ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING

National Conference on Advanced Innovation in Engineering and Technology (NCAIET-2015)

Alva's Institute of Engineering and Technology, Moodbidri

Vol. 3, Special Issue 1, April 2015

Towards Secure and Consistency Dependable in Large Cloud Systems

Sahana M S[1], Chanchal Antony[2]

M.Tech Scholar, Department of Computer Science, Alvas Institute of Engineering and Technology, Moodbidri, Karnataka, India[1]

Senior Assistant Professor, Department of Computer Science, Alvas Institute of Engineering and Technology, Moodbidri, Karnataka, India[2]

Abstract: Cloud storage services have become very popular due to their well security, availability and cost effective. To provide presence always-on access, a cloud service provider (CSP) maintains several replicas for each piece of data on geographically distributed servers. A key problem of using the replication Technique in clouds is that it is very expensive and crude to achieve strong consistency on everywhere. So first present a causal consistency as a service (CaaS) model, which consists of a large data cloud and several small audit clouds. In the CaaS model, a data cloud is maintained by a CSP, that constitute an audit cloud can verify whether the data cloud provides the promised level of consistency or not. So propose a two-level auditing, which only requires a loosely synchronized clock in the audit cloud. Then design different algorithms to quantify the seriousness of violations with two metrics: the commonality of violations, and the staleness of the value of a read. Finally, a heuristic auditing strategy (HAS) method is used to reveal as many violations as possible.

Keywords: Two level auditing, CaaS model, commonality, staleness, heuristic auditing strategy (HAS), cloud storage.

I. INTRODUCTION

Cloud computing has become commercially popular ,it promises to provide scalability, elasticity, and availability at a low cost [1], Cloud storage services can be regarded as a typical service in cloud computing, It includes in delivery of data storage as a service, and also it includes database-like services and network attached storage, depend upon utility computing basis. By using the cloud storage services, the customers can access data stored in a cloud anytime and anywhere using any device.To meet the promise of everywhere access, the cloud service provider (CSP) stores single data replicas on multiple geographically distributed servers. A problem of using the replication technique in clouds is very expensive and crude to achieve strong consistency on a worldwide scale. If a data is replicated in four clouds then it has to be updated with new version before the user receives the data otherwise the old version will be accessed by the user, because sometimes old version may not satisfy all the needs.

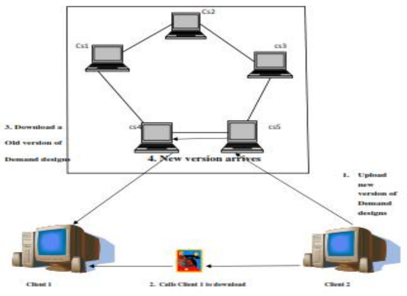
Cs1
Cs2
cs3
cs4
cs5
3. Download a Old version of Demand designs
4. New version arrives
1. Upload new version of Demand designs
Client 1
2. Calls Client 1 to download
Client 2

Fig 1: An Application requires causal consistency

As shown in the above figure within a cloud server there are five clouds will be there. Client 1 at china, Client 2 at Bombay. Client 1 and 2 cooperating on a project using cloud storage services. Client 2 uploading a new version of the demand requirements to a cs5. Client 2 calls client 1 to download the latest version of demand designs. So that a causal consistency is established between client 1 and client 2. It means that, which ensures that client 2 update is committed to all of the replicas before client 1 read. If cloud provides only eventual consistency client 1 access an old version of designs form cs4. Some times that may not satisfy all needs.

In cloud storage, consistency determines correctness and also the actual cost per transaction. Hence presenting the consistency as a service (CaaS) model for this situation. The CaaS model consists of a large data cloud and multiple small audit clouds. The implementation of the data cloud is opaque to all users due to the virtualization technique. Thus, it is hard for the users to verify whether each replica in the data cloud is the latest one or not. Inspired by the solution in, it allows the users in the audit cloud to verify cloud consistency by analyzing a trace of interactive operations.

An auditor is elected from the audit cloud to perform global auditing with a global trace of operations. Local auditing focuses on monotonic-read and read-your-write consistencies, which can be performed by a light-weight online algorithm.

