Big Data Security issues and challenges in Cloud Computing Environment

Suren Kumar Sahu
Department of Computer Science and Engineering
Gandhi Engineering College, Bhubaneswar,India

Lambodar Jena
Department of Computer Science and Engineering
Gandhi Engineering College, Bhubaneswar,India

Santosh Satapathy
Department of Computer Science and Engineering
Gandhi Engineering College, Bhubaneswar,India

Abstract—Big data is a data analysis methodology implemented in recent advanced technologies and architecture. However, big data entails a huge commitment of hardware and processing resources, making adoption costs of big data technology prohibitive to small and medium sized businesses in cloud computing environment . Cloud computing is a set of services that are provided to a customer over a network on a leased basis and with the ability to scale up or down their service requirements. It advantages includes scalability, resilience, flexibility, efficiency and outsourcing non-core activities. It offers an innovative business model for organizations to adopt its services without upfront investment irrespective of the potential gains achieved from the cloud computing. The organizations are slow in accepting it due to the security issues and associated challenges. Security is one of the major issues which hamper the growth of cloud and use of big data in this environment. In this paper an overview, architecture and components of Hadoop, HDFS and Map-Reduce programming model, big data security issues and challenges related to cloud computing are presented and discussed.

***Keywords*— Hadoop, HDFS, Map-Reduce, Name node, Cloud computing**

I. INTRODUCTION

Big data is a collection of data sets so large and complex which is also exceeds the processing capacity of conventional database systems. The data is too big, moves too fast, or does not fit the structures of our current database architectures. Big Data is typically large volume of un-structured (or semi structured) and structured data that gets created from various organized and unorganized applications, activities and channels such as emails, tweeter, web logs, Facebook, etc. The main difficulties with Big Data include capture, storage, search, sharing, analysis, and visualization. The core of Big Data is Hadoop[2] which is a platform for distributing computing problems across a number of servers. It is first developed and released as open source by Yahoo, it implements the MapReduce[1] approach pioneered by Google in compiling its search indexes. Hadoop's Map Reduce involves distributing a dataset among multiple servers and operating on the data: the "map" stage. The partial results are then recombined: the "reduce" stage. To store data, Hadoop utilizes its own distributed file system, HDFS, which makes data available to multiple computing nodes. Hadoop increases the usability and performance [3, 4].HDFS has become a Very helping tool to maintain and store the complex data. Big data[5] explosion, a result not only of increasing Internet usage by people around the world, but also the connection of billions of devices to the Internet.

Fig.1. Big Data

II. BIG DATA CHARACTERISTICS

Big data can be described by the following characteristics:

- Volume
- Variety
- Velocity
- Variability
- Veracity
- Complexity

Volume:
The quantity of data that is generated is very important in this context. It is the size of the data which determines the value and potential of the data under consideration and whether it can actually be considered as Big Data or not. The name 'Big Data' itself contains a term which is related to size and hence the characteristic.
Variety :
The next aspect of Big Data is its variety. This means that the category to which Big Data belongs to is also a very essential fact that needs to be known by the data analysts. This helps the people , who are closely analyzing the data and are associated with it, to effectively use the data to their advantage and thus upholding the importance of the Big Data.
Velocity:
The term 'velocity' in the context refers to the speed of generation of data or how fast the data is generated and processed to meet the demands and the challenges which lie ahead in the path of growth and development.
Variability:
This is a factor which can be a problem for those who analyze the data. This refers to the inconsistency which can be shown by the data at times, thus hampering the process of being able to handle and manage the data effectively.
Veracity:
The quality of the data being captured can vary greatly. Accuracy of analysis depends on the veracity of the source data.
Complexity:
Data management can become a very complex process, especially when large volumes of data come from multiple sources. These data need to be linked, connected and correlated in order to be able to grasp the information that is supposed to be conveyed by these data. This situation, is therefore, termed as the 'complexity' of Big Data.

III. CLOUD COMPUTING

Cloud Computing is a term used to describe a new class of network based computing that takes place over the Internet or a model that relies on a large, centralized data center to store and process a great wealth of information. It can be defined as a collection of integrated and networked hardware, software and Internet infrastructure called a

platform i.e. using the Internet for communication and transporting hardware, software and networking services to clients. This platform hides the complexity and details of the underlying infrastructure from users and applications by providing very simple graphical interface or API (Applications Programming Interface) and also provides on-demand services that are always on, anywhere, anytime and anyplace.

