International Journal of Engineering Research and Science & Technology

IJERST

ISSN : 2319-5991

www.ijerst.com

Email: editor@ijerst.com or editor.ijerst@gmail.com

Int. J. Engg. Res. & Sci. & Tech. 2023

International Journal of Engineering Research and Science & Technology

ISSN 2319-5991 www.ijerst.com

Vol. 11, Issuse.2,April 2023

DETECTING SUSPICIOUS FILE MIGRATION OR REPLICATION IN CLOUD

Dr.M.UMA DEVI , [1],Rayudu Hemanth Naga Pavan [2],Chitta Sambasiva [3],Patchipala Venkata Sairam [4],Gunnam Revathi [5],Srirangam Sai Sri Charishma [6]

ABSTRACT

There has been a prolific rise in the popularity of cloud storage in recent years. While cloud storage offers many advantages such as flexibility and convenience, users are typically unable to tell or control the actual locations of their data. This limitation may affect users' confidence and trust in the storage provider, or even render cloud unsuitable for storing data with strict location requirements. To address this issue, we propose a system called LAST-HDFS which integrates Location-Aware Storage Technique (LAST) into the open source Hadoop Distributed File System (HDFS). The LAST-HDFS system enforces location-aware file allocations and continuously monitors file transfers to detect potentially illegal transfers in the cloud. Illegal transfers here refer to attempts to move sensitive data outside the (”legal”) boundaries specified by the file owner and its policies. Our underlying algorithms model file transfers among nodes as a weighted graph, and maximize the probability of storing data items of similar privacy preferences in the same region. We equip each cloud node with a socket monitor that is capable of monitoring the real-time communication among cloud nodes. Based on the real-time data transfer information captured by the socket monitors, our system calculates the probability of a given transfer to be illegal. We have implemented our proposed framework and carried out an extensive experimental evaluation in a large-scale real cloud environment to demonstrate the effectiveness and efficiency of our proposed system.

1.INTRODUCTION

With the ever-increasing popularity of cloud computing, the demand for cloud storage has also increased exponentially. Computing firms are no longer the only consumers of cloud storage and cloud computing, but rather average businesses, and even end-users, are taking advantage of the immense capabilities that cloud services can provide. While enjoying the flexibility and convenience brought by cloud storage, cloud users release control over their data, and particularly are often unable to locate the actual their data; this could be in-state, in-country, or even out-of-country. Lack of location control may cause privacy breaches for cloud users (e.g., hospitals) who store sensitive data (e.g., medical records) that are governed by laws to remain within certain geographic boundaries and borders. Another situation where this problem arises is with governmental entities that require all data to be stored in the same country that the government operates in; this challenge has seen difficulties with cloud service providers (CSPs) quietly moving data out-of-country or being bought out by foreign companies.

ASSOCIATE PROFESSOR

DEPT OF COMPUTER SCIENCE AND ENGINEERING

PRAGATI ENGINEERING COLLEGE(A),SURAMPALEM(EAST GODAVARI)A.P,INDIA

For example, Canadian laws demand that personal identifiable data must be stored in Canada. However, large cloud infrastructure like the Amazon Cloud has more than 40 zones distributed all over the world [1], which makes it very challenging to provide guaranteed adherence to regulatory compliance. Even Hadoop, which historically has been managed as a geographically confined distributed file system, is now deployed in large scale across different regions (see Facebook Prism or recent patent).

To date, various tools have been proposed to help users verify the exact location of data stored in the cloud, with emphasis on post-allocation compliance. However, recent work has acknowledged the importance of a proactive location control for data placement consistent with adopters' location

requirements, to allow users to have stronger control over their data and to guarantee the location where the data is stored. In this work, we infiltrate into one of the most widely adopted cloud data storage systems– Hadoop Distributed File System (HDFS), and design an enhanced HDFS system, called LAST-HDFS. The LAST-HDFS extends HDFS' capabilities to achieve location-aware file allocations and file.

2. LITERATURE SURVEY

1. AWS global infrastructure

The AWS Cloud infrastructure is built around AWS Regions and Availability Zones. An AWS Region is a physical location in the world where we have multiple Availability Zones. Availability Zones consist of one or more discrete data centers, each with redundant power, networking, and connectivity, housed in separate facilities. These Availability Zones offer you the ability to operate production applications and databases that are more highly available, fault tolerant, and scalable than would be possible from a single data center. For the latest information on the AWS Cloud Availability Zones and AWS Regions,

2. Geographically-distributed file system using coordinated namespace replication

A cluster of nodes implements a single distributed file system, comprises at least first and second data centers and a coordination engine process. The first data center may comprise first DataNodes configured to store data blocks of client files, and first NameNodes configured to update a state of a namespace of the cluster. The second data center, geographically remote from and coupled to the first data center by a wide area network, may comprise second DataNodes configured to store data blocks of client files, and second NameNodes configured to update the state of the namespace. The first and second NameNodes are configured to update the state of the namespace responsive to data blocks being written to the DataNodes. The coordination engine process spans the first and second NameNodes and coordinates updates to the namespace stored such that the state thereof is maintained consistent across the first and second data centers.

