Cloud Forensics : Isolating Cloud Instance

Mariam J. AlKandari, Huda F. AlRasheedi
& Ayed A. Salman
Computer Engineering Department
Kuwait University
Kuwait City, Kuwait
mariamalkandari@gmail.com, eng.hudaalrasheedi@gmail.com,
ayed.salman@ku.edu.kw

***Abstract*—Cloud computing has been the trending model for storing, accessing and modifying the data over the Internet in the recent years. Rising use of the cloud has generated a new concept related to the cloud which is cloud forensics. Cloud forensics can be defined as investigating for evidence over the cloud, so it can be viewed as a combination of both cloud computing and digital forensics. Many issues of applying forensics in the cloud have been addressed. Isolating the location of the incident has become an essential part of forensic process. This is done to ensure that evidence will not be modified or changed. Isolating an instant in the cloud computing has become even more challenging, due to the nature of the cloud environment. In the cloud, the same storage or virtual machine have been used by many users. Hence, the evidence is most likely will be overwritten and lost. The proposed solution in this paper is to isolate a cloud instance. This can be achieved by marking the instant that reside in the servers as "Under Investigation". To do so, cloud file system must be studied. One of the well-known file systems used in the cloud is Apache Hadoop Distributed File System (HDFS). Thus, in this paper the methodology used for isolating a cloud instance would be based on the HDFS architecture.**

***Keywords:* cloud computing; digital forensics; cloud forensics**

I. INTRODUCTION

In the last few years cloud computing has become the most used computing model by individuals and even by enterprise. Cloud is a new computing model that allows storing, accessing or even modifying the data over the Internet instead of using a computer hard drives or programs. As the files are far away from users stored in somewhere on the server in the datacenter, so this allows the client to view, modify, download and share their files everywhere and anytime over the Internet. Software as a Service (SaaS), Cloud Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) are three types of cloud computing models that can be provided by cloud service providers (CSPs) for the client [1]. Nowadays, it is possible to use the cloud on mobile phones by using the application provided by CSPs such as Dropbox, Google Drive, OneDrive and etc. Digital Forensic problem are rising due to the use of shared cloud everywhere and anytime among different users.

Digital forensics is the science of investigating an evidence of a computer based crime. Cloud computing plays a significant role in increasing many problems related to digital forensic and adding a new topic related to forensics which is cloud forensics. Cloud forensics can be defined as investigating for evidence over the cloud, so it can be viewed as a combination of both cloud computing and digital forensics [1].

Cloud forensics is a new research area that grabbed the attention of many researchers. It is facing lot of problems and challenges which led many researchers trying to find a proper solution for it. These challenges and problems are growing rapidly with the increase usage of the cloud. Our aim in this paper is to consider one of these problems and challenges to find the best solution to overcome the problem.

In case of cloud crime, digging and investigating is required in order to collect the evidences from the cloud. These evidences should not be changed or modified once acquired, that means the data on the cloud should be the same as before the crime. Thus, isolating the cloud instance is a must in order to keep the original version of the evidence. Isolating a cloud instance is one of the main issues that facing cloud forensics, because of the structure of the cloud [2]. In cloud the clients' data is distributed over several servers in many locations and not limited to a specific datacenter in one location.

In this paper we propose a solution for isolating a cloud instance by marking the servers that have evidence stored in it. In order to mark the servers, distributed file system (DFS) have been studied and understood carefully. In this paper, we will consider Apache Hadoop Distributed File System (HDFS), which is one of the frequently used cloud file system adapted by many vendors.

The structure of this paper is as follows: section II includes the basic information about HDFS and its architecture. Section III presents the literature review and related work by other researchers. Section IV introduces the reasons for isolating a cloud instant. Methodology and detailed design is described in section V. Finally, we conclude the paper in section VI.

II. BACKGROUND

Apache Hadoop is the most popular distributed file system implemented by Apache, which is used for distributed processing of big data. In 2015, 67% of companies used Hadoop as distributed file system [7]. Apache Hadoop uses master slave architecture where the NameNode is the master, and DataNodes are the slaves. The following subsections describe in details the architecture of Apache Hadoop.