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING

National Conference on Advanced Innovation in Engineering and Technology (NCAIET-2015)

Alva's Institute of Engineering and Technology, Moodbidri

Vol. 3, Special Issue 1, April 2015

Global auditing focuses on causal consistency, which is performed by constructing a directed graph. If the constructed graph is a directed acyclic graph (DAG), Hence heuristic auditing strategy(HAS) method is used which adds appropriate reads to cover as many violations as possible. It helps in design algorithms to quantify the severity of violations with different metrics. The heuristic auditing strategy (HAS) covered as many violations as possible. Several experiments were performed using a combination of simulations and a real cloud deployment to validate HAS.

Internet applications often rely on globally distributed highly available storage systems to meet the promise of ubiquitous 24x7 operations. The main challenge in building a globally distributed system is dealing with network partitions. Brewer's CAP principle states that any shared data system can provide only two of the following three properties: consistency, availability, and partition tolerance. Since partitions are inevitable in wide-area networks, storage system designers are only left with the option of trading-off consistency for availability. Traditional systems such as databases and file-systems choose to sacrifice availability and only offer strict consistency.

II. LITERATURE SURVEY

Over the last few years, Cloud storage systems and so-called NoSQL data stores have found widespread adoption. In contrast to traditional databases, these storage systems typically sacrifice consistency in favor of latency and availability as mandated by the CAP theorem [1], so that they only guarantee eventual consistency. Existing approaches to bench-mark these storage systems typically omit the consistency dimension or did not investigate eventuality of consistency guarantees. In this work we present a novel approach to benchmark staleness in distributed data stores and use the approach to evaluate Amazon's Simple Storage Service (S3).

A new class of data storage systems, called NoSQL(Not Only SQL), has emerged to complement traditional database systems, with rejection of general ACID transactions as one common feature. Different platforms, and indeed different primitives within one NoSQL[2] platform, can offer various consistency properties, from Eventual Consistency to single-entity ACID. For the platform provider, weaker consistency should allow better availability, lower latency, and other benefits.

This paper investigates what consumers observe of the consistency and performance properties of various offerings. We find that many platforms seem in practice to offer more consistency than they promise[2]; we also find cases where the platform offers consumers a choice between stronger and weaker consistency, but there is no observed benefit from accepting weaker consistency properties.

Today we are increasingly more dependent on critical data stored in cloud data centers across the world. To deliver high-availability and augmented performance, different replication schemes are used to maintain consistency among replicas [3]. With classical consistency models, performance is necessarily degraded, and thus most highly-scalable cloud data centers sacrifice to some extent consistency in exchange of lower latencies to end-users. More so, those cloud systems blindly allow stale data to exist for some constant period of time and disregard the semantics and importance data might have, which undoubtedly can be used to gear consistency more wisely, combining stronger and weaker levels of consistency.

Cloud storage solutions promise high scalability and low cost. Existing solutions, however, differ in the degree of consistency they provide. Our experience using such systems indicates that there is a non-trivial trade-off between cost, consistency and availability. High consistency implies high cost per transaction and, in some situations, reduced availability. Low consistency is cheaper but it might result in higher operational cost [4] because of, e.g., overselling of products in a Web shop.
In this paper, we present a new transaction paradigm that not only allows designers to define the consistency guarantees on the data instead at the transaction level, but also allows to automatically switching consistency guarantees at runtime. We present a number of techniques that let the system dynamically adapt the consistency level by monitoring the data and/or gathering temporal statistics of the data.

Motivated by the increasing popularity of eventually consistent key-value [5]stores as a commercial service, we address two important problems related to the consistency properties in a history of operations on a read/write register (i.e., the start time, finish time, argument, and response of every operation). First, we consider how to detect a consistency violation as soon as one happens. To this end, we formulate a specification for online verification algorithms, and we present such algorithms for several well-known consistency properties. Second, we consider how to quantify the severity of the violations, if a history is found to contain consistency violations.

