Securing MapReduce Programming Paradigm in Hadoop, Cloud and Big Data Eco-system

Anitha Patil
Department of Computer Engineering
Pillai HOC College of Engineering and Technology
Rasayani, India
{panitha243@gmail.com}

ABSTRACT: *In the wake of technologies like cloud computing, virtualization and big data, MapReduce is the new programming paradigm used for processing voluminous data known as big data. MapReduce computations take place in thousands of commodity computers associated with cloud. Thus it can exploits Graphics Processing Units (GPUs) associated with cloud with its parallel processing abilities. Enterprises in the real world are shifting from traditional computing to cloud computing and traditional data mining to big data analytics. The rationale behind this is the exponential growth of data. Storing and processing such data needs big data eco-system associated with cloud computing. In this context, MapReduce programming model is supported by distributed programming frameworks like Hadoop. However, it is very challenging to secure MapReduce computations from malicious attacks. In the literature many secure cloud storage mechanisms are found. However, securing MapReduce programming paradigm in Hadoop and big data eco-system is still to be explored. In this paper, we proposed an algorithm based on differential privacy to protect big data from malicious Mapper and Reducer. We built a prototype application to demonstrate proof of the concept. The result showed the utility of the proposed approach.*

Keywords: Big Data, MapReduce Programming, Hadoop, HDFS

Received: 19 December 2019, Revised 18 March 2020, Accepted 3 May 2020

DOI: 10.6025/jcl/2020/11/3/87-96

1. Introduction

In the recent past, enterprises started giving more importance to data and data analytics for making strategic decisions. Since there is exponential growth of data in the real world applications, it became essential to have good eco-system that can be used to store and manage big data which is characterized by volume, velocity and variety. The emergence of cloud computing has enabled big data processing by providing large shared pool of computing resources in pay as you go fashion. At the same time, Hadoop kind of distributed programming framework came into existence. This framework supports storing and processing massive amount of data and exploit parallel processsing with thousands of commodity computers [1]. Since big data processing needs computational power and storage, cloud computing became handle for achieving it. With cloud computing, virtualization, big data, Hadoop and other big data platforms an eco-system is made in distributed environment.

MapReduce is the programming model used for big data processing. It is the new programming approach which contains map and reduce tasks and the work is done in distributed environment by many commodity computers. As explored in [5], Amazon is providing it own MapReduce programming model known as Elastic MapReduce which is based on Hadoop. Therefore Hadoop is the framework used for processing big data by supporting MapReduce programming. However, there are security concerns when mapper or reducer is compromised. Map and reduce tasks are vulnerable to various kinds of attacks. In this paper we considered privacy attacks on mapper and reducer.

We proposed an algorithm based on differential privacy for protecting big data from malicious mappers and reducers. The algorithm assumes the behaviour of adversaries to use a unique value for finding presence or absence of an identity in the big data. This kind of behaviour is meant for inferring sensitive information. Our algorithm adds noise to the produced output in order to defeat the purpose of privacy attack launched by adversaries. We implemented the algorithm using a prototype application that runs in Hadoop for processing big data. We used EDGAR dataset collected from [39] which has characteristics of big data. The results revealed the utility of the proposed algorithm. The remainder of the paper is structured a follows. Section 2 reviews related works throwing light on Hadoop, MapReduce, Big data and cloud computing eco-system. Section 3 presents the distributed eco-system. Section 4 provides details of the proposed algorithm. Section 5 presents experimental results. Section 6concludes the paper and provides directions for future work.

2. Related Works

This section focuses on cloud and big data eco-system in distributed environment. It reviews literature on the related topics. Wang et al. [1] studied the utility of MapReduce programming paradigm across different data centres in distributed environment. They used two big data example for processing. They include the big data associated with Large Hardon Collidor (LHC) and High Energy Physics (HEP). They extended Hadoop and named it as G-Hadoop which exploits multiple data centres. Zhao et al. [2] exploited G-Hadoop with cryptography and SSL for secure big data processing across different data centres. It also simplified job scheduling and authentication procedures.

Xavier et al. [3] investigated on different virtualization system that is used for MapReduce clusters. Virtualization is the technology used to leverage utility of computing resources. They focused on container based virtualization as it prevails in the real world MapReduce frameworks. They found high performance with Linux Container (LXC). Katal et al. [4] on the other hand studied the need for big data processing is MapReduce programming paradigm. They also specified issues and challenges related to big data processing. They described existing big data projects related to Big Science, Government, Private Sector, and international development. With respect to data analytics, they found challenges such as volume, storage, analysis, significance, best practices, technical challenges and skills needed. Karande et al. [5] studied the advantages of Hadoop cluster optimizations for big data analytics with high performance. They used Amazon S3 for storage and Elastic MapReduce (EMR) for processing big data using MapReduce programming paradigm.

