IARJSET

ISSN (Online) 2393-8021
ISSN (Print) 2394-1588

International Advanced Research Journal in Science, Engineering and Technology

Vol. 6, Issue 12, December 2019

An Efficient Integrity Assurance Framework for Could Computing

Manal Alyoubi[1], **Jawaher Alqahtani**[2]

Information Systems Department, Faculty of Computing &Information Technology,

King Abdulaziz University, Jeddah, Saudi Arabia[1,2]

Abstract: In last decade, cloud computing has introduced noticeable changes in our personal and business lives, by offers pool of different types of re- sources can be accessing on demand. But, this technology has not reached the full deployment due to security and privacy concerns. Most of these issues derived from implementing MapReduce framework. Because processing data in MapReduce requires long running, which gives the malicious workers the enough time to discover the security vulnerabilities in it. The malicious workers can misbehave in a collusive way or in a non-collusive way. The misbehave attack can alter the program executing by gives an incorrect result computation. At more specific, they affect service integrity assurance in MapReduce. This paper surveys different improvement frameworks that addressed service integrity assurance based on MapReduce. It addressed the main techniques behind each framework with its characteristics. We find that new frameworks need some enhancements to ensure the service integrity in MapReduce.

Keywords: Cloud Computing, Integrity, Privacy, Business

I. INTRODUCTION

We are living in a world that full of air, water, and the internet. The internet has to be in all our life aspects, and that became the necessity in this century. The high growth on internet generates what is called now BIG DATA. The big data always associated with distributing computing technology, because distributing computing facilitates storing the large data in many machines. The trend in recent year towards from larger and expensive, to smaller and cheap product PCs and servers which are together to build the so-called Cloud Computing System, it has capability to providing services, storage, computation, management and so on . Moreover, It is based on the network, which has the format of service for the consumer[23].

Resources in cloud systems can be shared with a large number of users.The cloud system improves its capacity through adding more hardware to deal with the increased load effectively when the workload is growing. The services on cloud computing delivered to the users through a data center, which is on servers. The services are accessible anywhere in the world, with the cloud ap- pearing as a point of accessing for all the computing needs of consumers. So the cloud computing has to ensure the security and uprightness of information put away in the cloud framework. As the distributed computing framework has more data, which might be the private information of a client, the information must not be decimated or got. Since the information in the cloud framework might be critical for the client, the attacker may give careful consideration to getting the information [20].

With the recent propagation of cloud computing services, the MapReduce framework has turned into an incredibly common way to deal with process huge information in circulated situations. The MapReduce is a design and programming model for efficiently processing(distribution or store) large measure of data in a parallel. It gives the consistent dispersion of figuring undertakings among registering hubs, plainly to software engineers. Its present plan permits clients' information to be productively prepared in different parallel tasks, with little control over the end customers[20].

Considering its advantages, MapReduce has been adopted by some big orga- nizations, for example, Google, Yippee, and Amazon. An open source usage of MapReduce called Hadoop, created by Yippee has additionally empowered wide appropriation of MapReduce [20]. MapReduce systems have concerns about security and privacy, though. In this paper, we present some of the new frame- works that ensure MapReduce service integrity. Then we provide the comparison between all of them based on some characteristics.

The rest of the paper is organized as follows. Section 2 highlights related work. Section 3, gives background information about our survey. In Section 4, discusses some frameworks. Next, presents the comparison between framework methods & features. We continue in Section 5 with result and analysis and conclude the survey in Section 6

DOI 10.17148/IARJSET.2019.61206

II. RELATED WORK

The major concerns in distributed computing are security and privacy [4]. In security a lot of aspects must be taken in high consideration, the service integrity assurance one of these aspects that has been addressed by many research works [18]. Fernandes et al. [6] elaborated an extensive survey on security issues in cloud environments; they also demonstrated the impact of these security issues in real-life examples. According to Chen and Zhao [4] it is hard to define a unified security measures in cloud computing due to the conflict of interest. Zhou et al. [23] have given a survey on security and privacy in cloud computing, and they defined the five goals that must be achieved in cloud environment; which include the integrity. According to Schiffman et al. [14] integrity impact and effect is greater than confidentiality and availability impact.

