2015 IEEE 12th International Conference on Autonomic Computing

# Integrity Protection for Big Data Processing with Dynamic Redundancy Computation

Zhimin Gao*, Nicholas DeSalvo*, Pham Dang Khoa*, Seung Hun Kim*†,
Lei Xu*, Won Woo Ro†, Rakesh M. Verma*, and Weidong Shi*
*Department of Computer Science
University of Houston, Houston, Texas
Email: {zgao5, nsdesalvo,pdkhoa, skim76, lxu13,rmverma2, wshi3}@uh.edu

†School of Electrical and Electronic Engineering,
Yonsei University, Seoul, Republic of Korea
Email: wro@yonsei.ac.kr

***Abstract*—Big data is a hot topic and has found various applications in different areas such as scientific research, financial analysis, and market studies. The development of cloud computing technology provides an adequate platform for big data applications. No matter public or private, the outsourcing and sharing characteristics of the computation model make security a big concern for big data processing in the cloud. Most existing works focus on protection of data privacy but integrity protection of the processing procedure receives little attention, which may lead the big data application user to wrong conclusions and cause serious consequences. To address this challenge, we design an integrity protection solution for big data processing in cloud environments using reputation based redundancy computation. The implementation and experiment results show that the solution only adds limited cost to achieve integrity protection and is practical for real world applications.**

## I. Introduction

Big data is high-volume, high-velocity, and high-variety information assets [1] that demand cost effective and innovative forms of information processing for enhanced insight and decision making. An enormous computation/storage resources are required to process the huge amount of data, and cloud computing provides a suitable infrastructure for these applications [2], [3]. Different frameworks are developed to simplify big data processing procedures, , e.g., Storm, Spark, and MapReduce. However, the main focuses of the designs of these frameworks are performance, scalability, and user friendliness. Security receives little attention in the design of these frameworks, and big data processing procedures are not well protected when deployed in cloud environment.

To improve security in big data processing, lots of effort has been made to protect the confidentiality of data being processed, but integrity does not receive enough attention, which is also an important aspect of secure big data processing. The main purpose of big data technology is to help the user to find the patterns behind the massive amounts of data and make better decisions. If an adversary damages the integrity of this process by modifying, deleting, or inserting into the intermediate results, it may lead to a totally wrong conclusions. For commercial usage, it means loss of profit. For defense applications, it may cause loss of lives.

To alleviate this concern, we propose an integrity protection solution for big data processing, which is based on . Our contributions in this work can be summarized as follows:

1) We propose a reputation based trust rating system for integrity protection in big data processing scenario;
2) We implement the proposed solution in MapReduce framework and use experimental results to show the effectiveness of the proposed scheme.

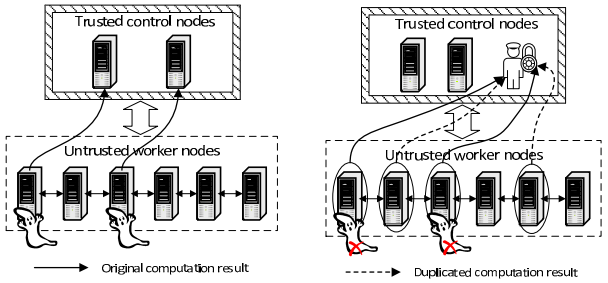
## II. Integrity Protection using Reputation based Redundance Computation

The proposed integrity protection mechanism is operated through a reputation based system. As showed in Fig. 1, all computing nodes are classified in two categories: *control nodes* and *worker nodes*. On traditional platform, an attacker (e.g., local insider or remote hacker) with system privilege can compromise big data integrity (e.g., modifying the data being processed, inserting fake data, removing data) by exploiting vulnerabilities of the application, system, and hardware data processing stack. On our enhanced platform, an integrity monitor is added to the control nodes. Worker nodes are not necessarily trusted. The integrity monitor utilizes duplicate computations to locate the suspicious worker nodes.

Once the system is initialized, each worker node is given a neutral reputation score, and these scores are maintained by a node that works in tandem with the control nodes. When the jobs are processed, the integrity monitor runs an inspecting procedure which picks certain steps in these jobs and schedules to different worker nodes for duplicated computation. This procedure then verify the results of the duplicated computation. If there is a disagreement between the result of a certain worker node and a checking node (i.e., the node which performs the duplicated computation), that means either the original node or the checking node has been compromised. The amount of duplication as well as the nodes to be duplicated are handled by the scheduler in the control node.

The operation of the reputation system itself is dynamic, i.e., reputation scores of worker nodes change during system execution. Specifically, the results of each worker node are checked and the trust score of the node becomes higher as the

978-1-4673-6971-8/15 $31.00 © 2015 IEEE
DOI 10.1109/ICAC.2015.34




(a) Big data processing framework without integrity protection.