A. NameNode

The structure of HDFS divided the files into two types; the metadata and the actual data of the file. The metadata of the

IRJECE

file is stored in a server called NameNode. This server contains all the information about the file such as access time, modification time and permissions to these files. These data are named inodes in NameNode. The location of the blocks in the DataNode server, that contain the data, is also critical information stored in NameNode. As it is shown in Fig. 1, each cluster consists of a single NameNode that maps many DataNodes in the same cluster. Moreover, many clients permitted to use same DataNode concurrently. NameNode saves the records of the metadata and the list of blocks in the RAM called image. If the image is stored in the local host's native file system then it is called checkpoint [3].

B. DataNode

DataNode is another storage server used to store the actual data of the file. DataNode consists of blocks and each block is represented using two files, see Fig. 1. The first file is used to store the data itself and the second one is used to store the metadata of the block including the block's checksum and generation stamp.

A new DataNode can join a cluster simply by assigning a unique namespace ID. This ID will be saved in the NameNode. When starting the DataNode, a handshaking is performed between the DataNode and NameNode in order to verify the namespace ID and the software version of the DataNode [3]. If the ID or the software version is not matching with the one on the NameNode then the DataNode will shut down.

DataNode is responsible to send block report containing block ID, generation stamp and length of each block replica stored in this DataNode. This report will be sent to the NameNode server to keep it up to date. DataNode also sends heartbeats to NameNode in order to ensure that the DataNode is operating and the block replicas are obtainable [3].

C. Operations on Cloud File

1) ReadProcess

When a user needs to read a file, the HDFS client at the user side will request the NameNode first to get the metadata of the requested file. The most important information needed at this stage is information related to locating a file. Thus NameNode will respond with the list of DataNodes that host the requested file blocks and a list of DataNodes that host the replica of that file.

The HDFS client chooses from the provided list, the closest DataNodes that host the file block based on their location from the user. After selecting the DataNodes, the HDFS client will directly communicate with that DataNode to fetch the file data. For instant, if the contacted DataNode didn't response, the client will choose another DataNode that host the file blocks replicas. DataNode fails to respond if the following cases occur:

- DataNode is out of the service (the node is done shutdown for some reason)
- DataNode is no longer hosting the file block and another DataNode is responsible for replying. This can happen in case of load balancing between the nodes.
- The replica of that block is corrupted

These are few reasons where the client has to contact other DataNodes for file replicas [3].