A lot of methods and techniques have been proposed to address these issues in grid and P2P computing. Replication, sampling, verification solution and log analysis are some of these techniques [17]. Milojicic et al. [9] developed ways to measure how good or bad a peer is "reputation" in P2P systems, which help to prevent bad-behaved peers from causing the harm to a whole system. Li et al. [8] improved the authentication framework in the user side performance in grid computing, which lead to improvement in both computation and communication performance. The researchers took advantage of these techniques that were used in grid computing and P2P distributed computing, to study the possibility of implementing them in cloud computing [17]. Ahmed and Saeed [1] have focused on how to applying the three concepts volume, variety, and velocity in big data regarding security. The importance of service integrity assurance because of intensive data storage, fast access and response, and due to different party distributed computation [4]. CSA members specified three steps to achieve the top level of security and privacy. The three steps are (in order); Model a threat model that represent different scenario in bad workers .Explore the solution. Implement the solution in physical environment [20]. Zhang et al. [21] stated that "Most system security research on the cloud focuses on data-storage security and virtualization security. Little effort has been made to facilitate security and privacy protections during the computation specific to this new computing platform ".

Xiao and Xiao [19] defined cloud integrity threats in two main points: 1- Data / manipulation loss: Each user used the cloud as data storage, should be aware that his data can be exposed to loss or modification intentionally or unintentionally and also exposed to adversaries attacks. 2-Dishonest computation in remote servers: The computation processes executed in large-scale nodes which mean there no full guarantee regarding to integrity. Also, the cloud servers may act maliciously to return incorrect computing results.

One of the frameworks that cloud computing used is MapReduce. Even the MapReduce enables the users to share a data in very large scale, the user has no clue of how is data transmitted between different workers which may be untrusted [2]. Zhang et al. [22] clearly distinguished the difference between different MapRe- duce implementation private system and public system. In private system, all entities have to be under high degree of control and mostly used under single organization. The most advantage of this type is insuring the data. In contrast, in public system the entities comes from different domains, so the data security and integrity is always in questionable. Even of that, the service vendors' always preferred public system, because the geographic range of users is very big, that lead to more powerful. Roy [12] developed Airvat, which defined as "the first system that integrates mandatory access control with differential privacy, enabling many privacy-preserving MapReduce computations without the need to audit untrusted code". Yoon and Squicciarini [20] studied the possibility of improving the integrity without modifying MapReduce framework. By "correlate Hadoop logs with specific system calls, and match them against the identified invariants. Then identify whether a malicious node has subverted the functioning of the MapRe- duce operations, or otherwise altered the original workflow of the computations". This is a novel approach in detection malicious worker in MapReduce frame- work.

For integrity assurance, Bendahmane et al. [2] proposed novel mechanism based on replication-based voting method and reputation-based trust manage- ment system. It is the defeat of both collusive and non-collusive malicious workers. The accuracy rate is high together with high overhead. This paper will focus on other frameworks that interested on ensuring the service integrity in MapReduce.

III. PAGE STYLE

3.1 MapReduce Framework

MapReduce [5]is a framework for processing and generating massive data in a scalable, fault-tolerant manner. Each environment has different MapReduce interface and implementation. MapReduce is running all tasks in parallel on a cluster of hundreds or thousands of computation nodes. Two main methods in MapReduce frameworks are Map and Reduce. These methods are coordinated by the Master node which manages all jobs and resources in a cluster. While the rest of nodes called workers. A worker may be a map worker or a reduce worker. In addition to master and workers, there is a distributed file system DFS. When a user requested job computation, DFS splitting the job into sev- eral tasks which put in the task queue. The map workers or Mappers assign key/value pairs for every input data and then this pairs passes to the map func- tion which in return produced intermediate key/value pairs, to buffered it in memory temporarily. All these pairs would be fed into the reduce workers by the master. The reducer processing the intermediate results and arranged them in group based on similarity, and then produces a new set of output, which stored into the DFS. After completed all the map tasks and reduce tasks, the MapReduce call in the user program returns back to the code. The execution overview is shown in (figure 1 [5]).