3. Last-HDFS: Location-aware storage technique for Hadoop distributed file system

Enabled by the state-of-the-art cloud computing technologies, cloud storage has gained increasing popularity in recent years. Despite of the benefit of flexible and reliable data access offered by such services, users have to bear with the fact of not actually knowing the whereabouts of their data. The lack of knowledge and control of the physical locations of data could raise legal and regulatory issues, especially for certain sensitive data that are governed by laws to remain within certain geographic boundaries and borders. In this paper, we study the problem of data placement control within distributed file systems supporting cloud storage. Particularly, we consider the open source Hadoop file system (HDFS) as the underlying architecture, and propose a location-aware cloud storage system, named LAST-HDFS, to support and enforce location-aware storage in HDFS-based clusters. In addition, it also includes a monitoring system deployed at individual hosts to oversee and detect potential data placement violations due to the existence of malicious data nodes. We carried out an extensive experimental evaluation in a real cloud environment that demonstrates the effectiveness and efficiency of our proposed system.

4. One of our hosts in another country

Physical location of data in cloud storage is an increasingly urgent problem. In a short time, it has evolved from the concern of a few regulated businesses to an important consideration for many cloud storage users. One of the characteristics of cloud storage is fluid transfer of data both within and among the data centres of a cloud provider. However, this has weakened the guarantees with respect to control over data replicas, protection of data in transit and physical location of data. This paper addresses the lack of reliable solutions for data placement control in cloud storage systems. We analyse the

currently available solutions and identify their shortcomings. Furthermore, we describe a high-level architecture for a trusted, geolocation-based mechanism for data placement control in distributed cloud storage systems, which are the basis of an on-going work to define the detailed protocol and a prototype of such a solution. This mechanism aims to provide granular control over the capabilities of tenants to access data placed on geographically dispersed storage units comprising the cloud storage.

5. A position paper on data sovereignty: The importance of geolocating data in the cloud

In this paper we define the problem and scope of data sovereignty - the coupling of stored data authenticityand geographical location in the cloud. Establishing sovereignty is an especially important concern amid legaland policy constraints when data and resources are virtualized and widely distributed. We identify the keychallenges that need to be solved to achieve an effective and un-cheatable solution as well as propose an initialtechnique for data sovereignty**.**

3. EXISTING SYSTEM

3.1 Existing System

3.1.1 Secure and constant cost public cloud storage auditing with duplication

In order for cloud storage to be effective, data integrity and storage efficiency are two key nodes. Data integrity for cloud storage is guaranteed by POR and PDP approaches. Storage efficiency is increased by POW, which safely deletes redundant data from the storage server. To accomplish both data integrity and storage efficiency, however a minimal combination of the two strategies leads to non-trivial duplication of information (i.e., authentication tags), which is in opposition to POW's goals. Recent solutions to this issue have been shown to be insecure and to incur significant computational ad communication costs. In order to offer effective and safe data integrity auditing together with storage, a new solution is required.

In this study, we present a novel strategy for the solution of this open problem, based on homomorphic linear authenticators and polynomial-based authentication tags. Deduplication of files and the related authentication tags is possible thanks to our architecture. Storage deduplication and data integrity auditing are accomplished simultaneously. Constant real-time communication and computational expense on the user's end are further characteristics of our suggested approach. Both batch and public audits are supported.

As a result, our suggested method performs better than current POR and PDP schemes while incorporating deduplication as an additional utility. We use the Computational Diffie-Hellman problem to demonstrate the security of our suggested system. Experimental findings of Amazon AWS and numerical analysis demonstrate how effective and scalable our system is.

3.2 Proposed system

In this paper, we propose two secure systems, SecCloud and SecCloud-D, in an effort to achieve data integrity and deduplication in the cloud.

By combining the management of a MapReduce cloud with an auditing entity, SecCloud enables its users to ensure the authenticity of data stored in the cloud and to generate metadata tags prior to upload. SecCloud+ enables the guarantee of file confidentiality in addition to supporting integrity auditing and secure deduplication. We propose an approach for performing direct audits of encrypted data's integrity

3.2 PROPOSED SYSTEM

This section introduces our methodology to detect the DDoS attack. The five-fold steps application process of data mining techniques in network systems discussed in characterizes the followed methodology. The main aim of combining algorithms used in the proposed approach is to reduces noisy and irrelevant network traffic data before preprocessing and classification stages for DDoS detection while maintaining high performance in terms of accuracy, false positive rate and running time, and low resources usage. Our approach starts with estimating the entropy of the FSD features over a time-based sliding window. When the average entropy of a time window exceeds its lower or upper thresholds the co-clustering algorithm split the received network traffic into three clusters.

