# A Secure MapReduce Scheme for Big Data

G.B. Aswani Kumar
Programmer Analyst
Cognizant Technology Solutions
Chennai,T.N,India

Dr.K. Venkataramana
Associate Professor
KMM Institute of P.G Studies
Tirupati,A.P,India

***Abstract:*** **- Currently the sheer volume of data is being produced exponentially which exhibits unique characteristics as compared with traditional data. The rapid development of electronic technology and communication, which makes it hard to cost-effectively store and manage these big data. The voluminous data in terms of petabytes produced is a combination of Structured and unstructured which is difficult to process and to store. This has contributed to the big data problem faced by the industry due to the inability of conventional database systems and software tools to manage or process the big data sets within tolerable time limits. It is not possible for single or few machines to store or process this huge amount of data in a finite time period. So Cloud computing, is considered as one of most attractive solutions for big data, and provides the advantage of reduced cost through sharing of computing and storage resources. However, the growing concerns in term of the privacy of data stored in public cloud have slowed down the adoption of cloud computing for big data because sensitive information may be contained among the big data or the data owner themselves do not want any other people to scan their data. In this paper we propose a new model of secure big data sharing scheme using Hadoop clusters for storage in Hadoop Distributed File System (HDFS) in cloud data centers which uses Map Reduce Framework to process large data sets.**

***Keywords-- Big Data, Cloud computing, Security, Hadoop, Map Reduce, public key, data integrity***

## I INTRODUCTION

In this electronic age, increasing number of organizations are facing the problem of explosion of data and the size of the databases used in today's enterprises has been growing at exponential rates. Data is generated through many sources like business processes, transactions, social networking sites, web servers, etc. and remains in structured as well as unstructured form [1]. Processing or analyzing the huge amount of data or extracting meaningful information is a challenging task.

The term "Big data" is used for large data sets whose size is beyond the ability of commonly used software tools to capture, manage, and process the data within a tolerable elapsed time. Big data sizes are a constantly moving target currently ranging from a few dozen terabytes to many petabytes of data in a single data set [2]. Difficulties include capture, storage, search, sharing, securing, analytics and visualizing. big data as "datasets whose size is beyond the ability of typical database software tools to capture, store, manage, and analyze". The National Institute of Standards and Technology (NIST) [3] suggests that, "Big data is where the data volume, acquisition velocity, or data representation limits the ability to perform effective analysis using traditional relational approaches or requires the use of significant horizontal scaling for efficient processing."

Cloud computing has become a viable, mainstream solution for data processing, storage and distribution, but moving large amounts of data in and out of the cloud presented an insurmountable challenge for organizations with terabytes of digital content [4]. Cloud computing promises on demand, scalable, pay-as-you-go compute and storage capacity. Compared to an in-house datacenter, the cloud eliminates large upfront IT investments, lets businesses easily scale out infrastructure, while paying only for the capacity they use.

Big data has its origins in the cloud. Big data enables the cloud services we consume. For example, SaaS lets us collect data that was infeasible or impossible in a world of packaged software. An application can record every interaction from millions of users. This service in turn drives demand for big data technologies to store, process, and analyze these interactions and inject the value of the analysis back into the application through query and visualization [5].

The expansion of the cloud continues to drive both the creation of new big data technologies and big data adoption by making it easier and cheaper to access storage and computing resources. Companies can run their big data platforms on infrastructure provided as a service (IaaS) or consume the big data platform as a service (PaaS). Both models work in the public cloud and in on premise systems.

In cloud computing the growing concern is about the privacy and security of data stored in public cloud in which sensitive information may be contained among the big data or the data owner themselves do not want any other people to scan their data. From security and privacy to pricing models, the combination of big data and cloud computing is having a substantial impact on the nontechnical aspects as well [5]. To perform accurate analysis on Big Data which is stored in data nodes in cloud we require security and data protection controls for which we have proposed the security scheme in this paper at the level of processing of data level.

## II HADOOP

Hadoop stack will enhance Big Data architecture to extract, aggregate, load and process large volume of data in distributed manner which in turn reduce the complexity and over all turn-around time in processing large volumes of data. Apache Hadoop is an open-source software framework that supports massive data storage and processing. Hadoop enables distributed processing of large amounts of data on large clusters of commodity servers.