IRJECE

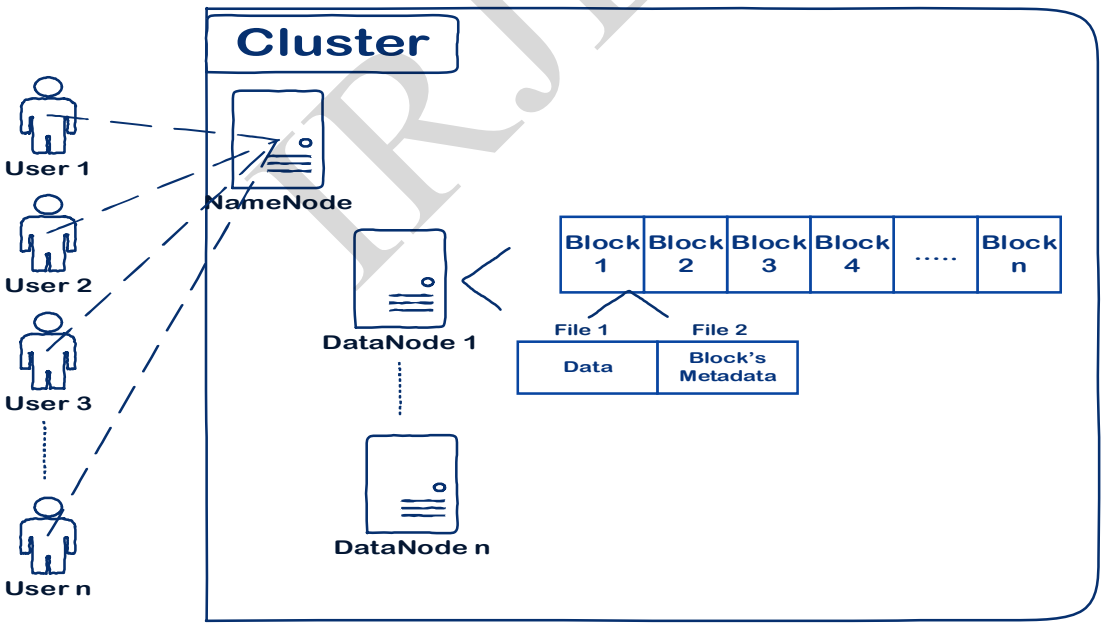

Fig. 1. Block Diagram of HDFS

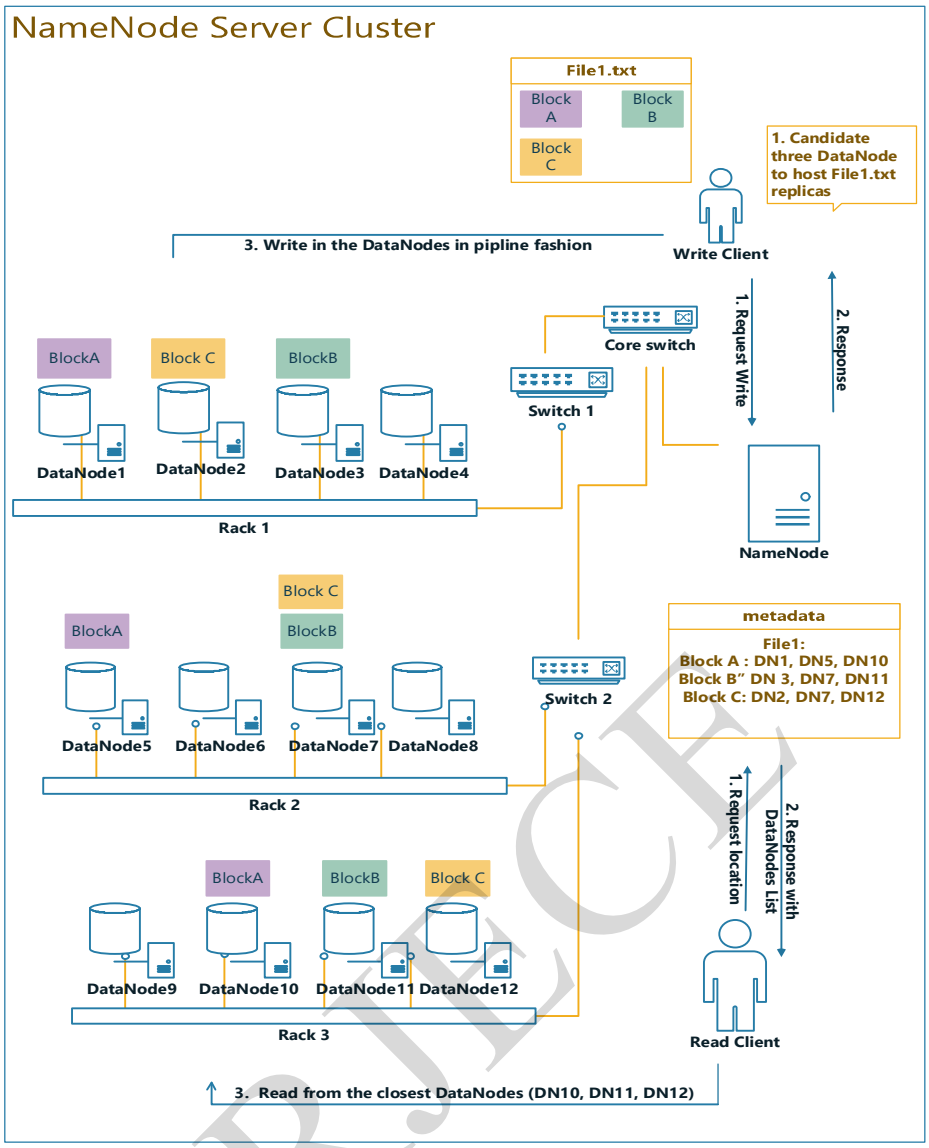

Fig. 2. Read-Write Example on Cloud using HDFS

2) Write Process

When a client needs to write a file, a client should create a file then then he/she can write on it. The HDFS client will request the NameNode to candidate three DataNodes to host the replicas. The NameNode will candidate by default three DataNodes based on an allocation strategies such as load balance, the location of the user and etc. After acquiring a list of candidate DataNodes, the client will write on these servers in pipeline fashion [3].