(b) Big data processing framework with integrity protection.

Fig. 1: Big data integrity protection and threat model. The control nodes are trusted while the worker nodes may be compromised. In a nutshell, control nodes schedule some redundancy computation on different worker nodes and figure out potentially malicious worker nodes by comparing their computation results.

number of correct verification increases. If a disagreement is detected, the reputation score will decrease. For disagreement detection, the effect will also propagate to the neighbour worker nodes. Specifically, the reputation scores of worker nodes involved in related computation are lowered in a graduated fashion, i.e., worker nodes further down the computation chain will have their scores reduced less than those that are closer.

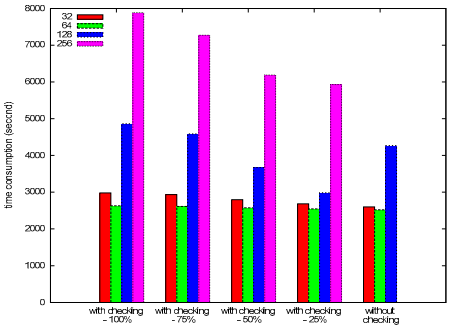
To further verify compromised worker nodes, the system may send fake jobs to suspicious nodes and their neighbors and do a fully step-by-step check to see if disagreements occurs again (if a new suspicious worker node is detected in this process, this procedure will repeat recursively). While the attacker may avoid this further verification by stopping malicious activities, it is very hard for him/her to avoid all the checks.

## III. Evaluation and Conclusion

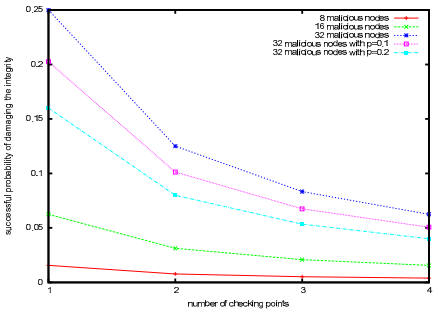
We implement the proposed solution for MapReduce and evaluate it on a cluster with 4 physical nodes, and each node has a Xeon E5603 CPU. We apply the proposed integrity protection solution to the map phase of WordCount and 39 GB text files are used.

Fig. 2a compares the performance with different ratios of duplicated computation and hardware configurations. The results are mainly affected by the number of running tasks and checksums which are sent back to the master node. Compare to the original Hadoop, the time for completing a MapReduce job is still within a reasonable range if we assign an appropriate number of tasks. For instance, in the second experiment using 4 physical nodes, the total execution time for a job with 100% duplicated tasks is only 29 seconds more than the one without duplicated computation. The extra time consists of the time for generating, transferring and comparing checksums.

We run a simulation to verify the effectiveness of the proposed integrity protection scheme, i.e., the possibility that a rogue worker node being detected. Fig. 2b shows the relationship between the number of checking points and the probability that an adversary damages the integrity without




(a) Comparison of time consumptions. Each group of columns indicates a certain ratio of duplicated nodes. The columns in the group stand for the time consumption by running different number of mappers.




(b) The simulation results of the detection probability. The probability that a malicious worker node is not detected decrease quickly when more checks are done.

Fig. 2: Evaluation result of the reputation based integrity protection solution.

being detected. The simulation results confirm that adding a small number of duplicated computation can reduce the risk of integrity breakings dramatically.

To sum up, integrity of big data processing is an important security feature. We presented a framework of integrity protection for big data processing in the cloud computing environment. The proposed solution utilizes duplicated computation to locate potential malicious worker nodes and introduces a dynamic reputation system to increase the success probability of finding out malicious nodes. Theoretical analysis and simulation results show that the proposed solution can reduce the risk of integrity breach significantly. We also implement the solution based on Hadoop and test the performance using WordCount. The experimental results illustrate the effectiveness of the proposed solution for real applications.

## References

[1] Gartner, "Gartner IT Glossary." [Online]. Available: http://www.gartner.com/it-glossary/

[2] D. Agrawal, S. Das, and A. El Abbadi, "Big data and cloud computing: Current state and future opportunities," in *Proceedings of the 14th International Conference on Extending Database Technology - EDBT/ICDT 2011*. New York, NY, USA: ACM, 2011, pp. 530–533.

[3] C. Ji, Y. Li, W. Qiu, U. Awada, and K. Li, "Big data processing in cloud computing environments," in *Proceedings of the 12th International Symposium on Pervasive Systems, Algorithms and Networks - I-SPAN 2012*. Washington, DC, USA: IEEE Computer Society, 2012, pp. 17–23.