Fernandez et al. [6] explored the term big data, cloud computing, distributed programming frameworks and the usage of MapReduce. They proposed a big data framework that can help in working with big data in terms of data mining and extracting business intelligence. Vavilapalli et al. [7] studied resource navigator named YARN which is associated with Apache Hadoop. It decouples programming and resource management for more effective processing of big data. In fact YARN is Hadoop's compute platform which delegates many scheduling functions in order to bring about effectiveness in the programming model. Kumar et al. [8] focused on the K-Means algorithm execution with Hadoop cluster for actually verifying and validating MapReduce functionality in distributed environment. It does mean that traditional K-Means algorithm is executed in parallel processing environment with MapReduce programming.

Grolinger et al. [9] studied various challenges that arise when big data is processed in MapRedce programming frameworks. The main challenges they identified are data storage, analytics, online processing, privacy and security issues. Kambatla et al. [10] threw light into various trends in big data analytics. The trends are examined in terms of hardware platforms to have data analytics, virtualization technologies, software stack for analytics applications, and application scope in the emerging applications. Miller et al. [11] investigate on open source frameworks for big data analytics. They found the frameworks such as Apache Hadoop, YARN, GPS, Pregel, and Apache Spark. Scala-based frameworks found by them include Spark, Kafka and Samza, and Scalation. Mythreyee et al. [12] studied the relationship between cloud and big data processing.

Win and Thien [13] investigated on the suitable big data analytics platform for mobile devices. They found that big data analytics platform for mobile devices can be built using RESTful web services. With this they proposed an analytics platform suitable for mobile devices. The platform includes mobile clients accessing cloud through web services. They built a prototype to show the utility of their framework. Cheng et al. [14] studied MapReduce programming for spatial data processing. The data is taken in the form of motion-imagery and subjected to efficient feature extraction. MapReduce programming paradigm with Hadoop is used to do it. The data is stored in Hadoop Distributed File System (HDFS). They employed an approach for this known as Cloud-Enabled WAMI Exploitation (CAWE). Pakize [15] studied Hadoop MapReduce for testing scheduling algorithms. The algorithms they explored include FIFO, fair scheduling, capacity scheduler, hybrid scheduler, self-adaptive MapReduce, delay scheduling, Maestro, Combination Re-Execution Scheduling Technology (CREST) and context-aware scheduler.

Big data and the usage of Hadoop are explored in [16] where layered architecture of big data system is proposed. Data aware caching mechanism for big data processing in Hadoop environment is investigated in [17]. The caching mechanism reduces database hits and produces more efficient query results with least latency. Huang et al. [18] focussed on the security of distributed environment where Hadoop or YARN is used for big data processing. They focussed on the security threat known as DoS and provide useful insights related to the effect of DoS attacks. Lee and Lee [19] used Hadoop MapReduce programming framework for measuring Internet traffic in scalable fashion. Zaharia et al. [20] studied the unified engine for big data processing known as Apache Spark. This framework is capable of providing big data processing capabilities in order to add big value to organizations.

Sharma and Navdeti [21] studied on the security issues and the process of securing big data in Hadoop. They focused on different security solutions provided by distributed frameworks. The security solutions include authorization, authentication, encryption of data which is at rest, encryption of data in transit, and audit trials. The frameworks against which these security solutions are examined include Hue, Zookeeper, Oozie, Pig, Hive, HBase, HDFS and MapReduce. Vasconcelos and Freitas [22] studied the performance of MapReduce with different platforms such as OpenVZ, KVM, and OpenNebula. They found that KVM performed better with respect to I/O benchmarks. Siddique et al. [23] focused on Apache Hama which is an emerging framework for big data applications. It follows a programming model known as bulk synchronous parallel programming model. Assuncao et al. [24] studied trends in big data and cloud while Pandey et al. [25] focused on big data storage and processing in Hadoop environment.

Idrissi and Aboerezq [26] explore cloud computing focusing on skyline queries for determining cloud services that meet user needs. Lemouddeen et al. [27] studied security issues in cloud computing. They found top ten threats as data breaches, data loss, insecure interfaces, malicious insiders, service hijacking, abuse, technology vulnerabilities, Denial of Service (DoS) and insufficient due diligence. Sudha and Viswanatham [28] focused on the security threats in cloud computing at network level, host level, application level and data level. Fayoumi [29] studied cloud to understand its load balancing issues and traffic and security goals. Kumar and Anand [30] investigated on workflow scheduling in cloud computing and found that security is a concern in using cloud.

Sari and Kurniawan [31] explored cloud computing infrastructure for knowledge management process. Simamora et al. [32] investigated the possible adoption of cloud computing for leveraging businesses with a case study. Kumar and Aramudhan [33] focused on trust based resource allocation in cloud computing for effective management of resources. Ghani et al. [34] focussed on green cloud computing for environmental friendly and energy efficiency. With green computing, they measured Data Centre Productivity (DCP) and Water Usage Effectiveness (WUE). Manogga et al. [35] on the other hand explored cloud to have successful e-Learning platforms that provide access to knowledge base without time and geographical restrictions.