Cloud computing is a way to increase the capacity or add capabilities dynamically without investing in new infrastructure, training new personnel, or licensing new software. But as more and more information are placed in the cloud, concerns begin to grow about the security of the cloud environment. Security issues in cloud computing has played a major role in slowing down its acceptance.

Fig.2. Cloud Computing

IV. Relationship between Cloud Computing and Big Data

Cloud computing and big data are conjoined. Big data provides users the ability to use commodity computing to process distributed queries across multiple datasets and return resultant sets in a timely manner. Cloud computing provides the underlying engine through the use of Hadoop, a class of distributed data-processing platforms. The use of cloud computing in big data is shown in Fig. 3. Large data sources from the cloud and Web are stored in a distributed fault-tolerant database and processed through a programming model for large datasets with a parallel distributed algorithm in a cluster. The main purpose of data visualization, as shown in Fig. 3, is to view analytical results presented visually through different graphs for decision making.

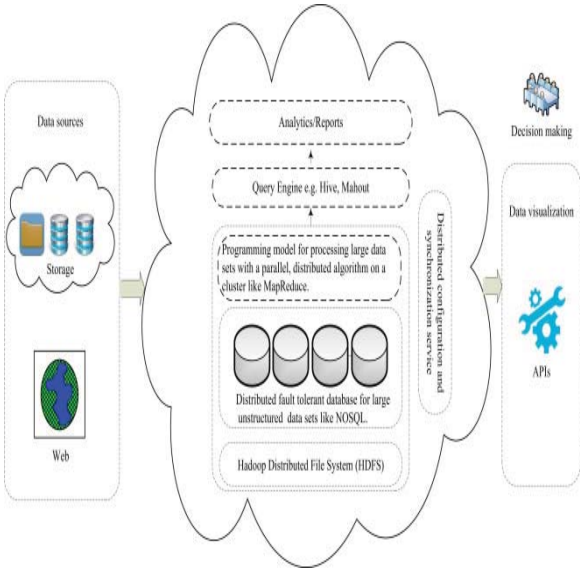

Fig. 3. Cloud computing usage in big data.

IV. HADOOP

Hadoop is an open-source software framework for storing and processing big data in a distributed fashion on large clusters of commodity hardware. Essentially, it accomplishes two tasks: massive data storage and faster processing.

Hadoop is a batch processing system for a cluster of nodes that provides the underpinnings of most BigData analytic activities because it bundle two sets of functionality most needed to deal with large unstructured datasets namely, Distributed file system and Map Reduce processing. it is a project from the Apache Software Foundation written in Java to support data intensive distributed applications. Hadoop enables applications to work with thousands of nodes and peta bytes of data.

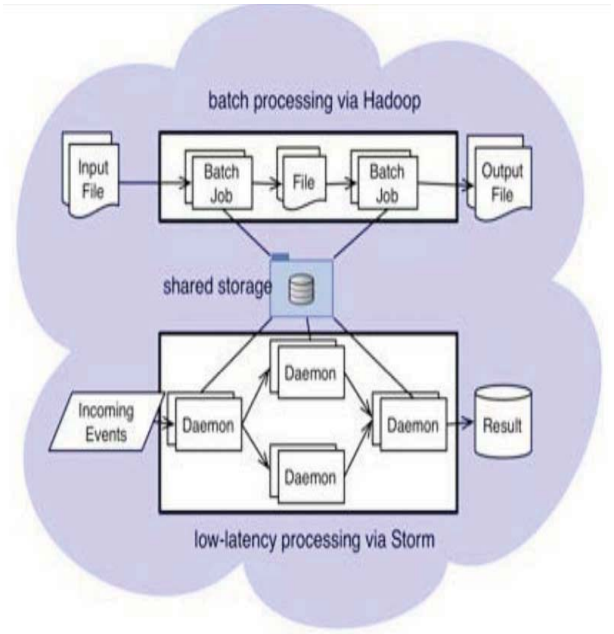

Fig. 4. Hadoop

V. ARCHITECTURE OF HADOOP

Hadoop is a Map/Reduce framework that works on HDFS or on HBase. The main idea is to decompose a job into several and identical tasks that can be executed closer to the data (on the Data Node). In addition, each task is parallelized : the Map phase. Then all these intermediate results are merged into one result : the Reduce phase. In Hadoop, The Job Tracker (a java process) is responsible for monitoring the job, managing the Map/Reduce phase, managing the retries in case of errors. The Task Trackers (Java process) are running on the different Data Nodes. Each Task Tracker executes the tasks of the job on the locally stored data.