The Hadoop Distributed File System (HDFS) is a distributed file system providing fault tolerance and designed to run on commodity hardware. HDFS provides high throughput access to application data and is suitable for applications that have large data sets. Hadoop provides a distributed file system(HDFS) that can store data across thousands of servers, and a means of running work (Map/Reduce jobs) across those machines, running the work near the data. HDFS has master/slave architecture. Large data is automatically split into chunks which are managed by different nodes in the hadoop cluster Figure-1.

The Apache Hadoop software library is a massive computing framework consisting of several modules, including HDFS, Hadoop MapReduce, HBase, and Chukwa. Hadoop requires commodity cluster hardware to operate which requires huge investment and the alternative solution is cloud. Various Hadoop cloud solutions in which the physical cluster is located in data centers all over the world. Customers can provision their clusters with specific software images, log in remotely, load data from external sources, and run Hadoop jobs. At the same time concerns of cloud also imply for Hadoop which must be addressed.

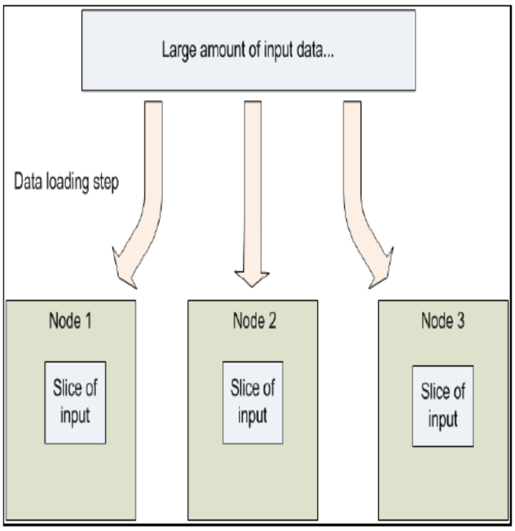

Figure 1: Data is distributed across nodes at load time

## III MAP REDUCE

MapReduce is a highly scalable programming paradigm capable of processing massive volumes of data by means of parallel execution on a large number of commodity computing nodes. It was introduced by Google [6], but today the MapReduce paradigm has been implemented in many open source projects, the most prominent being the Apache Hadoop [7]. The popularity of MapReduce can be accredited to its high scalability, fault-tolerance, simplicity and independence from the programming language or the data storage system. In the Big Data community, MapReduce has been seen as one of the key enabling approaches for meeting the continuously increasing demands on computing resources imposed by massive data sets. At the same time, MapReduce faces a number of obstacles when dealing with Big Data including the lack of a high-level language such as SQL, challenges in implementing iterative algorithms, support for iterative ad-hoc data exploration, and stream processing.

Map Reduce is a programming model for processing and generating large data sets. Users specify a map function that processes a key/value pair to generate a set of intermediate key/value pairs and a reduce function that merges all intermediate values associated with the same intermediate key [8]. "Map" step: The master node takes the input, partitions it up into smaller sub-problems, and distributes them to worker nodes. A worker node may do this again in turn, leading to a multi-level tree structure. The worker node processes the smaller problem, and passes the answer back to its master node. Map takes one pair of data with a type in one data domain, and returns a list of pairs in a different domain:

Map (k1, v1) → list (K2, v2)

"Reduce" step: The master node then collects the answers to all the sub-problems and combines them in some way to form the output – the answer to the problem it was originally trying to solve.

The Reduce function is then applied in parallel to each group, which in turn produces a collection of values in the same domain:

Reduce (K2, list (v2)) → list (v3)

Privacy is a major topic of concern whenever large amounts of information are used. Processes such as data mining and predictive analytics can discover or deduce information linkages. The data protection control can be provided through transparency and allowing input from the data provider. User input allows an individual to state their private information usage wishes [9]. This can be very difficult when dealing with a large number of mappers and reducers that Map Reduce often requires. Due to the size of the data and the complexity of the analytics performed during a MapReduce, may be difficult to control data privacy ,integrity and security. Map and Reduce steps in Map Reduce should be considered for data integrity and security.

## IV PROPOSED DATA SECURING SCHEME

The big data when distributed and mapped to data centre it should be secure to achieve accuracy in data analytics after reducing it. To achieve the accuracy, we have proposed a secure data model to be used with Map Reduce as shown in fig-2.

