Ministry of Education of the Republic of Belarus
Educational Institution
Belarusian State University of Informatic and
Radioelectronics

UDC 003.26

Ben Kafo Ali Ahmed S.

Information Security in Open Distributed Info-communication Environments

ABSTRACT

for master's degree in technical sciences
Specialty 1-45 80 02 «Telecommunication systems and computer networks»

Scientific supervisor
Salomatin S. B.,
Ph.D., associate professor.

Minsk 2019

INTRODUCTION

Recently, a new wave of large-scale data processing technologies has emerged in various research and business areas, such as genomics analysis, image archiving, visualization, simulation and business intelligence. Due to the ever-increasing demands in those fields, more and more people and organizations have come to realize the increasing needs of computational resources. MapReduce is first proposed by Google in 2004 [2]. It has attracted a lot of attention ever since its development. As a compensation mechanism to make up for individual's lack of computation resources, the MapReduce framework enables huge data processing by dynamically building up the computation environment which consists of a large number of computers. Moreover, the divide-and-concur approach enables the MapReduce model to process the huge input within only a limited period of time. In addition to its benefits in data processing, MapReduce also provides clients services at low cost with flexibility. Based on its underlying cloud computing concept, MapReduce programming model is also built in a pay-as-you-go manner. Service providers can now sell their computation resources as utilities. Meanwhile, clients are also able to process their large dataset in a timely manner by just paying the "computational fees" to achieve a win-win situation, which to some extent frees the clients from the burden of IT services. Considering its benefits, MapReduce has been widely adopted by a number of large companies, such as Google, Yahoo, Amazon, Facebook and AOL[15]. An open source implementation of MapReduce called Hadoop [3], was then developed by Yahoo. The ease of application development using Hadoop further encourages wide adoption of MapReduce.

GENERAL DESCRIPTION OF WORK

Aims of research

This study is aimed at developing efficient and innovative methods that can detect the security threats exist due to MapReduce's geographically distributed computation resource. In this research, we present two lightweight methods; watermark injection and random sampling for detecting the cheating behaviors occurred on both lazy and malicious MapReduce workers. Proposed methods significantly reduce the cost of results verification overhead by eliminating the redundant computation process and are validated on typical information retrieval applications, including the word frequency count, inverted index and the page rank calculation problem. This is done in consideration of the practical value of this study,

given that MapReduce is heavily adopted by major search engine companies, such as Google.

Objectives of research

Developing methods to detect lazy and malicious servers in open distributed environments specifically, in the context of an idea crowdsourcing open distributed system and they should be:

– efficient and innovative;

– reduce the cost of the detection and results verification;

– applicable on typical applications including text processing problems such as word frequency count, inverted index and the page rank calculation problem.

Problem statement and motivation

Despite its benefits, MapReduce also brings some privacy and security concerns to its clients. Unlike the traditional paradigm, computational services provided by MapReduce applications are running on a number of distributed machines which may scatter in multiple locations. It is this open distributed environment that puts client's data into potential dangers. MapReduce clients have no control over their data once it gets fed into the distributed applications and many of whom are either facing or concerning about possible privacy threats and security risks, such as sensitive data disclosure and violations of data integrity. For instances, financial data provider would concern about potential data leakage from the service providers to their competitors. Similarly, medical clients also would be worried about intentionally data miscalculation, including both tampering the computation result and providing fake results without actual calculation.

Given that the MapReduce model has emerged only within the recent years, only a limited number of studies have been targeted on its security and privacy issues under the context of MapReduce. To protect data from potential threats, security problems of distributed systems have been studied by many researchers. Although existing approaches usually can achieve nice results with acceptable detection rates, they still have limitations of high computational costs, given that they need to either fully or partially replicate the original data, and the calculation involves in some extra works. The research goal of this thesis is to develop a new verification method for the detection of untrusted MapReduce services with lower computational cost.

Research scope

It is obviously impossible to cover all types of security threats under the MapReduce environment in one research. This thesis focuses on only the methods to ensure the MapReduce data processing services, specifically detecting cheating behaviors occurred on both lazy and malicious workers. Besides, given that MapReduce is now heavily adopted by major search engine companies in providing their information retrieval services, the scope of this thesis in particular focuses on cheater detection involved in the word count, word index and page rank tasks. The methods proposed in this thesis can be further generalized beyond these applications.

Thesis outline

Chapter 1 provides the background information of MapReduce and its security issues and describes the motivations of this study. Chapter 1 also covers a comprehensive literature review, outlining the previous and current related research works and we conclude the following:

– MapReduce does have the ability to offer its clients powerful resources for performing their computational services in a cost-effective manner. However, as an integration of many other technologies such as utility computing, distributed computing, and more recently, web services [29], MapReduce suffers from security threats which also happen to many of those technologies. In that sense, security and privacy issues therefore become a major concern that prevents more widely adoption of MapReduce;

– often operated in an open environment, MapReduce faces a number of regular communication threats. With emphasis on wide accessibility, MapReduce provides flexible and scalable computing services for its clients. However, this also leaves MapReduce vulnerable. Sensitive data, such as stock exchanges, transaction information, and military data that are transmitted between Mappers and Reduces would be easily exposed to danger. In addition, MapReduce also faces security threats of data integrity, which happens when the attacker compromises a server within the MapReduce framework. Adversary can compromise servers in the MapReduce system and tamper data during the data processing process. Besides the security risks, MapReduce may also encounter privacy threats such as sensitive data disclosure under semi-honest environments;

– in addition to all the aforementioned common risks, MapReduce users also have some unique security concerns about lazy or malicious servers involving in a MapReduce task. A lazy server can be defined as computation providers who left their computational tasks undone in economic considerations of eliminating the high

computing expenses required. On the other hand, a malicious server can be defined as service providers who intentionally tamper the computation process and return back an incorrect result. In general, two main motivations lie behind this malicious service providing, including attacks by arbitrary purpose and attacks by strategic purpose.

