain chunks of data, he will not be capable of gaining access to all of the data for misuse.

C. Establishing Trust

In Hadoop system, the NameNode and the DataNodes handle the data for its storage and access when requested by the user. The user has no control over handling the data and so, there is a need for the establishment of trust between the NameNode and the User. Not every User should be given the privilege of accessing and storing the data. The User should authenticate himself to the NameNode before he is granted access. Hashing techniques are implemented to achieve the authentication. The hashing technique used in this system is SHA-256. The user authenticates him to NameNode by sending a hash function. NameNode then generates a hash

function and compares it with the hash function sent by the User.

D. Encryption Techniques in Proposed System

In the proposed system, multiple different encryption techniques are randomly implemented on the different nodes of the cluster. This system is implemented on the assumption that even if the hacker manages to break into a single cluster or node on the cloud and manages to break the cipher text, he will not gain access to the entire data. Without access to the entire data, the hacker will not be able to process the data for any information. In Hadoop system the entire data is broken down into chunks of 64MB and distributed on the cloud. The secure ciphers or encryption techniques that are implemented in this system are RSA, Rijndael, AES and RC6. The encryption and decryption of the data are handled by the Map Reduce [13] functions of the Hadoop system. Multiple Map and reduce functions are assigned to speed up the encryption and decryption by breaking the independent responsibilities for different Map functions and then the Reduce function brings all the encrypted/decrypted data together. By breaking the responsibilities and assigning them to Map Reduce will speed up the Encryption/Decryption process. This allows us to make the system highly secure without major fall in its performance and scalability.

VI. IMPORTANCE OF PROPOSED SYSTEM

Big data is data resource which is collected for reuse, usually without the awareness of its use during collection. This data might contain personal, individual records, government records, etc., which is not to be disclosed. While the data is stripped from the personal details of an individual or made anonymous, erosion of anonymity still occurs as the relationship between the individual pieces of information can be established to reveal the identity of the individual or the information which is not be disclosed. So, not everyone should be trusted with the data and the privacy of the data is very important.

Hadoop Framework for Big Data privacy and security was proposed as a solution to the before mentioned issues. In the proposed system we are establishing trust between the Name Node and the user, so not every user can gain access to the data. If the trust is not established between the user and the NameNode, once the hacker breaches the security of the system he can gain access to the data as the NameNode doesn't prevent a user from accessing the data. So, we are trying to establish trust between the NameNode and the user to restrict the access of the data.

VII. TEST RESULTS

The system whichcome into existence is successfully intiate faith between the user and the NameNode. Authorised users can only access to the information/data. Even if the security of the system is violated, the access of the system for the information needs authentication of the user and so unauthorised users has no privilege of system access.

The Proposed system making use of encrypted keys to authenticate the twitter network based on user. Every individual user will have a unique private key generated by using hashing SHA-256 algorithm technique. For testing, the static values generated by the keys of hashing technique are used. Using these keys we authenticate the twitter account to get the twitter streamers data by using keywords.

Hive scripts are written for the analysis, once the twitter streamed data is in Dynamo DB. To access the streamed data in Dynamo DB the private key generated by hashing technique for the individual user is required.

VIII. FUTURE WORK

The Random Encryption techniques- RSA, Rijndael, AES, RC6 are very secure and time consuming encryption standards if implemented in the existing conventional method. One of the primary issues of Big Data is velocity and so time is the biggest concern as it might lead to loss of information if the system is occupied in encrypting the information instead of collecting the information.

MapReduce is being used to speed up the encryption of information/data by dividing the process of encryption into chunks and assigning these chunks to the Map and Reduce functions. As the data/information is not encrypted using the conventional way and the usage of Map Reduce to encrypt the data has made the encryption process faster. Better/faster results are yet to be achieved to avoid data/information loss.

A Buffer system, as a solution to the problem is a work-in-progress. The buffer collects and stores the information while the system is occupied in the encryption of the data previously collected. Once the system completes the encryption, it picks data from the buffer and encrypts the data.

Once the optimal speed of encrypting data is achieved, the optimal buffer window size will be adjusted to ensure that no loss of data occurs. The four Encryption techniques will be used randomly to encrypt data on different clusters of the system.

ISSN:2319-6890)(online),2347-5013(print)
20 May 2016

IX .CONCLUSIONS

To achieve privacy and security of Big Data the distributed data's access and storage on the cloud is authenticated and secure random encryption techniques are implemented to make the system even more secure. Authentication followed by encryption techniques are implemented to make the system secure while maintaining the performance standards. Access to these huge volumes of private data might be very damaging when misused and so securing the system to the highest level is the priority.Instead of relying on expensive, proprietary hardware and different systems to store and process data, Hadoop enables distributed parallel processing of huge amounts of data across inexpensive, industry-standard servers that both store and process the data, and can scale without limits. With Hadoop, no data is too big. And in today's hyper-connected world where more and more data is being created every day.

REFERENCES

[1] J. Whitworth and S. Suthaharan. Security problems and challenges in machine learning-based Hybrid Big Data processing network systems. ACM SIGMETRICS Performance Evaluation Review. 41(4): 82-85, March 2014.

[2] G. Asharov, Y. Lindell, T. Schneider and M. Zohner. More efficient oblivious transfer and extentions for faster secure computation. CCS'13, Nov 4-8, 2013.

[3] Apache Hadoop. http://hadoop.apache.org/, 2012.

[4] P. K. Mantha, A. Luckow, S. Jha. Pilot-MapReduce: An Extensible and Flexible MapReduce Implementation for Distributed Data. Proc. of 2012 Int. Conf. on MapReduce and Its Applications, pp. 17-24.

[5] V. G. Korat, A. P. Deshmukh, K. S. Pamu. Introduction to Hadoop Distributed File System. Int. J. of Engineering Innovations and Research, 1(2): 230-236, March 2012.

[6] S. Ghemawat, H. Gobioff, and S. T. Leung: The Google File System. Proc. of 2003 ACM Symposium on Operating Systems Principles (SOSP), October 2003, Bolton Landing, NY, pp. 29-43.

[7] D. Borthakur. The Hadoop distributed file system: Architecture and design. Technical report, Apache Software Foundation, 2007.

[8] J. Shafer, S. Rixner, and A. L. Cox. The Hadoop Distributed File system: Balancing Portability and Performance. Proc. of 2010 IEEE Int. Symposium on Performance Analysis of Systems & Software (ISPASS), March 2010, White Plain, NY, pp. 122-133.

[9] N. Miloslavskaya, M. Senatorov, A. Tolstoy and S. Zapechnikov. Big Data information security maintenance. SIN'14, Sept 09-11 2014.

[10] M. Weidner, J. Dees, and P. Sanders. Fast OLAP Query Execution in Main Memory on Large Data in a Cluster. Proc. of 2013 IEEE Int. Conf. on Big Data, pp. 518-524, October 2013, Silicon Valley, CA.

[11] A. Cuzzocrea, R. Moussa, and G. Xu. OLAP*: Effectively and Efficiently Supporting Parallel OLAP over Big Data. Proc. of Model and Data Engineering, Lecture Notes in Computer Science, Volume 8216, 2013, pp. 38-49.

[12] L. Xu, X. Wu and X. Zhang. CL-PRE: a certificateless proxy re-encryption scheme for secure data sharing with public cloud. Proc. of 2012 ACM Symposium on Information, Computer and Communications Security (ASIACCS'12), May 2-4, 2012, pp. 87-88.

[13] J. Dean and S. Ghemawat. MapReduce: Simplified Data Processing on Large Clusters. Communications of the ACM, 51(1): 107-11.