3. Eco-system for Big Data Processing

This section provides important details related to big data, big data processing, MapReduce programming model, Hadoop and cloud computing.

3.1. Big Data and Need for It

Of late, the term big data became a hot topic in academic and research circles. The name itself indicates that it is very huge amount of data which brings about advantages and challenges. It became an essential thing as organizations are producing large volumes of data and there are technologies to store date, handle it and process such data. Mining big data can provide comprehensive business intelligence. This is the reason why most of the companies are not willing to lose its benefits. It is said that 90% of the total data in the world is generated in the last 2 years only. It reflects how exponentially data is growing.

There are many sources that are continuously producing data. They include sensor networks deployed in strategic areas, online transactions, click streams, satellites and other online applications where data is continuously produced.

Big data is the normal data that we see day to day only. But it assumes certain characteristics known as Volume, Variety and Velocity (V3). Here it is very easy to understand the Volume factor. The data is very huge and measured in petabytes or higher measurement. The big data is in different format with number of dimensions and heterogeneity. This feature is known as Variety. It is available in the form of structured data, unstructured data and semi-structured data. Big data is also associated with another feature known as velocity. It refers to the fact that data comes from different sources and in fact data is streamed continuously with certain speed. In order to process big data an environment with thousands of commodity computers is needed. This can be provided by cloud computing.

Big data needs storage of different kinds of data. Relational data is stored databases which is traditional approach. Relational Database Management System (RDBMS) is sufficient for storing relational or structured data. However, commercial applications expect high scalability and response time. There should be provision to hand unstructured and semi-structured data as well. With respect to relational model, SQL language is used to perform operations and retrieve data. Data mining and data analytics on big data can provide comprehensive business intelligence to make strategic decisions.

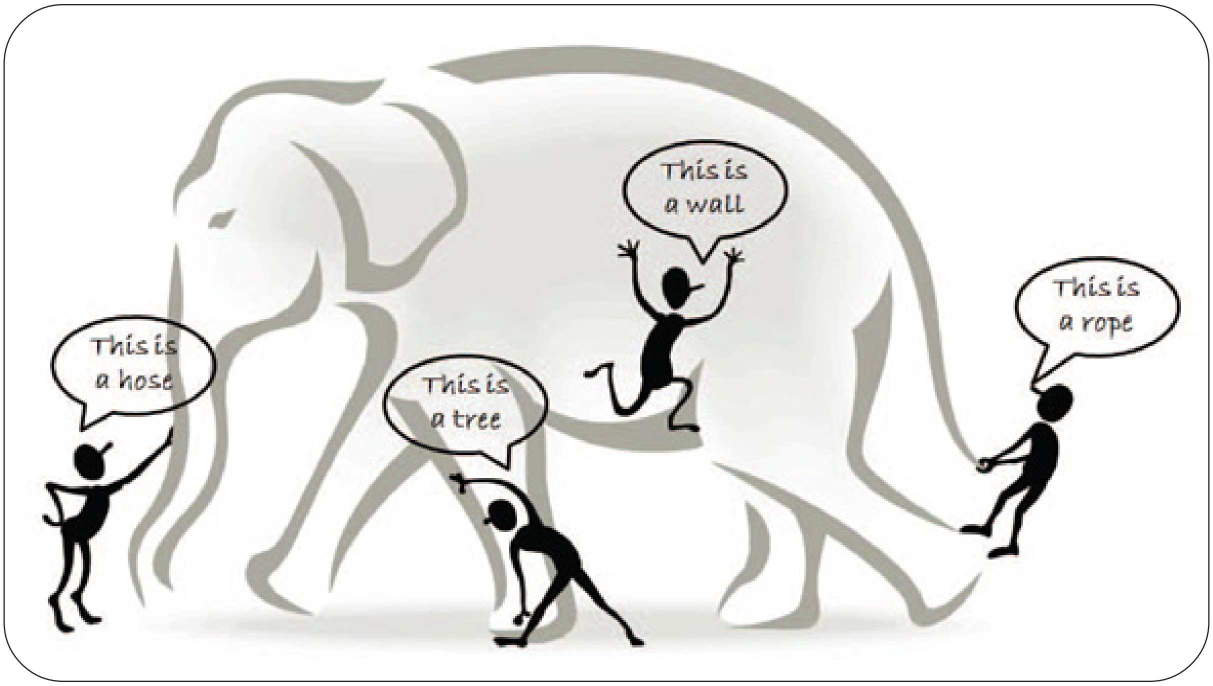

Figure 1. Shows how bias conclusions are made [36]

As illustrated in Figure 1, each individual has limited view or blind view on the thing they see and they made biased conclusions. The tail of elephant is understood as a rope. This is the problem when the whole picture is not visible or not considered. This simple example shows the importance of processing big data instead of using a part of data for performing data mining operations. When big data is considered for mining, it has complete data and processing such data needs a cloud ecosystem containing different frameworks or technologies such as Hadoop, Hadoop Distributed File System (HDFS), MapReduce programming framework, and cloud computing.