3) Share process

Sharing process consist of multiple users having access to the same file. A mutual exclusive condition is met, where only one user can write on the file at a time, while multiple users can read at the same time. When one user attempts to write a file, writer will get a lease for that file. During the writing process, the writing user will keep sending a heartbeat signal to the NameNode to continue getting the lease. The expiration of the lease is depending on soft limit or hard limit. If the soft limit is expired, another client can preempt the lease from the writing client. Moreover, if the client didn't renew the lease or close the fileafter one hour, the HDFS will assume that the client is quit and will automatically recover the lease, [3].

Fig. 2 illustrates an example of read and write process as well as an example of block allocation in HDFS architecture.

III. LITERATURE REVIEW

Cloud forensics is the new research area that attracted the attention of many researchers in the last few years. Because cloud computing has been widely used by many users and

IRJECE

enterprises, it was found that forensics is essential to be applied for the cloud in case of existence of crime.

Emi et al. in [1] gave a brief overview of the basics related cloud computing, digital forensics and cloud forensics. In addition, they described the challenges and forensics issues that will face the investigators when searching for an evidence in the cloud. They also provided some solutions and guidelines in order to overcome these obstacles [1]. Saad et al. in [2] also gave an overview of the challenges, problems, existing solutions and open problems related to cloud forensics.

Isolating a cloud instant is one of the main problems of cloud forensics. W. Delport et al. in [5] addressed the problem of isolating a cloud instant and proposed several solutions. In traditional forensic, the crime scene, where the crime took place, has to be secured and isolated to guarantee that evidence is safe from loss and not being tampered with. They believe that similar approach is also necessary in cloud forensic.

In [5] many techniques have been proposed to overcome this problem. These solutions are Instance Relocation, Server Farming, Failover, Address Relocation, Sandboxing, Man in the Middle (MITM) and Let's Hope for the Best (LHFTB). These techniques are similar to each other, as they depend on having multiple replicas of the instant. They claim that there is no best solution, however these techniques can be combined to get better solution. In addition none of these techniques has been tested or implemented.

Apache Hadoop Distributed File System HDFS is one of the most popular distributed file systems used for cloud computing. Konstantin et al. did their research explaining HDFS and its architecture [3]. They explained HDFS as a file system that consists of two types of servers: NameNode server, which stores the metadata of the files, and the other server is DataNodes that contains the data itself. R.Vijayakumari et al. did a comparison between two distributed file system used in storing big data which are Hadoop and google file system [6]. The researchers compared these distributed file systems related to design, goals, processes, file management, scalability, protection, security, cache management replication etc.

Mapreduce is a programming model or tool, which is used on top of Google and Hadoop file systems, in order to be able to perform a certain operation and analyze the data in an efficient way. Shankar et al. in [4] explained the Mapreduce that have been used to analyze big data in HDFS. K. Dwivedi et al in [8] present the need of HDFS and mentioned some terminology related to HDFS. The authors also summarized the related work done by the previous researchersby proposing a distributed file system and comparing their advantages and disadvantages. P. Leimich et al. in [7] proposed methodology for cloud forensic based on Hadoop. The methodology focused on investigating the NameNode RAM and analyzing the DataNodes. The proposed methodology consists of nine phases that start with evidence preservation and identification until the last step of reporting and presentation. The proposed approach has been tested on Hadoop environment with one master node and three slave nodes.

IV. REASONS FOR ISOLATING A CLOUD INSTANT

In a crime scene, isolating the scene with ropes or barricades is a must. The first responder will take this action to prevent the evidences from tampering or changing. Only authorized staff can enter the scene, to limit the destruction of evidence and protect them from loss. Changing in the evidence will reduce the admissibility of the evidence in the court [5]. In digital forensic, isolating and seizing the digital device is possible and acquiring the hard disk is also convenient.

However, in case of cloud forensic, it is impractical to isolate the whole server. This is because that the suspected data stored in a server does not only belong to the user, as there are many users sharing the same server and insuring the availability of user data is one of the main requirements in the cloud. Thus, proposing a solution to isolate the suspected instant without affecting the data is helpful in case of digital crime.