III. PROBLEM STATEMENT

A key problem is the usage of replication. Auditing technique in clouds is that it is very expensive to achieve. Auditing consistency on a worldwide scale cannot be achieved. Although the infrastructures under the cloud are much more powerful and reliable than personal computing devices, they are still facing the broad range of both internal and external threats for data integrity. There do exist various motivations for CSP to behave unfaithfully toward the cloud users regarding their outsourced data status. In particular, simply downloading all the data for its integrity verification is not a practical solution due to the expensiveness in I/O and transmission cost across the

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING

National Conference on Advanced Innovation in Engineering and Technology (NCAIET-2015)

Alva's Institute of Engineering and Technology, Moodbidri

Vol. 3, Special Issue 1, April 2015

network. Besides, it is often insufficient to detect the data corruption only when accessing the data, as it does not give users correctness assurance for those accessed data and might be too late to recover the data loss or damage.
Encryption does not completely solve the problem of protecting data privacy against third-party auditing but just reduces it to the complex key management domain. Unauthorized data leakage still remains possible due to the potential exposure of decryption keys.

Objectives
1) Main objective is to provide a CAAS model and also a two level auditing structure to help users to verify whether the cloud service provider is providing the promised consistency.
2) It has to meet the promise of providing 24/7 access for end users.
3) Data should be well secured in cloud server.
4) It has to maintain causal consistency.

Methodology
1) In this concept four modules are there.
2) Data Owner
3) Cloud Server
4) Audit Cloud
5) End User

A. Data Owner
Data owner will browse the data from different sources and he will send the data to the cloud server. While sending the data, the data should be encrypted form , by using AES algorithm he ill encrypt the data, and also for each and every data MAC will be generated, depending upon the file content MAC will be generates.
While encrypting the data secret key is generated by using SHA1 algorithm, after encrypting the data, data owner will submit the data to the cloud server; data should be stored in the cloud server.

B. Cloud Server
Cloud server consists of many small clouds. They must register with the cloud server. Small clouds also called as audit clouds. Among all the audit clouds auditor is elected, he is responsible for all activities.
When a data is updated to a single cloud, the same data should update to all the clouds. It has to maintain causal consistency among all the clouds.

C. Audit Cloud
Auditor is elected among all the clouds. He is responsible to verify all the data in cloud server. He is act as an intermediate between cloud server and end user. If any data is corrupted then new MAC address will be generated. Auditor will compare the MAC address if any changes occur, he will report all the information to the data owner.

By using HAS method it reveals as many violations as possible.

D. End user
End user will request the data to the data owner. He will permit the end user to read the data either write the data. Then only he can permit to access the cloud server. If he violates the request then audit cloud will report to the data owner. If any data is requested audit cloud will perform two level of auditing. Local auditing is to check the type of request, and global auditing is to check the file assessment, if any data is corrupted audit will repair the data and then provide to the end user.

IV. EXISTING SYSTEM
Although the existing schemes aim at providing integrity verification for different data storage systems, the problem of supporting both public audit ability and data dynamics has not been fully addressed. How to achieve a secure and efficient design to seamlessly integrate these two important components for data storage service remains an open challenging task in Cloud Computing.

Existing commercial clouds usually restrict strong consistency guarantees to small datasets, or provide only eventual consistency. Described several solutions to achieve different levels of consistency while deploying database applications on replication. Trace based verification technique is used earlier.

V. PROPOSED SYSTEM
Hence propose a Heuristic Auditing Strategy (HAS) which adds appropriate reads to reveal as many violations as possible.
Key contributions are as follows:
1) Present a causal consistency as a service (CaaS) model that constitute an audit cloud can verify whether the data cloud provides the promised level of consistency or not.
2) It Propose a two-level auditing structure that only requires a loosely synchronized clock for ordering operations in an audit cloud.
3) Design different algorithms to quantify the severity of violations with different metrics.
4) Heuristic auditing strategy (HAS) covered as many violations as possible.

VI. SYSTEM DESIGN
UML stands for Unified Modeling Language. It represents a unification of the concepts and notations presented by the three amigos.

The goal is for UML to become a common language for creating models of object oriented computer software. UML is popular for its diagrammatic notations. As shown in the fig 2, it consisting of four modules.