3.2. Cloud Computing

As defined by NIST, cloud computing is the technology which can provide computing resources on demand in pay per use fashion. Cloud has large shared pool of computing resources that are provisioned to be used by people or organizations and pay as they use. The cloud computing technology is based on top of virtualisation technology [27]. Cloud computing has certain essential characteristics. They are on demand self service, broad network access, resource polling, rapid elasticity and measured service. Cloud has services to be rendered such as Platform as a Service (PaaS), Infrastructure as a Service (IaaS) and Software as a Service (SaaS). It has various deployment models such as private cloud (owned by an organization and public cannot access it), public cloud (anyone can access it), community cloud (group of organizations can access it) and hybrid cloud which is made up of private cloud and public cloud [31].

3.3. Big Data Platforms

Handling big data needs special platforms. Figure 2 shows various platforms and their relationship with batch processing and real time processing. Out of them one of the most powerful platforms which are widely used for handling big data using MapReduce programming paradigm is Apache Hadoop. It is a framework for distributed programming. Reliability and high scalability are advantages of Hadoop. Dryad is another parallel programming framework. Like Hadoop it is used as a platform and infrastructure. Good programmability and high performance execution engine are advantages of Dryad. Apache Mahout is the framework which provides machine learning algorithms that can be used with big data. Baspersoft BI Suite is the business intelligence software which is cost effective and scalable. Pentaho Business Analytics is the business analytics platform which is flexible, robust, scalable and best used for knowledge discovery from big data.

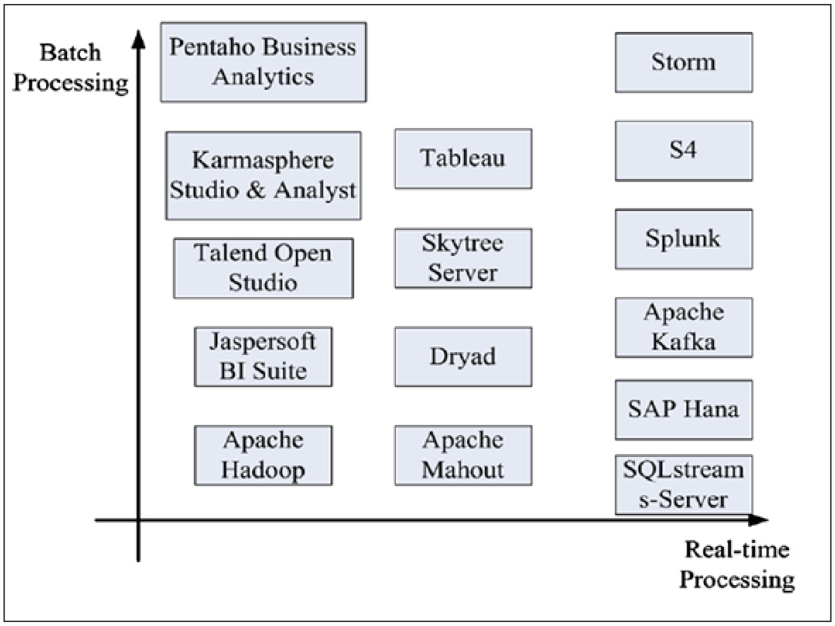

Figure 2. Big data platforms for batch and real-time processing [37]

Skytree Server supports advanced analytics with its machine learning features. It can be used to handle massive data for high speed processing. Tableau is used for business analytics and data visualization. It provides ease of use dashboards, faster, fit and smart for big data processing. Karmasphere Studio & Analyst is the big data workspace for standards-based big data analytics with collaboration. Talend Open Studio is data management and application integration platform which is easy to use with graphical environment. All the platforms aforementioned are meant for batch processing. The platforms for real time processing of big data include Storm, S4, SQL Stream Server, Splunk, Apache Kafka, and SAP Hana. Storm is the real time computation system which is highly scalable, easy to set up and fault-tolerant. S4 is meant for processing continuous unbounded data which is pluggable, scalable and fault-tolerant. SQL Stream Server on the other hand is used for sensor, telematics and M2M applications which are SQL based supporting big data stream processing. Splunk is meant for collecting and harnessing machine data which is highly scalable ease to use and fast. Apache Kafka is a public-subscribe system in distributed environment. It is best used for high throughput stream processing. SAP Hana is the platform that can be used by business in the real time. It supports real time analytics and fast in-memory computing.

3.4. Apache Hadoop

Hadoop is one of the widely used distributed programming frameworks. It is being used by Google, Facebook and other companies to deal with big data effectively. It supports a novel programming model known as MapReduce. This programming paradigm has two important tasks. They are known as Map task and Reduce task. The MapReduce framework as shown in

Figure 3 supports processing of big data with its support for parallel processing. Thousands of commodity computers associated with Hadoop are utilised for processing massive data is short span of time. When big data workload is given, Hadoop splits the data into multiple pieces and each piece of work is assigned to a worker node. The worker nodes complete their job and return the results. The whole process contains two important activities known as Map and Reduce.