V. METHODOLOGY

Hadoop is an open source framework which is used in HDFS architecture [4]. The proposed methodology will use this fact to upgrade Hadoop to support cloud forensic in general and isolate a cloud instant in specific. The proposed solution assumes that the Cloud Service Provider (CSP) uses HDFS. The framework methodology consists of the following steps:

- Upgrade Hadoop framework to include a flag called "Investigation". The "1" flag indicates that the file is under investigation and "0" indicate a normal file.
- In case of cloud crime, the forensic examiner should contact the Cloud Service Provider (CSP) administrator to locate the suspected file.
- The administrator requests the NameNode to locate the suspected file and get the file replicas.
- The administrator contact the NameNode to request it to mark the file as "Investigation".
- NameNode will change the flag from 0 to 1 and send acknowledge back to the administrator indicating that the file is marked as "Investigation".

HDFS architecture is fault tolerant because it includes a backup for the NameNode. Thus the file marked as "Investigation" in the original NameNode will also has its flag sets to 1 in the backup NameNode. There is no change done to the backup technique used in the original HDFS architecture.

The following scenario describes read and write process of a suspected file.

A. Read Suspected File Scenario

Assume user "A" wants to read to the suspected file called "File1.txt". Then the following events will occur, see Fig. 3:

- HDFS client of user "A" will request the NameNode first to get the location of "File1.txt".
- NameNode will check the flag "Investigation" of "File1.txt" if it 1 or 0.
- If "File1.txt" is under investigation which means its flag is 1, then NameNode will respond to user "A"

IRJECE

with the message "Cannot open, the file is under investigation".
- HDFS client will close the session.

B. Write on Suspected file Scenario

Assume that the user "A" wants to write on suspected file called "File1.txt". Then the following events will occur:

- HDFS client of user "A" will request the NameNode first to get the location of "File1.txt" first to write on it.
- NameNode will check the flag "Investigation" of "File1.txt" if it 1 or 0.
- If "File1.txt" is under investigation and its flag is 1, then NameNode will respond to user "A" with the message "You cannot write on this file, file is under investigation".
- HDFS client will close the session.

C. Sharing Suspected File Among Different Users Scenario

Assume that user "A" wants to share a suspected file called "File1.txt" with user "B". Then the following events will occur:

- HDFS client of user "A" will request the NameNode first to get the location of "File1.txt".
- NameNode will check the flag "Investigation" of "File1.txt" if it is 1 or 0.
- If "File1.txt" is under investigation that is the flag is 1, then NameNode will respond to user "A" with the message "Cannot share the file, it is under investigation".
- HDFS client will close the session.

D. Suspected File is Shared Among Different Users Scenario

Assume that user "A" wants to share a suspected file called "File1.txt" with user "B". Then the following events will occur:

- HDFS client of user "A" will request the NameNode first to get the location of "File1.txt".
- NameNode will check the flag "Investigation" of "File1.txt" if it is 1 or 0.
- If "File1.txt" is under investigation and its flag is 1, then NameNode will respond to user "A" with the message "Cannot share the file, it is under investigation".
- HDFS client will close the session.

User B will get the exact response as user A, whether user B request to read the file at the same time of user A's request, or after some duration of time.

Assume that users "A" or "B" want to write on the same file. Then the following events will occur:

- HDFS client of user "A" or "B" will request the NameNode first to get the location of "File1.txt" first to write on it.
- NameNode will check the flag "Investigation" of "File1.txt" if it is 1 or 0.
- If "File1.txt" is under investigation andits flag is 1, then NameNode will respond to user "A" or "B" with the message "You cannot write on this file, file is under investigation".
- HDFS clients will close the session.

After the case is closed and the investigation over the cloud is finished, forensics examiner should contact the Cloud Service Provider (CSP) administrator to reset "Investigation" flag back to normal. The framework methodology consists of the following steps:

- In case investigation is finished, the forensic examiner should contact the Cloud Service Provider (CSP) administrator to locate the suspected file.
- The administrator requests the NameNode to locate the suspected file and get the file replicas.
- The administrator contact the NameNode to request changing of "Investigation" flag back to normal that is to reset it back to 0.
- NameNode will change the flag from 1 to 0 and send acknowledge back to the administrator indicating that the file is marked as normal file.