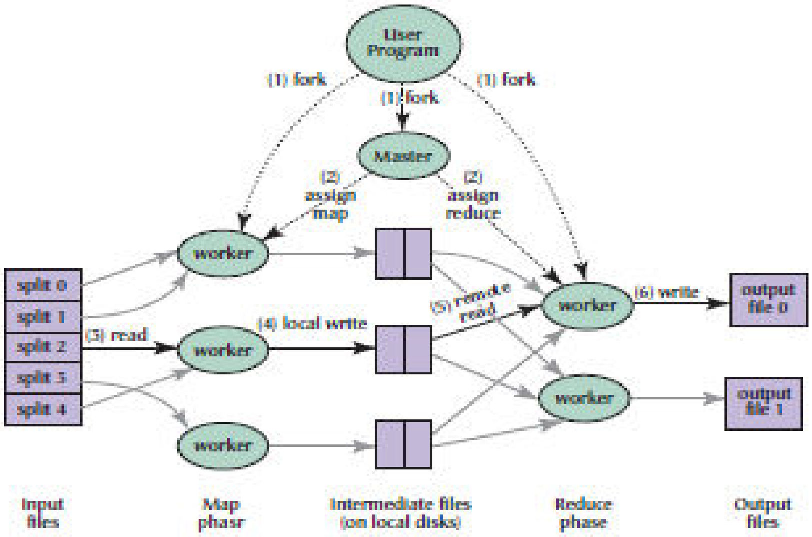

Figure 1: MapReduce execution overview[5]

Hadoop [7] is an open source, java-based programming framework which supports MapReduce implementation. It adopted master/slave model. Hadoop processing large data set among cluster of nodes. Distributed file system in Hadoop HDFS [7] is an Internet-scale file systems . By HDFS, the file systems on each local node linked together. Also, HDFS replicate the data across multiple nodes.

3.2 Service Integrity Issue on Cloud

In open systems, security assurance becomes more challenging. Integrity is one of the pillars of security issues along with authenticity, confidentiality, and availability. As this survey will focus on service integrity assurance, which is very different from data integrity assurance. To clarify the difference, this brief definition to both of them[19] :

- Data Integrity means to detect any violations at stored data in a cloud server, which includes lost or modification (Assurance the data honesty).
- Service Integrity means to detect any incorrect computing that may occur because of malware or malicious users (Assurance the program execution).

Specifically, there are two types of malware or malicious workers:

- Non-collusive malicious worker: worker misbehaves independently by re- turn the wrong result which sabotages the computation process.
- Collusive malicious worker: worker misbehaves with a prior agreement with another bad worker, to return the same wrong result, and then avoid detection by a master.

The wrong result by collusive or not-collusive workers in a cluster environment causes a critical damage. Which make imperative for researchers to work on a framework that guarantees high computation accuracy and integrity assurance. Four general strategies to check external computation integrity [19] :

1. Re-computation: the local machine re-do the computation, to compare the results. Accuracy detection rate is 100%. But it also requests high computation cost.
2. Replication: the same computation task assigns to multiple nodes, to com- pares the results. This method considers weak against collusive malicious worker.
3. Auditing: while the computation being execution; a logging component registers all important computation into a log file, which is sent to one or multiple auditors for review.
4. Trusted computing: the main concept behind this type of integrity verifi- cation is remote attestation, where the hardware is responsible for sending report to verifier whenever the software runs. To guarantee unaltered soft- ware.

IV. PROBLEM STATEMENT

Researchers in many disciplines are increasingly impressive over the use of the cloud computing in a daily base. Even of that, TechTarget survey which con- ducted from September 2012 to March 2013 shows the growth of cloud computing adoption rate is below the expectation rate. That is primarily because of security concerns associated with it [10] . As stated before, security has many aspects which include integrity. MapReduce has a critical security issue at integrity assurance as it is vulnerable against the malicious workers. That because one malicious worker can manipulate the whole computation result. This survey studies different service integrity assurance frameworks. It presents each frame- work with its architecture, characteristics, strength points, and limitations.

4.1 Service Integrity Assurance Frameworks

For integrity assurance, different researchers proposed different frameworks. All these frameworks at this study have implemented based on Hadoop. Some of the notable works, are presented below.