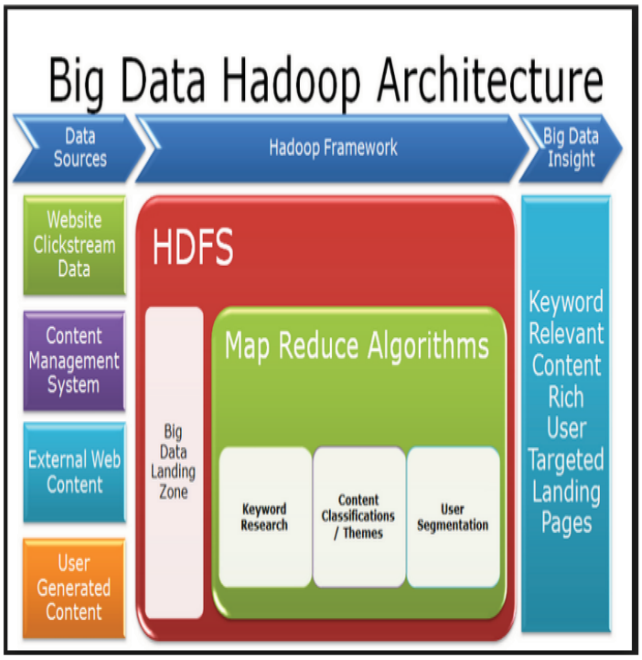

Fig. 5. Big Data Hadoop Architecture

The core of the Hadoop Cluster Architecture is given below.:

HDFS (Hadoop Distributed File System): HDFS is the basic file storage, capable of storing a large number of large files.

MapReduce: MapReduce is the programming model by which data is analyzed using the processing resources within the cluster.

Each node in a Hadoop cluster is either a master or a slave. Slave nodes are always both a Data Node and a Task Tracker. While it is possible for the same node to be both a Name Node and a JobTracker

Name Node: Manages file system metadata and access control. There is exactly one Name Node in each cluster.

Secondary Name Node: Downloads periodic checkpoints from the name Node for fault-tolerance. There is exactly one Secondary Name Node in each cluster.

*Job Tracker***:** Hands out tasks to the slave nodes. There is exactly one Job Tracker in each cluster.

*Data Node***:** Holds file system data. Each data node manages its own locally-attached storage and stores a copy of some or all blocks in the file system. There are one or more Data Nodes in each cluster.

*Task Tracker***:** Slaves that carry out map and reduce tasks. There are one or more Task Trackers in each cluster.

A. Uses of Hadoop

- Building search index at Google, Amazon, Yahoo
- Analyzing user logs, data warehousing and analytics
- Used for large scale machine learning and data mining applications
- Legacy data processing where it requires massive computational

B. HADOOP distributed file system (hdfs)

"HDFS[6] is a file system designed for storing very large files with streaming data access patterns, running on clusters on commodity hardware."

HDFS was designed keeping in mind the ideas behind Map/Reduce and Hadoop. What this implies is that it is capable of handling datasets of much bigger size than conventional file systems (even petabytes). These datasets are divided into blocks and stored across a cluster of machines which run the Map/Reduce or Hadoop jobs. This helps the Hadoop framework to partition the work in such a way that data access is local as much as possible.

A very important feature of the HDFS is its "streaming access". HDFS works on the idea that the most efficient data processing pattern is a write-once, read-many-times pattern. Once the data is generated and loaded on to the HDFS, it assumes that each analysis will involve a large proportion, if not all, of the dataset. So the time to read the whole dataset is more important than the latency in reading the first record. This has its advantages and disadvantages. One on hand, it can read bigger chunks of contiguous data locations very fast, but on the other hand, random seek turns out to be a so slow that it is highly advisable to avoid it. Hence, applications for which low-latency access to data is critical, will not perform well with HDFS.