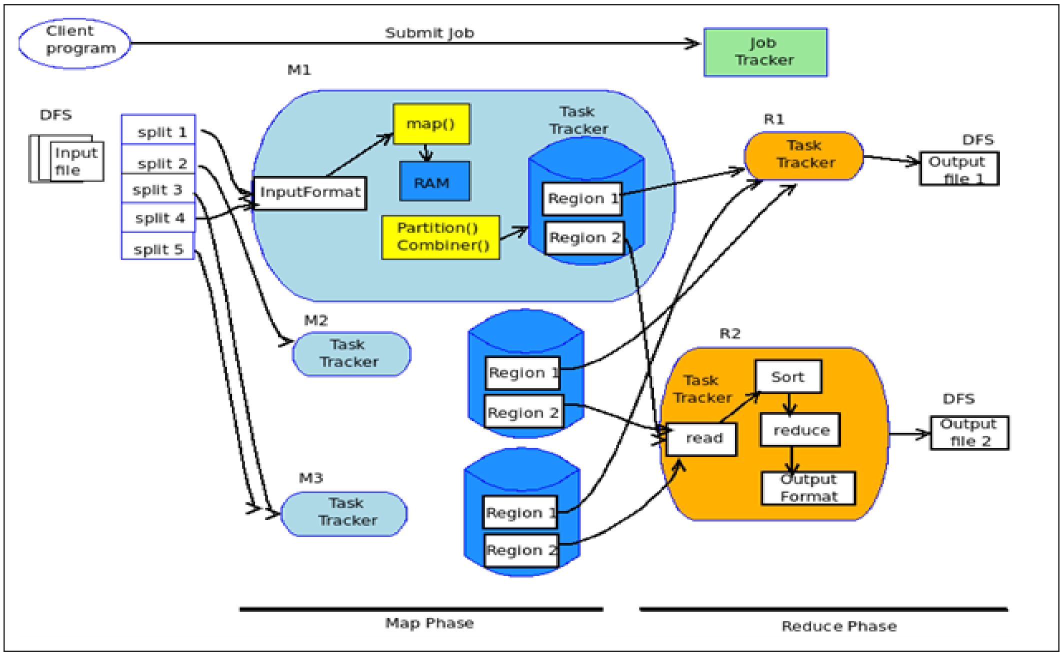

Figure 3. MapReduce programming paradigm in Hadoop

The input data is stored in HDFS which is distributed file system associated with Hadoop. HDFS can provide access to data stored in different servers which are geographically distributed. Job Tracker and Task Tracker are two important components involved in the process. Job Tracker is responsible to track the job as a whole while task tracker takes care of given task which is part of a job. In the map phase the workload which is split into multiple pieces are executed by thousands of commodity computers or worker nodes. The intermediate results are given to reduce phase. Then the reduce phase performs operations like sorting or summarizing and produce final output. The final output is also stored in HDFS. Figure 4 provides an illustration of functionality of MapReduce.