4.1.1 SecureMR: A Service Integrity Assurance Framework for MapReduce

Wei et al. [18] claimed that their work is the first research address service integrity issue in Hadoop. Also, secureMR prevent both replay and Denial of service (DoS) attacks. The main method is decentralized replication-based integrity verification. The two entities, a distributed file system DFS and a master are trusted. They extend the MapReduce phases by adding the verify phase, which includes workers called verifiers. The verifiers are trusted. The untrusted nodes are mapper and reducer. The system architecture design consists of five security components: secure manager, secure scheduler, secure executed, secure committer and secure verifier. The consistency verification is carried by multi- workers. The master has a secure manager and a secure scheduler for ensuring three tasks: duplication, task assignment and consisting checking. Both map- pers and reducers have secure task executor that plays a major role in defending against DoS and fake assignment. The mapper also has a secure committer for producing commitments. The verifier holds the secure verifier to complete the verification task. All these components communicate with each other under Commitment protocol and Verification protocol. The overhead caused by applying SecureMR is low, because in SecureMR the replication task is probabilistic, which means not all tasks have to duplicated. Duplicated occurs when the same task assign to different workers, to compare the two intermediate results. The inconsistent results show an attack or a faulty node. In the commitment protocol, mappers take a hash value for each partition of its intermediate result (commitments) and sending them to master. In verification protocol , reducers and master verify the consistency of the intermediate results.

4.1.2 VIAF: Verification-based Integrity Assurance Framework for MapReduce

Wang and Wei [15] proposed VIAF (Verification-based Integrity Assurance). Hadoop is modified to applying two methods. These methods are replication –based method to detect non-collusive workers, and quiz – based method to detect collusive workers. They assumed the master and DFS are trusted. Due to adding a new type of workers the verifiers, they have three types of workers: master, reducer, and verifiers. The only untrusted nodes are mappers. In this system, the master sends one task from task queue to any two workers. By applying the first stage (replication –based method), the two workers will do the calculation, so there will be two situations:

1. If the two workers have the same hash code , the master will save their result (history) to their history cache , then each worker will get credit.
2. If the two workers have different hash code, the master will reschedule the same task to different worker. In this case one of the workers must be non-collusive malicious worker.

At the second stage (quiz –based method), the master sends verifier to check the result in situation 1:

1. If the verifier gets a different result ,that mean the two workers in situation 1 are collusive malicious worker, they will be sending to blacklist. The credit will be taken off and their history cached will be rescheduled.
2. If the verifier gets the same result, that mean the two workers in situation 1 are trusted.

4.1.3 A service integrity assurance frame work for cloud computing based on MapReduce

The Authors, Ren and Tang [11] gets to this framework by improving VIAF design. Both frameworks have the same system method and the system assumption. They improve the VIAF framework by adding sub-domain management mappers as shown in (figure 2[11]). The assumption, all mappers start from an ordinary domain. But if the mapper gains enough security, it will be placed into security domain. Last domain is isolated domain. Isolated domain will have each mapper does not return the result in the time limit, each map- per does not have the same result, and each mapper has a different result than verifier. The second enhancement was by adding three models to the master node. Security Module to assign each worker with its security scale, based on its computation result. Timeout Module, to set a certain time to execute the task. Last, cache module reuse verified task and does not have to re-compute the reused task which improves the limitation in VIAF. Due to all of that, the efficiency has improved.

DOI 10.17148/IARJSET.2019.61206

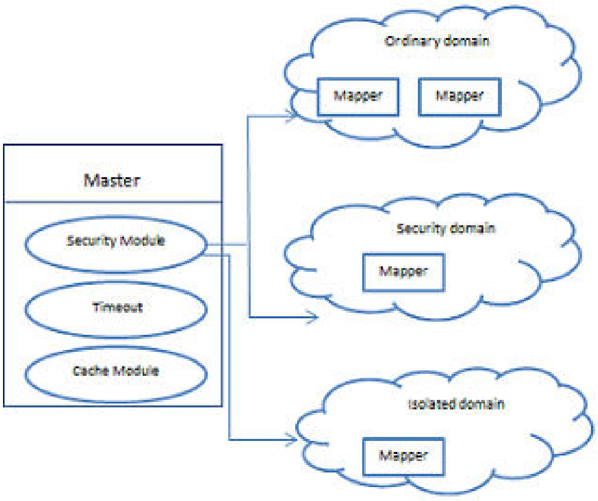

Figure 2: The Three Domains divided in the framework [11]

