Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 2, February 2014, pg.463 – 466

REVIEW ARTICLE

Privacy in Map Reduce Based Systems: A Review

Rosmy C Jose, Shaiju Paul

Department of Computer Science & Engineering, Jyothi Engineering College, Cheruthuruthy, Kerala, India
rosmycjose29@gmail.com, intouch.paulson@gmail.com

Abstract—Today, every organisation generates and adds huge amount of data to the cloud. This vast amount of data which cannot be effectively captured, processed and analysed by traditional database and search tools is called Big Data. The processing of big data is made possible by using Map Reduce, a programming model and an associated implementation, introduced by Google. MapReduce process data, which are located at different data nodes. It pushes computations to where the data resides rather than the opposite. So, Map Reduce Framework or source codes may leak sensitive data during computation process. In current implementation (Airavat) Mapper code is written by user and Reducer code is selected from a list provided by the system. If these codes are given by the system itself, usability may become low. Therefore, in the proposed system both Map and Reduce codes can be written by the user. So usability will be high. A Computation System ensures the privacy leak through storage channels (network connections, files) or privacy leak through the output of the computation is stopped. Use SELinux is used to prevent storage channel leaks. Leaks through the output of the computations are checked by using differential privacy mechanisms.

Keywords- BigData, MapReduce, Hadoop, Airavat

I. INTRODUCTION

In recent years, large amount of data is being produced. Facebook, Twitter, YouTube, stock market ex- changes, sensor devices are the main contributors of these data. These large datasets are called Bigdata. Traditional relational database can't be used to store these datasets. Also the management and manipulation of these large datasets by using conventional database tools are difficult. Cloud computing can be a possible solution as it provides rapid scalability in storage of data [1]. Cloud computing involves large-scale, distributed computations on data from multiple sources.

There are a lot of systems that can provide cloud computing services. One of the most famous implementation is MapReduce, proposed by Google in 2004. It is a programming model and an associated implementation for processing and generating large data sets [2]. It simplifies various operations on 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.

The most popular implementation of MapReduce is Hadoop [3]. It contains 2 modules namely, Hadoop Distributed File System (HDFS) and MapReduce processing. There are many advantages by using MapReduce. It is also come with many problems in relation with privacy. Careless or malicious application of MapReduce may expose sensitive data. Leakage of these sensitive data can be done either by writing it into a world-readable file, so that they can be indexed by search engines, or by outputting a specific result to signal the presence of a sensitive item in datasets of input. Traditional approaches to data privacy are based on syntactic anonymization[5] i.e., removal of personally identifiable information such as names, addresses, and Social Security numbers. Anonymization algorithms such as k- Anonymity, l-diversity, t-closeness which also remove identifiable information. Unfortunately, anonymization does not provide meaningful privacy guarantees [4].

Preserving privacy and security in cloud computing is a big challenge. I.Roy introduced a MapReduce (Hadoop) based system called Airavat [6], which provides strong security and privacy guarantees for distributed computations on sensitive data with little effort and system resources. Since users can only use reducers provided by Airavat, usability is critical.

To raise the usability, the system should provide permission to write reducer by himself. And computation system analyse the function he use in his reducer and then add enough noise to reducer's output to preserve privacy. To achieve these criteria a new system is arrived. It's architecture is based on Airavat system, with some modifications to raise usability. User of this system can write Mapper and Reducer code by himself and submit to computation system. The computation system ensures privacy.

II. MAPREDUCE

The computation takes a set of input key/value pairs, and produces a set of output key/value pairs[2]. The computations in MapReduce expressed as two functions:

Map and Reduce. Map, written by the user, takes an input pair and produces a set of intermediate key/value pairs. The MapReduce library groups together all intermediate values associated with the same intermediate key I and passes them to the Reduce function. The Reduce function, also written by the user, accepts an intermediate key I and a set of values for that key. It merges together these values to form a possibly smaller set of values. The intermediate values are supplied to the users reduce function via an iterator. This allows us to handle lists of values that are too large to fit in memory.