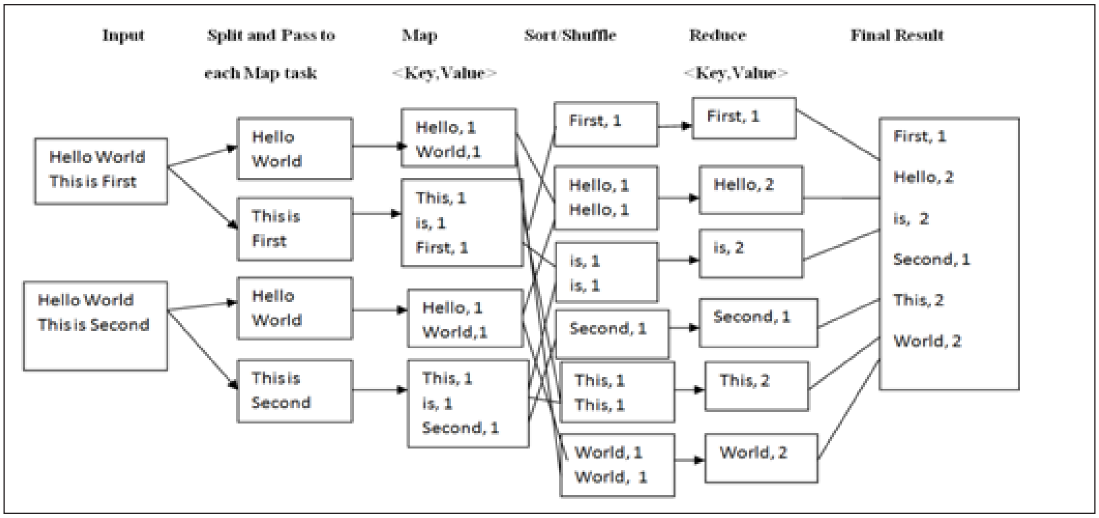

Figure 4. Illustrates MapReduce functionality with word cloud application

Word count is the typical application to demonstrate the utility of MapReduce programming paradigm. The purpose of the program is to count the occurrences of each work in a document corpus. Document corpus may be all e-Books in the world that constitutes big data. Manually doing this task takes number of years of time. When such input is given to MapReduce framework, it can easily handle it and provide output in few seconds. The rationale behind this speed is the usage of thousands of computers with parallel processing. First of all, the given document corpus is split into number of parts. Each part is passed to a map task. The map task is carried out by a commodity computer located somewhere in the world in distributed environment. The map task counts the occurrence of each word in the given portion of big data. It produces intermediate output which contains key/value pairs. Key is nothing but a word and value is its count. The Map phase in Figure 1 shows duplicate keys as well when the result of all worker nodes is observed. That is the reason, the intermediate output needs to be sorted. It is done in the sort or shuffle phase. Similar keys are grouped together. Now it is easy to move to reduce phase where the count of each word with duplicates are summed up to produce final word count. The final world count for all words in big data document corpus is saved to HDFS.

3.5. Security Issues with MapReduce Programming Paradigm

From the literature it is understood that MapReduce programming can be subjected to various attacks. One such attack considered in this paper is private related inference attack. This attack is meant for finding the presence of absence of an entity in the big data or any such kind of malicious activity to infer sensitive data. Sensitive data when inferred can cause issues to people associated with the data. Big data is collected from EDGAR web site and used for experiments.

4. Proposed Algorithm

EDGAR dataset contains unique values that are generally targeted by adversaries to leak identity information. To overcome this problem, we proposed an algorithm that is based on differential privacy mechanisms studied in [38]. In the EDGAR dataset IP address is very sensitive with which adversaries can get private information related to the company which has participated in e-filings. Our algorithm achieves privacy by incorporating noise to reducer output and protects privacy the output data from attacks.

Algorithm 1: Differential Privacy Algorithm

```
BigData_Noise(key k, values vector D)
Initialise output vector D’
For each value  in D do
  For k in 1 to n
     If value appears only once Then
      value = unique value
      Add value to D’
    end if
  end for
result = ReduceTask(D’) x (1+R(ε))
  return output
```

As shown in Algorithm 1, differential privacy is applied to big data for securing data from privacy attacks. The big data is studied a unique values are identified. The IP values are unique and sensitive in nature in the given dataset. Differential privacy is used to protect big data from malicious mapper or reducer. Noise is added to output of reducer to ensure the privacy of data. Key and value lists in the reducer are separated to avoid malicious attacks. Adversary tries to create a strange value in order to have required sensitive identity. With noise addition as part of differential privacy, the attack is prevented.

5. Experimental Results

Experiments are made with big data in the form of EDGAR dataset collected from [39]. The dataset contains information of electronic flings of companies of USA. The proposed algorithm takes the data and performs access count of every IP. In genuine case mappers and reducers produce expected output. When an attacker is intended to the presence of an IP such as 105.50.115.124, the attack is prevented by adding noise to the data as adversary injects some data for which IP is expected to be known. The results of experiments are presented in Table 1.

IP	**Genuine Count**	**Count in Presence of Attacker**
101.75.91.112	32455	32454
105.50.115.121	6453	6452
105.50.115.122	8765	8764
105.50.115.123	11345	11344
105.50.115.124	13567	13566
105.50.115.125	6543	6542
105.50.115.126	8436	8435
105.50.115.127	3480	3479
105.50.115.128	15654	15653

Table 1. Results of experiments

As shown in Table 1, the results reveal the result of differential privacy on the output data. When there is presence of attacker, the algorithm is able to provide noise added value so as to defeat the purpose of attack. The adversary fails to know the presence of absence of the IP 105.50.115.124. The differential privacy value for the IP in presence of attacker is computed as follows. The epsilon value is computed as $8.85x10^{-12}$. Then the differential privacy equation is reduced to genuine count of given *IP*+[(1+ □) + *R*]. With *R* value considered to be 2.00000000001, the final result when applied the formula is 13566. This result is highlighted in Table 1.

6. Conclusions and Future Work

In this paper we discussed about big data, cloud computing and MapReduce programming paradigm. From the literature review, we came to know security issues related to big data processing. To overcome privacy attacks on the big data being processed in MapReduce programming, we proposed an algorithm based on differential privacy. The algorithm provides mechanisms to have secured processing of big data in both Map and Reduce tasks. In other words, it adds noise in presence of an attacker to defeat the purpose of privacy attack. EDGAR dataset is used for making experiments. Hadoop environment is used for implementation of differential privacy algorithm. We built a prototype application to demonstrate proof of the concept. The empirical results reveal the utility of the proposed system. In future we intend to extend our research on rogue worker nodes and data nodes and how to handle such nodes in big data eco-system.