4.1.4 TMR: Towards a Trusted MapReduce Infrastructure

Ruan and Martin [13]proposed (TMR) Trusted MapReduce framework. It is integrated MapReduce systems with the TCG (Trusted Computing infrastruc- ture) to allow clients to make assumptions on the security possessions of a MapReduce infrastructure, contain the consistent implementation of all the se- curity protection tools. They proposed a divided and parallel verification schema to reduce latency and remove scalability limitations when using the trusted com- puting mechanisms. The task scheduler in every worker accomplishes and exe- cutes tasks organized on them. Trust collector is added to the worker to make TCG trust evidence and apply the remote attestation service. In the master, the job scheduler assign jobs to workers and gathers their results. Worker and job information are stored and managed by worker manager and job manager separately. TMR inserts the trust manager to administer the trust information of workers, and the trusted verifier, which links to the trust collectors on work- ers and implements attestations to them. The gathered security properties are stored in the trusted storage shown in (Figure 3). In TMR, every worker (Wi) is associated with its AIK (Attestation Identity Key), and the master (M) serves as the privacy for confirming and handling all these AIKs. When a new worker is added to the MapReduce infrastructure, it is first registered with the master and allocated with an AIK. As an AIK cannot be forged and can only be used inside a specific genuine TPM, in TMR, only predictable workers can link to the Master.

4.1.5 CCMR: Result Integrity Check for MapReduce Computation on Hybrid Clouds

Wang et al.[16] proposed CCMR a “novel MapReduce framework Cross Cloud MapReduce, which overlays the MapReduce computation on top of the hybrid cloud (ONE private cloud and ONE public cloud), to detect collusive and non- collusive workers ”.The CCMR architecture can be seen in (figure 3[16]). They used Random replication, Random verification, and credit-based trust manage- ment. The system assumption is the master and the verifier on a private cloud, and they are trusted .The DFS on the public cloud is also trusted. But the MapReduce entities (mapper & reducer) which on public cloud are untrusted.In map phase Integrity Check, There will be two layers check:

- First layer : replication task which done in public cloud.
- Second layer : the verification task holds on the private cloud. They used hold and test technique. As the both check are not done to every task , they add the credit-based trust management to cache each mappers credit, so when mapper’s credit get the credit threshold.

In reducing phase integrity check, the same mechanism on map phase is also applied to reduce phase, the only difference is in reduce phase they dealing with sub-task.

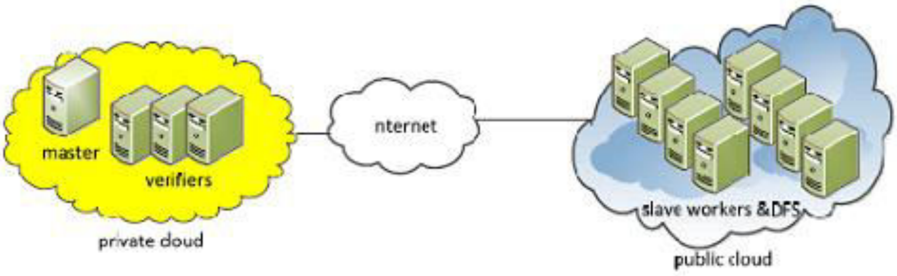

Figure 3: CCMR architecture [16]