Chapter 2 describes the system model and cheating model of this work. In chapter 2, we define our primary goal which is to defend against two types of cheating workers, lazy and malicious servers. We assume a lazy worker probably does not start processing its assigned task in the beginning but at any point of the file and processes it sequentially, or stops anywhere before the task finishes, or directly drops the task without computation. Based on this assumption, we provide a secure scheme to prevent the computation provider from cheating the client by claiming that they have done the job that they actually did not, and also to protect from malicious behavior of MapReduce. Note that a primary requirement is that the verification overhead must take much less time than the outsourced computation.

Chapter 3 presents the actual method for cheating detection. The proposed methods are:

– Watermark Injection in the context of cheating servers' detection when performing text processing specifically, the Work Frequency and Word Indexing;

– Random Sampling using either the naïve and the in-degree weighted sampling methods to detect cheating servers when performing the PageRank computation tasks.

Chapter 4 shows the experimental evaluation conducted with simulations and describes the experimental results and analysis in details. In chapter 4, we evaluate the effectiveness of our verification scheme, a prototype of watermark injection and random sampling verification schemes is implemented to test the overall method performance. Experiments conducted in this research aims to mainly measure the detection rate under different types of cheating models.

In chapter 4, we are able to get the following results:

– the watermark injection method performs effectively on detecting the cheatings occurred on the MapReduce word count tasks. Even the lowest detection rate returned in this experiment is larger than 93%. Besides, results of this watermark injection approach under the MapReduce environment are consistent with our theoretical derivations as shown in the chapter 3;

– the random sampling approaches, both naive and weighted, perform nicely on the PageRank cheating detection task under the MapReduce environment.

CONCLUSION

This study was aimed to develop result verification schemes that can detect the security threats exist due to MapReduce's geographically distributed computation resource. In this research, we have presented two lightweight results verification schemes for detecting the cheating behaviors occurred on both lazy and malicious MapReduce workers. Proposed methods were validated on typical information retrieval applications, including the word frequency count, inverted index and the page rank calculation problem. This was done in consideration of the practical value of this study, given that MapReduce is heavily adopted by major search engine companies, such as Google. We have demonstrated that through theoretical derivations watermark injection was proved to work perfectly protecting the service integrity involved in the word count and inverted index tasks. In the same way, the theoretical validity of the random sampling method was also confirmed by its desired detection performance on page rank calculation. In order to empirically evaluate the new schemes, simulations were conducted for both methods using Hadoop's MapReduce implementation. Consistent with the previous theoretical derivations, experimental evaluations again achieved satisfying results. Both watermark injection and random sampling methods showed their effectiveness on ensuring the computational service integrity under the MapReduce environment.

Like any other researches in this field, this study also has a few limitations, centering on the lack of generalizability. First, as we have introduced in the scope section, the two schemes introduced in this study are proposed to detect service integrity violations happened in the data-intensive text processing tasks. Although this work can be extended to some other MapReduce applications, the watermark injection and random sampling schemes may not be very well generalized to other numerical computation applications, such as the ones for data mining, machine learning, and statistical analysis etc. Second, only lazy and malicious attacks have been considered in this study. Although they do cover a large proportion of the possible security risks, still there are threats in purpose from those strategic attackers. For instance, attackers who want to increase the frequencies of their proposed keywords. In that case, our watermark injection method would not work as well as it did while detecting the lazy and malicious attacks. Watermarked words that one injected are usually nonexistent words, so it would be very unlikely for the rational strategic attackers to manipulate the word count for those watermarked words. Therefore, with small watermark percentages, it would be hard to detect this kind of strategic cheating behavior using the watermark verification scheme as we proposed.

Another trivial limitation for this study is that the detection method of random sampling in PageRank calculation requires client's input with link structures. Because only in this way can the result verification process know specific inlinks of each sampled node. Clients have to parse out the link structures themselves before they can send them to MapReduce. So, this to some extent increases client's load, even though the random sampling also avoids possible replications afterwards and thus save a lot of cost for the clients. In summary, developing a generic solution for MapReduce applications remains an open question and it can be a possible direction for my future research.

Except the few limitations, this research also has several strengths, including its innovations, satisfying detection performance, and low cost and complexity. To date, there has been very limited research work on MapReduce security issues, especially on the aspect of result verification. This thesis introduced innovative methods; watermark injection and random sampling that can detect cheating services exist in a MapReduce environment. Results of these new detection schemes removed the redundant computation process as required in the replication-based methods [10]. Therefore, it significantly reduced the cost of verification overhead. I believe that the research reported is an important step towards trusted computational services and will assist in directing avenues for future research. In practical, the results of this research will hopefully further promote the secure adoption towards trusted MapReduce services and therefore help to bring profits to MapReduce service providers with increasing number of potential clients.

Finally, the main chapters and the results of this work were demonstrated and discussed in the following conferences: the 54th and the 55th international scientific conferences for graduate and undergraduate students (Minsk, 2018 - 2019)