References

[1] Wanga, Lizhe., Taoc, Jie., Rajiv Ranjan, D., Martenc, Holger., Streit, Achim, C., Jingying Chene., Dan Chena. (2013). G-Hadoop: MapReduce across distributed data centers for data-intensive computing, *IEEE*, 2013, p. 1-14.

[2] Zhaoa, Jiaqi., Wangb, Lizhe., Taoc, Jie., Chend, Jinjun., Sunc, Weiye., Ranjane, Rajiv., Koodziejf, Joanna. (2014). Achim Streitc and Dimitrios Georgakopoulose, A security framework in G-Hadoop for big data computing across distributed Cloud data centres, *Journal of Computer and System Sciences*, 2014, p. 1-14.

[3] Xavier, Miguel G., Neves, Marcelo V.., De Rose, Cesar A. F.(2014). A Performance Comparison of Container-Based

Virtualization Systems for MapReduce Clusters, *ACM*, 2014, p1-9.

[4] Katal, Avita., Wazid, Mohammad., Goudar, R H. (2014). Big Data: Issues, Challenges, Tools and Good Practice, *IEEE*, p. 1-6.

[5] Pradhananga, Yanish., Karande, Shridevi., Chandraprakash Karande. (2016). High Performance Analytics of Bigdata with Dynamic and Optimized Hadoop Cluster, *IEEE*, p1-7.

[6] Fernandez, Alberto., Rio, Sara del., Lopez, Victoria., Bawakid, Abdullah., del Jesus, Maria J., Benítez, Jose M., Herrera., Francisco (2014). Big Data with Cloud Computing: an insight on the computing environment, MapReduce, and programming frameworks, *ACM*, p. 1-31.

[7] Vavilapallih, Vinod Kumar., Murthyh, Arun C., Douglasm, Chris., Agarwali, Sharad., Konarh, Mahadev., Evansy, Robert., Gravesy, Thomas., Lowey, Jason., Shahh, Hitesh., Sethh, Siddharth., Sahah, Bikas., Curinom, Carlo., San, Owen O'Malleyh. (2013). Apache Hadoop YARN: Yet Another Resource Negotiator, *ACM*, p. 1-16.

[8] Kumar, Amresh., Kiran, M., Mukherjee, Saikat., Ravi Prakash, G. (2013). Verification and Validation of MapReduce Program model for Parallel K-Means algorithm on Hadoop Cluster, *International Journal of Computer Applications*, 72, 2013, p 1-8.

[9] Grolinger, Katarina., Hayes, Michael., Higashino, Wilson A., L'Heureux, Alexandra., Allison, David, S., Miriam, A. M. Capretz. (2014). Challenges for MapReduce in Big Data, *IEEE*, 2014, p 1-10.

[10] Karthik Kambatla., Giorgos Kollias., Vipin Kumar., Ananth Grama. (2014). Trends in big data analytics, *IEEE*, 2014, p1-13.

[11] Miller, John A., Bowman, Casey. (2016). Vishnu Gowda Harish and Shannon Quinn, Open Source Big Data Analytics Frameworks Written in Scala, *IEEE*, 2016, p. 1-5.

[12] Poornima, Mythreyee., Purohit, S., Apoorva, D.R. (2017). *A Study on Use of Big Data in Cloud Computing Environment*, *IJARIIT*, 2017, p1-7.

[13] Win, Ngu Wah., Thein, Thandar. (2015). An Efficient Big Data Analytics Platform for Mobile Devices, *IJCSIS*, 2015, p1-5.

[14] Erkang Chenga., Liya Maa., Adam Blaissea., Erik Blaschb., Carolyn Sheaffb., Genshe Chenc., Jie Wua., Haibin Linga., Efficient Feature Extraction from Wide Area Motion Imagery by MapReduce in Hadoop, *ACM*, 2015, p1-9.

[15] Pakize, Seyed Reza (2014). A Comprehensive View of Hadoop MapReduce Scheduling Algorithms, *ijcncs*, p1-10.

[16] Harshawardhan S. Bhosale., Devendra., Gadekar, P. (2014). A Review Paper on Big Data and Hadoop, *ijsrp*, p1-7.

[17] Zhao, Yaxiong., Wu, Jie ., Liu, Cong. (2014). Dache: A Data Aware Caching for Big-Data Applications Using the MapReduce Framework, *Tsinghua Science and Technology*, p. 1-12.

[18] Huang, Jingwei., Nicol, David M., Campbell, Roy H. (2014). Denial-of-Service Threat to Hadoop/YARN Clusters with Multi-Tenancy, *IEEE*, p. 1-8.

[19] Lee, Yeonhee., Lee, Youngseok. (2013). Toward Scalable Internet Traffic Measurement and Analysis with Hadoop, *ACM*, 2013, p. 1-8.