4.1.6 IntegrityMR: Integrity Assurance Framework for Big Data Analytics and Management Applications

DOI 10.17148/IARJSET.2019.61206

Wang et al. [17] proposed integrity assurance framework for big data ana- lytics and management application. They extend the existing hybrid cloud to multiple public clouds. They adopted the same methods that used in CCMR. They also have the same assumption that CCMR have. The only difference is the IntegrityMR has more than one public cloud to choose from. The veri- fiers re-compute tasks to check inconsistency results and to detect non-collusive and collusive workers.IntegrityMR, chooses a public cloud randomly, to pick random mapper. The intermediate result from mapper sends to master. Then master sends the intermediate result to the reducer. In order to reduce the large overhead that occurs because of transmission between different public cloud, In- tegrityMR assigns the map tasks and reduce tasks to the same cloud, and the replicated map tasks and reduce tasks to the different cloud. An attacker at IntegrityMR has to face two challenges. The first challenge is the multiple pub- lic clouds design and the second challenge is the difficulty in predicting against the hold-and-test technique. The overhead in IntegrityMR is very high due to applying two replication layer and inter-cloud communication.

4.1.7 Towards Secure Tag-MapReduce Framework in Cloud

Bissiriou and Zbakh [3] proposed Tag-MapReduce framework which ensures privacy and data integrity verification in cloud computing. System methods are Tag based schema, for insuring privacy and vTPM based verification schema for integrity assurance. The architecture deployed on a hybrid cloud. The master and all nodes in private cloud are assumed to be trusted. The TPM is trusted also. While the mappers and reducers in public cloud are untrusted. Tagged-MapReduce extends MapReduce framework to prevent information leakage be- tween nodes.

Secure Tag-MapReduce applied three techniques: tag-based verification scheme, Trusted Platform Module (TPM) and Property-Based TPM Virtualization.

The tag acts as a definer for sensitive and non-sensitive data. The security tags help the programmer to use special ways of sensitivity during execution. In Tagged-MapReduce, the key-value form will be <k; v; t>, where t represents sensitive or non-sensitive data.

Regarding TPM module, they used virtual TPM instead of physical TPM to work perfectly in the hybrid cloud environment. TPM checks the integrity of MapReduce computation .

Property-Based TPM Virtualization (PB-vTPM) is a “ virtualization of hardware security modules the Trusted Platform Module (TPM) . is based on propertybased attestation mechanisms to measure the platform’s state and VM migration. PB-vTPM proposes a virtual TPM (vTPM) architecture that sup- ports various functions to measure the state of the platform, various usage strategies for cryptographic keys, and both based on a user-defined policy of the hypervisor system”.

4.2 Comparative Study

Table 1 states all frameworks were presented in this survey. It outlines for each approach the methods used for implemented, the type of cloud deployment, the assumption regarding trusted and untrusted nodes, strengths, and weakness.

Table 1: Basic information for each framework

Framework	Cloud Type	Methods	Trusted Node	Untrusted	Strengths Node	Weakness
SecureMR[18]	Public cloud	Decentralized Replication based integrity verification scheme	Master DFS Verifier	Mappers Reducer	1-Non-repudiation 2-Resilience to DoS 3-Efficiency in data processing 4-No false alarm	1-Detect only non-collusive workers. 2-Complexity is relatively high
VIAF[15]	Public cloud	1-Replication based method 2-Quiz based method	Master DFS Verifier Reducer	Mappers	100% detection rate for noncollusive worker	1-Don’t reuse verified task so they have to re-compute the reused task. 2-100% replication for each task
Service integrity assurance based on sub-domain management [11]	Public cloud	1-Replication based method 2-Quiz based method 3- Sub domain management	Master DFS Verifier Reducer	Mappers	High accuracy for detecting both collusive and con collusive workers	The reducers’ verification were ignored

TMR[13]	Public cloud	Remote attestation scheme issued by TPM	Master DFS Verifier	Mappers Reducer	Small overhead, 100% detection rate	No flexibility
CCMR[16]	Hybrid cloud	1-Random replication 2-Random verification 3-Credit based trust management	Master DFS Verifier	Mappers Reducer	1-High result integrity 2-Low job error rate	High overhead
IntegrityMR [17]	Multible Public cloud	1-Random replication 2-Random verification 3-Credit based trust management	Master DFS Verifier	Mappers Reducer	Applying layer MapReduce integrity assurance.	Issues in cross-cloud communication and DFS bottleneck
Secure Tag-MapReduce[3]	Hybrid cloud	Sensitivity Tag / Remote attestation scheme issued by vTPM	Master TPM	Mappers Reducer	100% detection rate Small overheads	NA

Table 2 presents the comparison between all frameworks. This comparison shows many features which impact the different methods for implementing integrity verification.