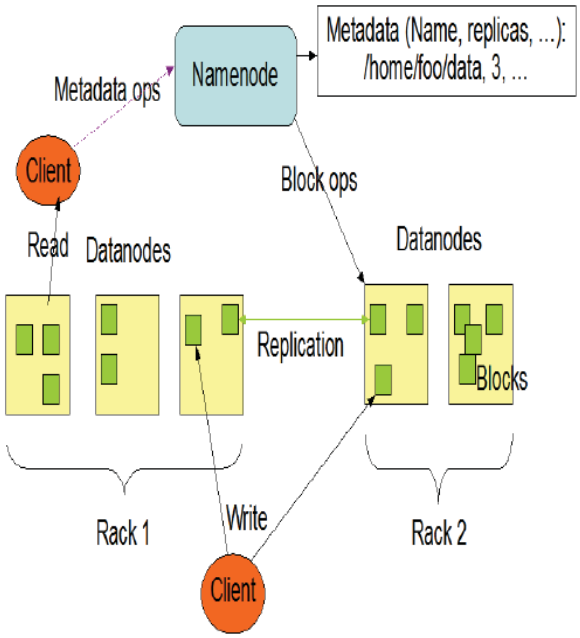

Fig. 6. Hadoop Distributed Cluster file System Architecture

C. MAP-REDUCE

Map-Reduce is a data processing or parallel programming model introduced by Google. In this model, a user specifies the computation by two functions, Map and Reduce. In the mapping phase, Map-Reduce[7] takes the input data and feeds each data element to the mapper. In the reducing phase, the reducer processes all the outputs from the mapper and arrives at a final result. In simple terms, the mapper is meant to filter and transform the input into something that the reducer can aggregate over. The underlying Map-Reduce library automatically parallelizes the computation, and handles complicated issues like data distribution, load balancing and fault tolerance. Massive input, spread across many machines, need to parallelize. Moves the data, and provides scheduling, fault tolerance. The original Map-Reduce implementation by Google, as well as its open-source counterpart, Hadoop, is aimed for

parallelizing computing in large clusters of commodity machines. Map Reduce has gained a great popularity as it gracefully and automatically achieves fault tolerance. It automatically handles the gathering of results across the multiple nodes and returns a single result or set. Map-Reduce model advantage is the easy scaling of data processing over multiple computing nodes.