[20] Zaharia, Matei., Xin, Reynold S., Wendell, Patrick., Das, Tathagata., Armbrust, Michael., Dave, Ankur., Xiangrui Meng., Rosen, Josh., Venkataraman, Shivaram., Franklin, Michael J., Ghodsi, Ali., Gonzalez, Joseph., Scot. (2016). Apache Spark: A Unified Engine for Big Data Processing, *ACM*, 59, 2016, p 1-10.

[21] Sharma, Priya P., Navdeti, Chandrakant P (2014). Securing Big Data Hadoop: A Review of Security Issues, Threats and Solution, *IJCSIT*, 5, p. 1-6.

[22] Vasconcelos, Pedro Roger Magalhaes., Freitas, Gisele Azevedo de Araujo. (2014). Performance Analysis of Hadoop MapReduce on an Open Nebula Cloud with KVM and OpenVZ Virtualizations, *ICITST*, 2014, p1-7.

[23] Siddique, Kamran., Akhtar, Zahid., Yoon, Edward J., Jeong, Young-Sik., Dasgupta., I., Dipankar Kim, Yangwoo. (2016). Apache Hama: An Emerging Bulk Synchronous Parallel Computing Framework for Big Data Applications, *IEEE*, 4 , 2016, p1-9.

[24] Assuncaoa, Marcos D., Calheirosb, Rodrigo N., Bianchic, Silvia., Nettoc, Marco A S., Buyyab, Rajkumar. (2014). *Big Data Computing and Clouds: Trends and Future Directions*, *ACM*, p1-44.

[25] Gupta, Arpit., Pandey, Rajiv., Verma, Komal. (2015). Analysing Distributed Big Data through Hadoop Map Reduce, *IEEE*, 129, 2015, p 1-7.

[26] Idrissi, Abdellah., Abourezq, Manar. (2015). Skyline In Cloud Computing, *Journal of Theoretical and Applied Information Technology*, 60 (3)1-12.

[27] Lemoudden, M., Ben Bouazza, N., El Ouahidi, B., Bourget, D. (2013). A Survey of Cloud Computing Security Overview of Attack Vectors and Defense Mechanisms, *Journal of Theoretical and Applied Information Technology*, 54 (2) 2013, p.1-6.

[28] Sudha, V., Madhu Viswanatham. (2013). Addressing Security and Privacy Issues in Cloud Computing, *Journal of Theoretical and Applied Information Technology*, 48 (2) p 1-13.

[29] Fayoumi, Ayman G . (2011). Performance Evaluation of a Cloud Based Load Balancer Severing Pareto Traffic, *Journal of Theoretical and Applied Information Technology*, 32 (1) p 1-7.

[30] Kumar, P., Sheila Anand. (2013). An Approach To Optimize Workflow Scheduling For Cloud Computing Environment, *Journal of Theoretical and Applied Information Technology*, 57 (3) 1-7.

[31] Ratna Sari., Yohannes Kurniawan. (2015). Cloud Computing Technology Infrastructure To Support The Knowledge Management Process, *Journal of Theoretical and Applied Information Technology*, 73 (3) 1-6.

[32] Simamora, Bachtiar H., Sarmedy, Julirzal., Kom, S. (2015). Improving Services Through Adoption Of Cloud Computing At Pt Xyz In Indonesia, *Journal of Theoretical and Applied Information Technology*, 73 (3) 1-10.

[33] Suresh Kumar, V., Aramudhan. (2014). Hybrid Optimized List Scheduling and Trust Based Resource Selection In Cloud Computing, *Journal of Theoretical and Applied Information Technology*, 69 (3), p 1-9.

[34] Ghani, Imran., Niknejad, Naghmeh., Jeong, Seung Ryul. (2015). Energy Saving in Green Cloud Computing Data Centers: A Review, *Journal of Theoretical and Applied Information Technology*, 74 (1) 1-16.

[35] Manongga, Danny., Utomo, Wiranto., Herry., Hendry. (2014). E-Learning Development as Public Infrastructure Of Cloud Computing, *Journal of Theoretical and Applied Information Technology*, 62 (1) 1-6.

[36] Wu, Xindong., Zhu, Xingquan., Wu, Gong-Qing. (2014). *Data Mining with Big Data, IEEE*, 26 (1), 97-107.

[37] Philip Chen, C. L. (2015). Chun-Yang Zhang, Data-intensive applications, challenges, techniques and technologies: A survey on Big Data, Elsevier, p. 32-44.

[38] Agrawal, R., Srikant. (2000). Privacy-Preserving Data Mining. *In:* Proceedings of the ACM SIGMOD Conference on Management of Data. Dallas, Texas, *ACM SIGMOD International Conference on Management of Data*, 2000, p 439-450.

[39] Securities and Exchange Commission. (2016). EDGAR Log File Data Set, Available: https://www.sec.gov/data/edgar-log-file-data-set. Last accessed 10 November.