Table 2: Integrity Assurance features for each framework

Framework	Replication Task	Verification Task	Overhead	Quiz Threshold
SecureMR[18]	Probabilistic	Deterministic	5-12%	NO
VAIF[15]	Deterministic	Probabilistic	100-149%	Yes
Service integrity assurance based on sub-domain management [11]	Deterministic	Probabilistic	3-4%	Yes
TMR[13]	NO	Remote attestation	1.3-3.1%	No
CCMR[16]	Probabilistic	Probabilistic	19-83%	Yes
IntegrityMR[17]	Deterministic	Probabilistic	100-120%	Yes
Secure Tag- MapReduce [3]	Input data	NO	1-3%	No

V. RESULT AND ANALYSIS

Specifically, the researchers used for the integrity assurance, these methods:

1. Remote attestation scheme issued by TPM or vTPM
2. Decentralized replication based integrity verification scheme.
3. Random replication.
4. Replication-based voting method .
5. Reputation-based trust management system.

These methods fall into the three main categories: replication based verification schema, tag-based verification scheme and Trusted Platform Module (TPM) based verification . In replication-based methods, the same tasks assigned to different nodes, to compare their output result. The resulting inconsistency indicates the node is malicious. But this way caused enormous computation and high transmission overhead, which effect in system accuracy. Due to that, the framework VAIF, CCMR and IntegrityMR comes with large overhead. Se- cureMR doesn't produce much overhead, but it fails at detect collusive malicious workers. Secure Tag-MapReduce framework, the latest research address this issue. In our opinion, It is succeed at achieved the following:

1. More secure because it deployed on the hybrid cloud .
2. Caused low overhead because of remote attestation.
3. The detection rate for both collusive and non-collusive workers is 100%.

4. Applied tag-based verification scheme, Trusted Platform Module (TPM) based verification and Property-Based TPM Virtualization.

VI. CONCLUSION

Despite, the great adoption of MapReduce in open system, the researches in ser- vice integrity assurance is not fully address all security challenges. As MapReduce runs on HDFS, which is very different from traditional distributed file systems. The traditional security methods do not suitable for solving its security issues. In this paper, we have surveyed different improvement frameworks that ad- dressed service integrity assurance based on MapReduce, covering the main mechanism behind each framework, the assumption, pros, and cons. Then, we did a comparative study between them. Even though the studies are promising, they can be further improved in a workload of local computation of verification because it must be less compared to original computation. And, improvement in assumption which should not be imposed at first place to ensure the complete integrity of opens systems.

REFERENCES

[1]. Elmustafa Sayed Ali Ahmed and Rashid A Saeed. A survey of big data cloud computing security. International Journal of Computer Science and Software Engineering (IJCSSE), 3(1):78–85, 2014.

[2]. Ahmed Bendahmane, Mohammad Essaaidi, Ahmed El Moussaoui, and Ali Younes. Result verification mechanism for mapreduce computation integrity in cloud computing. In Complex Systems (ICCS), 2012 Interna- tional Conference on, pages 1–6. IEEE, 2012.

[3]. Cherif AA Bissiriou and Mostapha Zbakh. Towards secure tag-mapreduce framework in cloud. In Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Comput- ing (HPSC), and IEEE International Conference on Intelligent Data and Security (IDS), 2016 IEEE 2nd International Conference on, pages 96–104. IEEE, 2016.

[4]. Deyan Chen and Hong Zhao. Data security and privacy protection issues in cloud computing. In Computer Science and Electronics Engineering (ICCSEE), 2012 International Conference on, volume 1, pages 647–651. IEEE, 2012.

[5]. Jeffrey Dean and Sanjay Ghemawat. Mapreduce: simplified data processing on large clusters. Communications of the ACM, 51(1):107–113, 2008.

[6]. Diogo AB Fernandes, Liliana FB Soares, Jo˜ao V Gomes, M´ario M Freire, and Pedro RM In´acio. Security issues in cloud environments: a survey. International Journal of Information Security, 13(2):113–170, 2014.