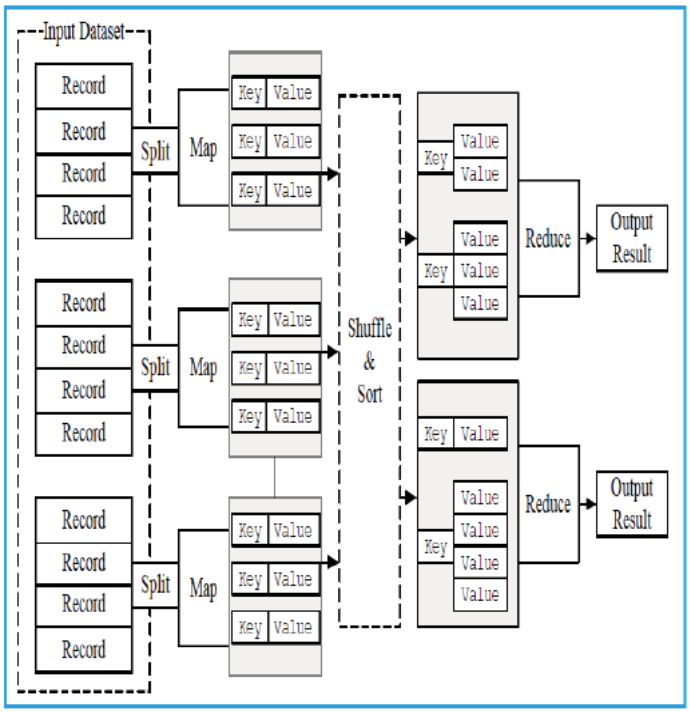

Fig. 7. Execution process of Map reduce programming Model

D. Map/Reduce Example

To better understand the Map/Reduce, lets consider an example. Given below are the map and reduce function for categorizing a set of numbers as even or odd.

```
map(String key, Integer values)
{
        //key : File Name
        //values : list of numbers
        for each v in values:
        if(v%2==0)
                EmitIntermediate("Even", v)
        else
                EmitIntermediate("Odd", v)
}
reduce(String key, Iterator values)
{
        //key: Even or Odd
        //values : Iterator over list of numbers
        //(categorized as odd or even)
        String val = ""verbatim
        while(values.hasnext())
        {
                val=val+","+values.toString()
```

```
        }
        Emit(key, val)
}
```

So given a list of numbers as 5;4;2;1;3;6;8;7; 9, the final output file will look like this -
Even 2,4,6,8
Odd 1,3,5,7,9

VI. Security Issues in Big Data

Cloud computing comes with numerous security issues[8] because it encompasses many technologies including networks, databases, operating systems, virtualization, resource scheduling, transaction management, load balancing, concurrency control and memory management.

Hence, security issues of these systems and technologies are applicable to cloud computing. For example, it is very important for the network which interconnects the systems in a cloud to be secure. In addition, resource allocation and memory management algorithms also have to be secure. The big data issues are most acutely felt in certain industries, such as telecoms, web marketing and advertising, retail and financial services, and certain government activities. The data explosion is going to make life difficult in many industries, and the companies will gain considerable advantage which is capable to adapt well and gain the ability to analyze such data explosions over those other companies.

Finally, data mining techniques can be used in the malware detection in clouds. The challenges of security in cloud computing environments can be categorized into network level, user authentication level, data level, and generic issues.

Network level:

The challenges that can be categorized under a network level deal with network protocols and network security, such as distributed nodes, distributed data, Inter node communication.

Authentication level:

The challenges that can be categorized under user authentication level deals with encryption/decryption techniques, authentication methods such as administrative rights for nodes, authentication of applications and nodes, and logging.

Data level:

The challenges that can be categorized under data level deals with data integrity and availability such as data protection and distributed data.

Generic types:

The challenges that can be categorized under general level are traditional security tools, and use of different technologies.

VII. BigData Security challenges

Secure computations in distributed programming frameworks. The first identified risk digs into the security of computational elements in frameworks such as MapReduce, with two specific security concerns outlined. First, the trustworthiness of the "mappers," which are the code that breaks data into pieces, analyzes it and outputs key-value pairs, needs to be evaluated. Second, data sanitization and de-identification capabilities need to be implemented to prevent the storage or leakage of sensitive data from the platform should be implemented through data sensitization and de-identification. Enterprises using complex tools such as MapReduce will need to use tools such as Mandatory Access Controls within SELinux and de-identifier routines to accomplish this; on the same note, enterprises should inquire as to how cloud providers are controlling and remediating this issue in their environments.

Security best practices for non relational data stores. The use of No SQL and other large-scale, non relational data stores may create new security issues due to a possible lack of capabilities in several vital areas, including any real authentication, encryption for data at rest or in transit, logging or data tagging, and classification. Organizations need to consider the use of separate application or middle ware layers to enforce authentication and data integrity. All passwords must be encrypted, and any connections to the system should ideally use Secure Sockets Layer/Transport Layer Security. Ensure logs are generated from all transactions around sensitive data as well.

Secure data storage and transactions logs. Data and transaction logs may be stored in multi-tiered storage media, but organizations need to defend against unauthorized access and ensure continuity and availability. Policy-based private key encryption can be used to ensure that only authenticated users and applications access the platform.
*Endpoint input validation/filtering***.** In a big data implementation, numerous endpoints may submit data for processing and storage. To ensure only trusted endpoints are submitting data and that false or malicious data is not submitted, organizations need to vet each endpoint connecting to the corporate network. The working group does not have a practical set of suggestions for mitigating this concern, unfortunately, aside from the recommendation to incorporate the Trusted Platform Module chips (found in many newer endpoint devices) into the validation process where possible. Host-based and mobile device security controls could potentially alleviate the risk associated with untrusted endpoints, along with strong processes around system inventory tracking and maintenance.
Real-time security monitoring. Monitoring big data platforms, as well as performing security analytics, should be done in near real time. Many traditional security information and event management platforms cannot keep pace with the large quantity (and formats) of data in use within true big data implementations. Currently, little true monitoring of Hadoop and other big data platforms exists, unless database and other front-end monitoring tools are in use.