Programs written in this functional style are automatically parallelized and executed on a large cluster of machines. There is a run-time system to manage its execution. This gives programmers a new interface, so that they can use it without any experience with parallel and distributed systems. A typical MapReduce computation processes many terabytes of data on thousands of machines. MapReduce is being adopted by many academic researchers for data processing in different research areas, such as high- end computing, data intensive scientific analysis, large scale semantic annotation and machine learning.

III. HADOOP

Hadoop is a project from the Apache Software Foundation written in Java. It enables management of petabytes of data in thousands of machines. The inspiration comes from Google's MapReduce and Google File System papers. Hadoop's biggest contributor has been the search giant Yahoo.

Hadoop is a Map/Reduce framework that works on HDFS or on HBase. Here a job is decomposed into several and identical tasks that can be executed closer to the data (on the DataNode). In addition, each task is parallelized -the Map phase. Then all these intermediate results are merged into one result -the Reduce phase. There are 2 modules in Hadoop- JobTracker and TaskTrackers. The JobTracker (a java process) is responsible for monitoring the job, managing the Map/Reduce phase, managing the retries in case of errors. The TaskTrackers (Java process) are running on the different DataNodes. Each TaskTracker executes the tasks of the job on the locally stored data. 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. A same node can be both a Name Node and a JobTracker.

The birth of Hadoop and MapReduce gives us new privacy problems. Map and Reduce functions can be written by user. So an attacker can scan the presence or absence of an item in datasets by malicious codes or tricky codes. Also they can use storage channel like network or writing file to leak information.

IV. AIRAVAT

Airavat, a system for distributed computations which provides end-to-end confidentiality, integrity, and privacy guarantees. This can be achieved by using a combination of mandatory access control and differential privacy[6]. Airavat is based on the popular MapReduce framework, thus its interface and programming model are already familiar to developers. The three main entities in this model are (1) the data provider (2) the computation provider, and (3) the computation frame-work (Airavat). Data providers put access control labels on their data and upload them to Airavat. Computation providers write their code in the familiar MapReduce paradigm. Data providers specify the parameters of their privacy policies.

Airavat is created by modifications to the MapReduce framework, the distributed file system, and the Java virtual machine. The underlying operating system is SELinux[7]. SELinux provides security to Linux based systems. Airavat uses SELinux's mandatory access control to ensure that untrusted code does not leak information via sys- tem resources, including network connections, pipes, or other storage channels such as names of running processes. To prevent information leakage through the output of the computation, Airavat relies on a differential privacy mechanism[8].

Airavat could efficiently support distributed computations and provide privacy guarantees. But the main drawback is that users can only use reducers provided by Airavat. There are only 3 reducers that users can use. They are Sum, Count and Threshold. So Airavat is useful only when these operations are present. Therefore, usability of it is weak and useless in some case.

V. NEW SYSREM

The new system enables execution of untrusted Mapper and Reducer code on sensitive data. There are three entities in the model are (1) data provider, (2) user who write program to analyze datasets, (3) the computation framework. The aim of the system is to prevent the attack of malicious program. That is, it prevents violation of privacy policy given by data providers. Thus leakage of information about individual item in datasets is resisted. User of this system can write Mapper and Reducer code by himself and submit to computation system. The system ensures that leak through storage channels like network connections, files or leak through the output of the computation is stopped.

Here the MapReduce and Hadoop distributed file system is modified to support SELinux policy. SELinux[6] can interrupt storage channel leaks by MAC. Here SELinux is used to create a sandbox-like environment to execute untrusted code. Two domains are created for SELinux's policy. One trusted and the other untrusted. The trusted parts such as MapReduce framework and DFS execute inside the trusted domain. These processes can read and write trusted files and connect to the network. Untrusted parts, such as mapper and reducer written by a user execute in untrusted domain and have very limit permissions.