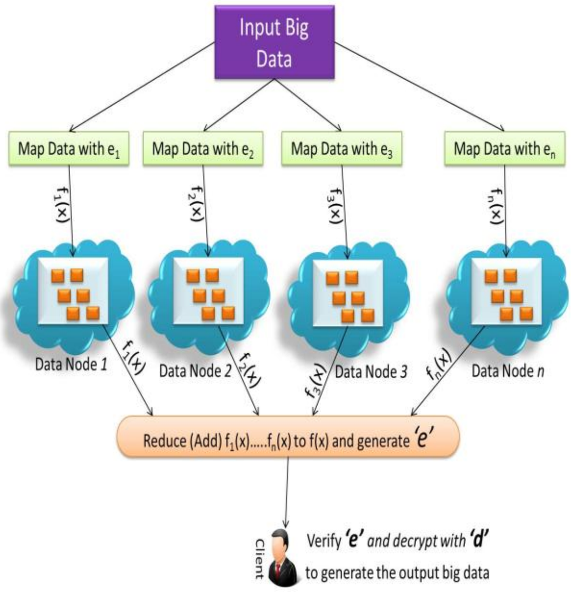

Figure-2 Proposed Scheme

In the proposed model data before distributed to clusters it is encrypted at Mapping stage and decrypted at Reduce stage. The Proposed scheme briefly stated as

1. Initially, a secure key pair <e, d> is generated by trusted authority of data centre and sent to the client through a secure channel. (e is encrypt key and d is decrypt key).
2. The data to be mapped is encrypted with public key e.
3. Polynomials f1(x), f2(x) …….. fn(x) are generated with e1, e2 ……. en as constant terms in f1(x), f2(x) …….. fn(x) (ei is the co-efficient of x0 in fi(x), i=1,2……n) where i , message digest for the polynomial is created and added to the encrypted data and stored in data centre.
4. The encrypted data from each data centre is retrieved to obtain f1(x), f2(x) …….. fn(x) and are reduced (added) to obtain f(x) and e.
5. Client verifies the obtained e with client's e. If it matches, client decrypts the encrypted data with d. If it does not match, client reports to the trusted authority of data centre.

## V SECURITY ANALYSIS OF PROPOSED SCHEME

The data in the data nodes are stored in encrypted format, the unauthorized users cannot view or modify at the time of storage or in transit which provides secrecy and privacy to the big data stored in the cloud data centres. As message digest is verified for e, the integrity of the big data can be assured. If the data is modified at the data node, the client can verify e and identifies that data is modified as 'e' does not match with original 'e' and the client notifies the trusted authority that data is modified and requests for the accurate data to perform final reduce step and achieves the original data.

## VI CONCLUSION

In this paper the main concerns about Big Data and cloud are studied. We have studied about main tool/techniques of Big data like Hadoop and Map Reduce and how they are used in Big data processing and storage. Since Big Data and Cloud are interlinked the security concerns always prevail. So we have proposed new security scheme in MapReduce which provides privacy, integrity and security to data stored in data centers in cloud and in future the model will be evaluated and results will be published in future publications.

## REFERENCES

1. Impetus white paper, March, 2011, "Planning Hadoop/NoSQL Projects for 2011" by Technologies,Available:http://www.techrepublic.c om /whitepapers/planninghadoopnosql-projects-for-2011/C2923717 ,March , 2011.
2. McKinsey Global Institute, 2011, Big Data: The next frontier for innovation, competition, and productivity, Available: www.mckinsey.com /~/media/McKinsey /dotcom /Insights 20and%20pubs/ MGI/Research/Technology %20and%20Innovation /Big%20Data/ MGI_big_data_ full_report.ashx,Aug, 2012.
3. M. Cooper and P. Mell. (2012). Tackling Big Data [Online].Available: http://csrc.nist.gov/ groups/SMA/ forum/documents/ june2012 presentations /f%csm_june2012_cooper_mell.pdf
4. Big Data Cloud, Available: http://cloud.asperasoft.com/ big-data-cloud/
5. Eli Collins," Intersection of the Cloud and Big Data", IEEE Cloud Computing 2014
6. J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," Commun ACM, 51(1), pp. 107-113, 2008.
7. Apache Hadoop, http://hadoop.apache.org.
8. Hung-Chih Yang, Ali Dasdan, Ruey-Lung Hsiao, and D.Stott Parker from Yahoo and UCLA, "Map-Reduce-Merge: Simplified Data Processing on Large Clusters",paper published in Proc. of ACM SIGMOD, pp. 1029–1040, 2007.
9. Katarina Grolinger, Michael Hayes et.al., "Challenges for MapReduce in Big Data" ,2014 IEEE 10th World Congress on Services.