[7]. Venkata Narasimha Inukollu, Sailaja Arsi, and Srinivasa Rao Ravuri. High level view of cloud security: issues and solutions. International Journal of Computer Science & Information Technology, 6(2), 2014.

[8]. Hongwei Li, Yuanshun Dai, and Bo Yang. Identity-based cryptography for cloud security. IACR Cryptology ePrint Archive, 2011:169, 2011.

[9]. Dejan S Milojicic, Vana Kalogeraki, Rajan Lukose, Kiran Nagaraja, Jim Pruyne, Bruno Richard, Sami Rollins, and Zhichen Xu. Peer-to-peer com- puting, 2002.

[10]. Safiya Okai, Mueen Uddin, Amad Arshad, Raed Alsaqour, and Asadullah Shah. Cloud computing adoption model for universities to increase ict proficiency. SAGE Open, 4(3):2158244014546461, 2014.

[11]. Yulong Ren and Wen Tang. A service integrity assurance framework for cloud computing based on mapreduce. In 2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems, volume 1, pages 240–244. IEEE, 2012.

[12]. Indrajit Roy, Srinath TV Setty, Ann Kilzer, Vitaly Shmatikov, and Emmett Witchel. Airavat: Security and privacy for mapreduce. In NSDI, volume 10, pages 297–312, 2010.

[13]. Anbang Ruan and Andrew Martin. Tmr: Towards a trusted mapreduce infrastructure. In 2012 IEEE Eighth World Congress on Services, pages 141–148. IEEE, 2012.

[14]. Mike Schiffman, A Wright, D Ahmad, and G Eschelbeck. The common vulnerability scoring system. National Infrastructure Advisory Council, Vulnerability Disclosure Working Group, Vulnerability Scoring Subgroup, 2004.

[15]. Yongzhi Wang and Jinpeng Wei. Viaf: Verification-based integrity assur- ance framework for mapreduce. In Cloud Computing (CLOUD), 2011 IEEE International Conference on, pages 300–307. IEEE, 2011.

[16]. Yongzhi Wang, Jinpeng Wei, and Mudhakar Srivatsa. Result integrity check for mapreduce computation on hybrid clouds. In 2013 IEEE Sixth International Conference on Cloud Computing, pages 847–854. IEEE, 2013.

[17]. Yongzhi Wang, Jinpeng Wei, Mudhakar Srivatsa, Yucong Duan, and Wen- cai Du. Integritymr: Integrity assurance framework for big data analytics and management applications. In Big Data, 2013 IEEE International Con- ference on, pages 33–40. IEEE, 2013.

[18]. Wei Wei, Juan Du, Ting Yu, and Xiaohui Gu. Securemr: A service integrity assurance framework for mapreduce. In Computer Security Applications Conference, 2009. ACSAC'09. Annual, pages 73–82. IEEE, 2009.

[19]. Zhifeng Xiao and Yang Xiao. Security and privacy in cloud computing. IEEE Communications Surveys & Tutorials, 15(2):843–859, 2013.

[20]. Eunjung Yoon and Anna Squicciarini. Toward detecting compromised mapreduce workers through log analysis. In Cluster, Cloud and Grid Computing (CCGrid), 2014 14th IEEE/ACM International Symposium on, pages 41–50. IEEE, 2014.

[21]. Kehuan Zhang, Xiaoyong Zhou, Yangyi Chen, XiaoFeng Wang, and Yaop- ing Ruan. Sedic: privacy-aware data intensive computing on hybrid clouds. In Proceedings of the 18th ACM conference on Computer and communica- tions security, pages 515–526. ACM, 2011.

[22]. Qi Zhang, Lu Cheng, and Raouf Boutaba. Cloud computing: state-of-the- art and research challenges. Journal of internet services and applications, 1(1):7–18, 2010.

[23]. Minqi Zhou, Rong Zhang, Wei Xie, Weining Qian, and Aoying Zhou. Se- curity and privacy in cloud computing: A survey. In Semantics Knowledge and Grid (SKG), 2010 Sixth International Conference on, pages 105–112. IEEE, 2010.