When the average entropy of a time window exceeds its lower or upper thresholds the co-clustering algorithm split the received network traffic into three clusters. Entropy estimation over time sliding windows allows to detect abrupt changes in the incoming network traffic distribution which are often caused by DDoS attacks. Incoming network traffic within the time windows having abnormal entropy values is suspected to contain DDoS traffic.

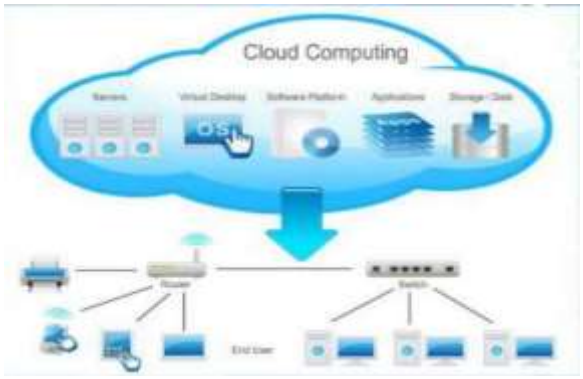
The focus only on the suspected time windows allows to filter important amount of network traffic data, therefore only relevant data is selected for the remaining steps of the proposed approach. Also, important resources saved when no abnormal entropy occurs. In order to determine the normal cluster, we estimate the information gain ratio based on the average

entropy of the FSD features between the received network traffic data during the current time window and each one of the obtained clusters.

As discussed in the previous section during a DDoS period the generated amount of attack traffic is largely bigger than the normal traffic. Hence, estimating the information gain ratio based on the FSD features allows to identify the two cluster that preserve more information about the DDoS attack and the cluster that contains only normal traffic. Therefore, the cluster that produce lower information gain ratio is considered as normal and the remaining clusters are considered as anomalous. The information gain ratio is computed for each cluster.

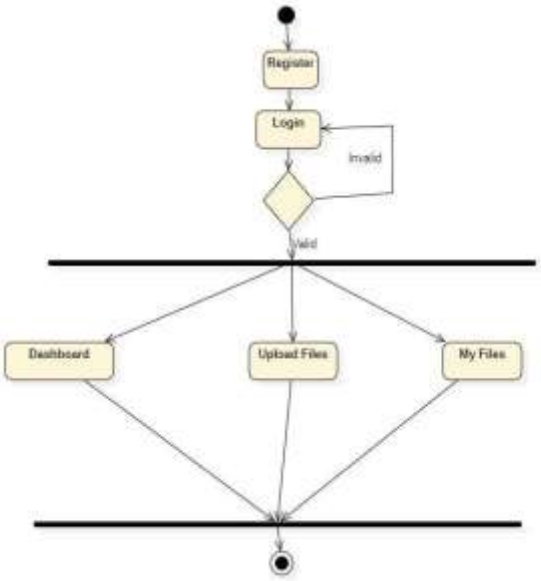
SYSTEM ARCHITECTURE .

Below diagram depicts the whole system architecture of the project



Activity Diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



5.SYSTEM IMPLEMENTATION

There are 3 modules:

1. FileLoader
2. NameNode
3. DataNode
4. File Loader:

In this module authorized users will register by creating an account and login to upload files which are stored in my files. Now details of uploaded files and total count of uploaded files will be displayed.

▪Register

- Login
- Upload Files
- My Files
- Logout

Name Node:

In this module there are two fields named as File Loader Request and Authorized File Loader. In this file loader request we can accept or reject the user request after creating in their account. In authorized file loader all the accepted user details are stored and displayed.

- Login
- File Loader Request
- Authorized File Loader
- Logout

TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

6.1 TYPES OF TESTING

■ Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the

documented specifications and contains clearly defined inputs and expected results.