Scalable and compos able privacy-preserving data mining and analytics. Big data implementations can lead to privacy concerns around data leakage and exposure. There are a number of security controls that can be put in place to help organizations deal with this problem, including the use of strong encryption for data at rest, access controls to data, and a separation of duty processes and controls to minimize the success of insider attacks.
*Cryptographically enforced data-centric security***.** Historically, the popular approach to data control has been to secure the systems that manage the data, as opposed to the data itself. However, those applications and platforms have proven vulnerable time and again. The use of strong cryptography to encapsulate sensitive data in cloud provider environments, as well as new and innovative algorithms that more capably allow for key management and secure key exchange, are a more reliable method for managing access to data, especially as it exists in the cloud independent of any one platform.
Granular access control. Enacting fine-grained access to big data stores such as NoSQL databases and the Hadoop Distributed File System requires the implementation of Mandatory Access Control and sound authentication. New NoSQL implementations such as Apache Accumulo can facilitate very granular access control to key-value pairs; cloud service providers should also be able to articulate the types of access controls that are in place in their environments.
*Granular audits***.** In conjunction with continuous monitoring, regular audits and analysis of log and event data can help to detect intrusions or attack attempts within the big data environment. The key control to focus on here is logging at all layers within and surrounding the big data environment.
Data provenance. Provenance in this case is focused on data validation and trustworthiness. Authentication, end-to-end data protection and fine-grained access controls can help to verify and validate provenance in big data environments; cloud service providers should have these controls in place already to address other issues.

VIII. CONCLUSION

Big data collection and processing is performed within many cloud service provider environments in some fashion. While most consumer organizations may not have big data platforms and controls in place internally, it is critical to understand the major threats and risks posed to enterprise data within the cloud environment.Considering the security issues and challenges, cloud environments can be secured for complex business operations. Using big data tools to analyze the massive amount of threat data received daily, and correlating the different components of an attack, allows a security vendor to continuously update their global threat intelligence and equates to improved threat knowledge and insight. Customers benefit through improved, faster, and broader threat protection. By reducing risk, they avoid potential recovery costs, adverse brand impacts, and legal implications. we conclude that promising progresses have been made in the area of big data and big data networking, but much remains to be done.

REFERENCES

[1] D. Borthakur, “The hadoop distributed file system: Architecture and design,” Hadoop Project Website, vol. 11, 2007.
[2] The Apache Hadoop Project. http://hadoop.apache.org/core/, 2009.
[3] Abouzeid, K. B. Pawlikowski, D. J. Abadi, A. Rasin, and A.Silberschatz. HadoopDB: An Architectural Hybrid of MapReduce and DBMS Technologies for Analytical Workloads.PVLDB,2(1):922–933, 2009.
[4] Thusoo, J. S. Sarma, N. Jain, Z. Shao, P. Chakka, S. Anthony,H. Liu, P. Wyckoff, and R. Murthy. Hive - A Warehousing SolutionOver a Map-Reduce Framework. PVLDB, 2(2):1626–1629, 2009.

[5] A, Katal, Wazid M, and Goudar R.H. "Big data: Issues, challenges, tools and Good practices.". Noida: 2013, pp. 404 – 409, 8-10 Aug.2013.
[6] K, Chitharanjan, and Kala Karun A. "A review on hadoop — HDFS infrastructure extensions.". JeJu Island: 2013, pp. 132-137, 11-12Apr. 2013.
[7] Wie, Jiang , Ravi V.T, and Agrawal G. "A Map-Reduce System with an Alternate API for Multi-core Environments.". Melbourne, VIC:2010, pp. 84-93, 17-20 May. 2010.
[8] Venkata Narasimha Inukollu , Sailaja Arsi and Srinivasa Rao Ravuri "Security issues associated with big data in cloud computing "International Journal of Network Security & Its Applications (IJNSA), Vol.6, No.3, May