Firstly data owner will browse and upload the file content by using AES algorithm for encryption and decryption. According to the file content MAC key will be generated, at the same time Meta data will send to the audit cloud.

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING

National Conference on Advanced Innovation in Engineering and Technology (NCAIET-2015)

Alva's Institute of Engineering and Technology, Moodbidri

Vol. 3, Special Issue 1, April 2015

In cloud server there are several clouds will be there, each data should be replicated to the all clouds to provide causal consistency.

Remote user will request for the permission with the data owner. After getting the permission he will send request to the audit cloud.

Audit cloud will act as an intermediate between end user and cloud server. By using two levels auditing data is provided to the end user. At last audit report will send to the data owner.

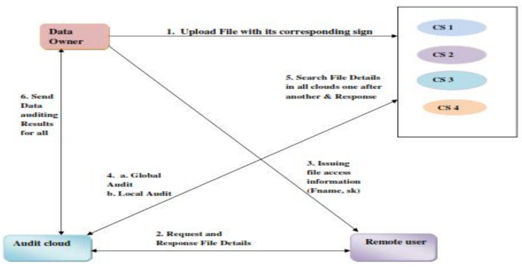
Data Owner
1. Upload File with its corresponding sign
CS 1
CS 2
CS 3
CS 4
5. Search File Details in all clouds one after another & Response
6. Send Data auditing Results for all
4. a. Global Audit b. Local Audit
3. Issuing file access information (Fname, sk)
Audit cloud
2. Request and Response File Details
Remote user

Fig 2: Architecture of the CaaS Model

As shown in **Fig 3** a use case is a methodology used in system analysis to identify, clarify, and organize system requirements. The use case is made up of a set of possible sequences of interactions between systems and users in a particular environment and related to a particular goal. It consists of a group of elements (for example, classes and interfaces) that can be used together in a way that will have an effect larger than the sum of the separate elements combined. The use case should contain all system activities that have significance to the users.

Initially data owner will browse the data and upload the data to the cloud server. In cloud server there are four cloud systems will be there, they are not dependent. At the same time data owner will send Meta data to the audit cloud, and verify the audit report whether the data will be safe or not.

Audit cloud act as an intermediate between cloud server and end user, there are two types of auditing, they are global auditing and local auditing.

End user initially request for the write or read permission from the data owner. End user will request the data to the audit cloud. He will perform either global auditing or local auditing then send the data to the end user.

If the data is corrupted then audit will send confirmation message to the data owner, if it is not corrupted he will send file content to the end user.

Cloud server will store the data, view the stored data. And it as capable of modifying the data. Within the cloud server there are so many independent clouds will be there. Within the cloud server a small audit cloud will be elected

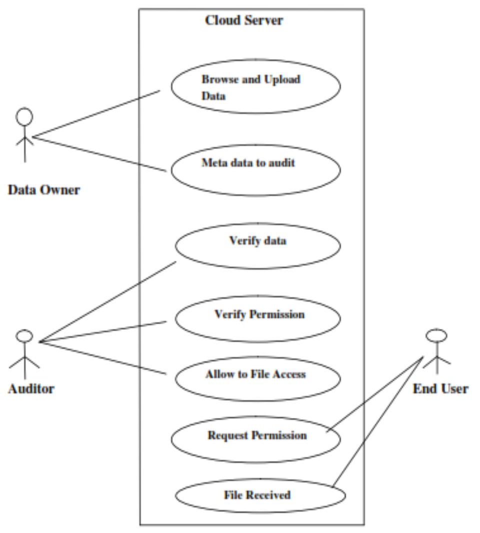

Fig 3: Use Case Diagram

- As shown in **Fig 4** in **level 0** initially data owner browse the data and upload that data to the cloud server and also it will send Meta data to the audit cloud for the verification purpose.
- In **level 1** Receiver is nothing but an end user, first he will request for the write or read permission from the audit cloud, and then only he is permitted to access the data from cloud server. If end user violates the permission then audit cloud will send message to the data owner.