■ Integration testing

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

■ Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

7.RESULTS

Fig. 7.1 Home page of the project

Fig. 7.2 Contact page

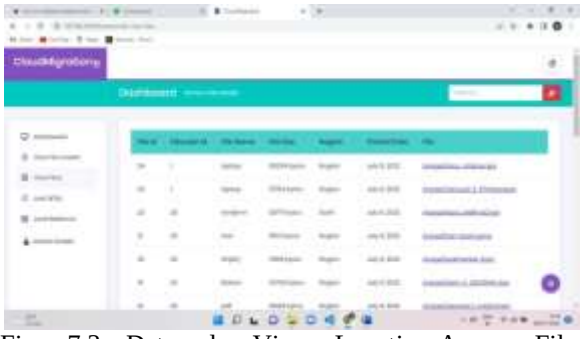


Fig. 7.3 Datanode View Location-Aware File Allocation

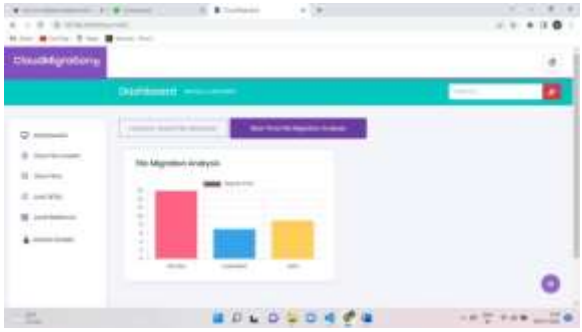


Fig. 7.4 Datanode View Real-Time File Migration Analysis

Fig. 7.5 Datanode View Real-Time File Migration Analysis

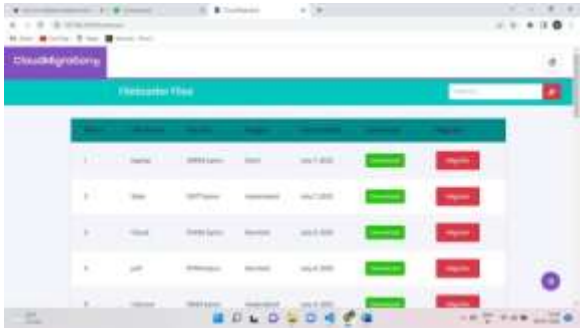


Fig. 7.6 Attacker View File

8. CO NCLUSION & FUTURE WORK

In this paper, we build, on top of the existing HDFS, a novel LAST-HDFS system to address the data placement control problem in the cloud. LAST-HDFS supports policy-driven file loading that enables location-aware storage in cloud sites. More importantly, it also ensures that the location policy is enforced regardless of data replication and load balancing processes that may affect policy compliance. Specifically, an efficient LP-tree and Legal File Transfer graph were designed to help optimally allocate files with similar location preferences to the most suitable cloud nodes which in turn enhance the chance of detecting illegal file transfers. We have conducted extensive experimental studies in both a real cloud testbed and a large-scale simulated cloud environment. Our experimental results have shown the effectiveness and efficiency of the proposed LAST-HDFS system.

In the future, we plan to take into account more complicated policies to capture other privacy requirements other than the location. We will adopt more sophisticated policy analysis algorithm and compute the integrated policy as the representative policy at each node to help speed up the policy comparison and selection of nodes for the newly uploaded files. Moreover, we also plan to leverage Intel SGX technology to secure socket monitors from being compromised.

REFERENCES

[1] Amazon, "AWS global infrastructure," in https://aws.amazon.com/aboutaws/global-infrastructure/, 2017.

[2] C. Metz, "Facebook tackles (really) big data with project prism," in https://www.wired.com/2012/08/facebookprism/, 2012.

[3] K. V. SHVACHKO, Y. Aahlad, J. Sundar, and P. Jeliazkov, "Geographically-distributed file system using coordinated namespace replication," in https://www.google.com/patents/WO2015153045A1?cl=zh, 2014.

[4] C. Liao, A. Squicciarini, and L. Dan, "Last-hdfs: Location-aware storage technique for hadoop distributed file system," in IEEE International Conference on Cloud Computing (CLOUD), 2016.

[5] N. Paladi and A. Michalas, ""one of our hosts in another country": Challenges of data geolocation in cloud storage," in International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE), 2014, pp. 1–6.

[6] Z. N. Peterson, M. Gondree, and R. Beverly, "A position paper on data sovereignty: The importance of geolocating data in the cloud." in HotCloud, 2011.

[7] A. Squicciarini, D. Lin, S. Sundareswaran, and J. Li, "Policy driven node selection in mapreduce," in 10th International Conference on Security and Privacy in Communication Networks (SecureComm), 2014.

[8] J. Li, A. Squicciarini, D. Lin, S. Liang, and C. Jia, "Secloc: Securing location-sensitive storage in the cloud," in ACM symposium on access control models and technologies (SACMAT), 2015.

[9] E. Order, "Presidential executive order on strengthening the cybersecurity of federal networks and critical infrastructure," in https://www.whitehouse.gov/the-press-office/2017/05/11/presidentialexecutive-orderstrengtheningcybersecurity-federal, 2017.

[10] "Hdfs architecture," http://hadoop.apache.org/docs/stable/ hadoop-project-dist/hadoop-hdfs/HdfsDesign.html.