To prevent information leakage through output of computation, this system uses a differential privacy mechanism. Computation on a set of inputs is differentially private if, for any possible input item, the probability that the computation produces a given output does not depend much on whether this item is included in the dataset or not. Formally, a computation F satisfies (ε , σ)-differential privacy (where ε and are privacy parameters) if, for all datasets D and D' whose only difference is a single item which is present in D but not D', and for all outputs $S \subseteq \text{Range}(F)$,

$$\Pr[F(D) \in S] \leq \exp(\varepsilon) \times \Pr[F(D') \in S] + \sigma$$

That is, the probability of getting an output by applying a function F on a database D is less than or equal to $\exp(\varepsilon)$ times that of database D'. Here D and D' are differed by at most one element. By looking in to the outputs of a function, that is applied on D and D', one can't say whether a particular entry is present in the database or not. So user have to select privacy parameter ε to satisfies the above condition. There are many mechanisms for achieving differential privacy.

In this system, noise is added to the output of a computation $f : D \rightarrow Y^k$:

$f(x) + (R(\Delta f / \varepsilon))^k$

where $R(\Delta f)$ is a function returns a random value from $-|\Delta f|$ to $|\Delta f|$

Reducer submitted by user is compiled to Java bytecode. In this system, bytecode pattern of 5 functions are registered to pattern database. The 5 functions are sum, average, count, max and min. After mapper and reducer are submitted, the framework will check the reducer to detect what kind of function that user used To be exact, the function pattern were registered in the system. When a function is recognized, the system will apply enough noise addition method to the output of reducer. If system cannot find out the pattern, strong noise addition method will be executed.

The enough noise addition method was described above. In strong noise addition method, a new process is added to check the max, min value in intermediate (key, value) pairs and frequency of each value. Because, a value that may leak information of a specific item in datasets must be unique from other values. By looking in to this unique value attacker can get sensitive information. There- fore, replace values that appear one time by other values which frequency is bigger than two, or simply remove it. The usage of this method doesn't violate accuracy of analysis, if there are less number of unique values However, there's a possibility that it have many unique values. Therefore, the result will be extremely incorrect. It is the main disadvantage of the new system.

VI. Conclusion

Cloud computing involves large-scale, distributed computations on data from multiple sources. MapReduce is Programming model that provides distributed computing capabilities. Airavat is the first system that integrates mandatory access control with differential privacy, to secure private information of users. It provides only 3 reducers. Compared with Airavat, the new proposal's usability is better. Because a user can write mapper code and reducer code by himself to analyze data. Even though it is written by user, untrusted codes will not be accessed. The code analysis component prevents the action of malicious codes. Generally, this system ensures the accuracy in large scale computation. Also, in the given data unique value almost does not exist. However, in some case accuracy is bad. In future research, it is intend to use static code analysis tools such as Findbugs [9] to analyze mappers and reducers.

References

[1] Sanjay P. Ahuja Bryan Moore."State of Big Data Analysis in the Cloud" in Network and Communication Technologies; Vol. 2, No. 1; 22 May 2013

[2] Jeffrey Dean and Sanjay Ghemawat, MapReduce: Sim- plified Data Processing on Large Clusters,(Google, Inc).

[3] Rabi Prasad Padhy, "Big Data Processing with Hadoop- MapReduce in Cloud Systems" in International Jour- nal of Cloud Computing and Services Science (IJ- CLOSER) Vol.2, No.1, February 2013, pp. 16-27

[4] S. Hansell, AOL removes search data on vast group of web users, (New York Times, Aug 8 2006).

[5] A. Narayanan and V. Shmatikov. Robust de- anonymization of large sparse datasets,(SP, 2008).

[6] Indrajit Roy , Srinath T. V. Setty , Ann Kilzer , Vitaly Shmatikov , Emmett Witchel, "Airavat: security and privacy for MapReduce", In Proceedings of the 7th USENIX conference on Networked systems design and implementation, p.20-20, April 28-30, 2010, San Jose,California

[7] Chris Runge, SELinux: A New Approach to Secure Systems, (www.redhat.com)

[8] C. Dwork. "Differential privacy". In ICALP, 2006. [9] "Findbugs", http://findbugs.sourceforge.net/