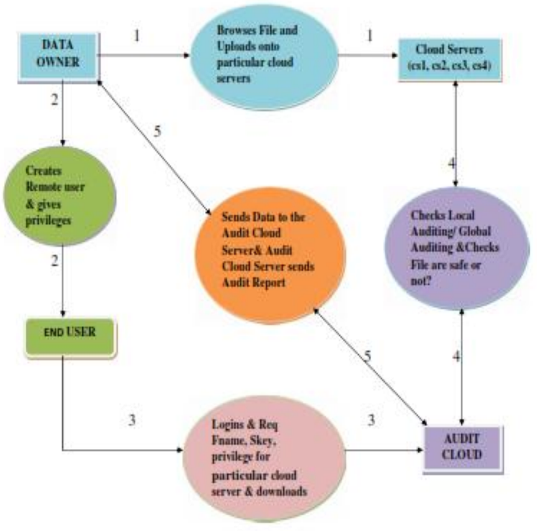
DATA OWNER
1
Browses File and Uploads onto particular cloud servers
1
Cloud Servers (cs1, cs2, cs3, cs4)
2
5
4
Creates Remote user & gives privileges
Sends Data to the Audit Cloud Server& Audit Cloud Server sends Audit Report
Checks Local Auditing/ Global Auditing &Checks File are safe or not?
2
END USER
5
4
3
Logins & Req Fname, Skey, privilege for particular cloud server & downloads
3
AUDIT CLOUD

Fig 4: Data Flow Diagram

Audit cloud act as an intermediate he will send data from cloud server to end user. Before sending a data he will verify the content, if it is true then only end user permitted to access the data, otherwise audit will send message to the data owner.

In level 2 there are two types of audit cloud; they are local auditing and global auditing. He is responsible to verify all

ISSN (Online) 2321 – 2004
ISSN (Print) 2321 – 5526

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH IN ELECTRICAL, ELECTRONICS, INSTRUMENTATION AND CONTROL ENGINEERING

National Conference on Advanced Innovation in Engineering and Technology (NCAIET-2015)

Alva's Institute of Engineering and Technology, Moodbidri

Vol. 3, Special Issue 1, April 2015

the data in cloud server. He is act as an intermediate between cloud server and end user.

If any data is corrupted then new MAC address will be generated. Auditor will compare the MAC address if any changes occur, he will report all the information to the data owner. By using HAS method it reveals as many violations as possible.

In Level 3 Cloud server is able to modify the file content and then process the data. If the same file is requested by the end user, before sending the data audit cloud will verify the data, if it is true then it send message to the end user, otherwise it will send confirmation message as false to the data owner.

VII. CHALLENGES

1) To meet the promise of ubiquitous 24/7 access, the cloud service provider (CSP) stores data replicas on multiple geographically distributed servers. A key problem of using the replication technique in clouds is that it is very expensive to achieve strong consistency on a worldwide scale.
2) If any data is corrupted in cloud server, it takes long time to get recover.
3) Maintaining the each and every cloud with latest updates is very crude task, it should happen quickly before the end user will retrieve the old updates.
4) Satisfy the end user requirements is also very challenging task.

VIII. APPLICATIONS

1) Rack space cloud deployment services.
2) Soft layer cloud services.
3) Micro-soft azure.
4) The domain name system (DNS) is one of the most popular applications that implement eventual consistency.

REFERENCES

[1]. D. Bermbach and S. Tai, "Eventual consistency: how soon is eventual?" in Proc. 2011 MW4SOC.
[2]. H. Wada, A. Fekete, L. Zhao, K. Lee, and A. Liu, "Data consistency properties and the trade-offs in commercial cloud storages: the consumers' perspective," in Proc. 2011 CIDR.
[3]. S. Esteves, J. Silva, and L. Veiga, "Quality-of-service for consistency of data geo-replication in cloud computing," Euro-Par 2012 ParallelProcessing, vol. 7484, 2012.
[4]. T. Kraska, M. Hentschel, G. Alonso, and D. Kossmann, "Consistency rationing in the cloud: pay only when it matters," in Proc. 2009 VLDB.
[5]. W. Golab, X. Li, and M. Shah, "Analyzing consistency properties for fun and profit," in Proc. 2011 ACM PODC.