After the file is marked as normal and the investigation flag is back to zero, the user will be able to read, write and share the file with other users.

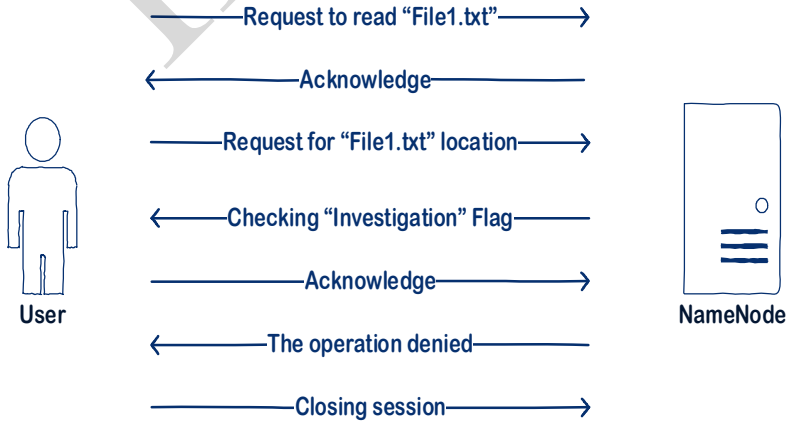

Fig. 3. Read Under Investigation File example

IRJECE

VI. CONCLUSION

Forensic investigation has many challenges when it comes to cloud computing. One of the main issues of cloud forensic is isolating a cloud instant. Isolating a cloud instance is an essential step in cloud forensic to ensure that the evidence has not changed or tampered. Providing a solution for this challenged problem relies on the distributed file system used in the cloud environment.

This paper explained the main terminologies of Apache Hadoop Distributed File System (HDFS) and provided an explanation of write, read and shared processes as framework. In this paper, a solution for isolating a cloud instant in Hadoop distributed file system is proposed. The proposed solution marked the targeted evidence as "Investigation" and ensures that this evidence cannot be written or read. For the future work, we intend to test this methodology on real Hadoop distributed file system environment.

ACKNOWLEDGMENT

Authors would like to thank Kuwait University for this opportunity to embark on this research. In addition, Ms. Mariam would like to express her gratitude to her motherTahani for her supports and efforts in editing and reviewing this research.

REFERENCES

[1] Morioka, Emi, and Mehrdad S. Sharbaf. "Cloud Computing: Digital Forensic Solutions." In Information Technology-New Generations (ITNG), 2015 12th International Conference on, pp. 589-594. IEEE, 2015.

[2] Alqahtany, Saad, Nathan Clarke, Steven Furnell, and Christoph Reich. "Cloud Forensics: A Review of Challenges, Solutions and Open Problems." In Cloud Computing (ICCC), 2015 International Conference on, pp. 1-9. IEEE, 2015.

[3] Shvachko, Konstantin, Hairong Kuang, Sanjay Radia, and Robert Chansler. "The hadoop distributed file system." In 2010 IEEE 26th symposium on mass storage systems and technologies (MSST), pp. 1-10. IEEE, 2010.

[4] Manikandan, Shankar Ganesh, and Siddarth Ravi. "Big data analysis using apache hadoop." In IT Convergence and Security (ICITCS), 2014 International Conference on, pp. 1-4. IEEE, 2014.

[5] Delport, Waldo, Michael Köhn, and Martin S. Olivier. "Isolating a cloud instance for a digital forensic investigation." In ISSA. 2011.

[6] Vijayakumari, R., R. Kirankumar, and K. Gangadhara Rao. "Comparative analysis of Google File System and Hadoop Distributed File System." International Journal of Advanced Trends in Computer Science and Engineering 3, no. 1 (2014): 553-558.

[7] Leimich, Petra, Josh Harrison, and William J. Buchanan. "A RAM triage methodology for Hadoop HDFS forensics." Digital Investigation 18 (2016): 96-109.

[8] Dwivedi, Kalpana, and Sanjay Kumar Dubey. "Analytical review on Hadoop Distributed file system." In Confluence The Next Generation Information Technology Summit (Confluence), 2014 5th International Conference-, pp. 174-181. IEEE, 2014.

IRJECE