IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 2, February 2015.
www.ijiset.com
ISSN 2348 – 7968

A Potential Cloud Service by Auditing Cloud

MalaimariSuganya.P[1], Padma Priya.R[2]

[1]Department of Computer Science and Information Technology
Nadar Saraswathi College of Arts and Science, Theni dt-625531, Tamilnadu, India

[2]Department of Computer Application
Nadar Saraswathi College of Arts and Science, Theni dt-625531, Tamilnadu, India

Abstract

Let assume that each user in the audit cloud is identified by a unique ID. Before outsourcing the job to the data cloud, the audit cloud and the data cloud will engage in a Service Level Agreement (SLA), which stipulates the promised level of consistency that should be provided by the data cloud.In the system, a two-level auditing model is adopted: each user records their operations in a User Operation Table (UOT), which is referred to as a local trace of operations. Local auditing can be performed independently by each user with their own UOT; periodically, an auditor is elected from the audit cloud.

The users will communicate to exchange messages after executing a set of reads or writes, rather than communicating immediately after executing every operation. Once two users finish communicating, a causal relationship on their operations is established.

1. Introduction

Cloud Computing

Cloud computing providing unlimited infrastructure to store and execute customer data and program. As customers you do not need to own the infrastructure, they are merely accessing or renting; they can forego capital expenditure and consume resources as a service, paying instead for what they use.

YOUR PERSONAL CLOUD

Storage space and processing power are the biggest hurdles that mobile devices like smartphones and tablets face today. Cloud computing has come to the forefront as it overcomes these hurdles, enabling you to access and edit your fi les and fi les uploaded by other users without the need for excessive hardware and support. Browser-based email and cloud services such as Grooveshark, Facebook, Google Docs and Youtube have smartphone and tablet extensions, designed specifi cally to run on the lower hardware specifi cations of such devices.Operating systems of today like Android, Ios and Windows 7 are also made compatible with mobile devices and equipped with applications that let you sync and access data across devices via their cloud servers. Apart from storage, developers have also used cloud computing capabilities to develop Software as a Service (SaaS), Platform as a Service (PaaS) and Webtop services to make your computing experience completely mobile.

Software as a service (saas)

Web accessible software enables users to use programs or applications without having to download or install them on their machines. You

ISSN 2348 – 7968

can access data hosted by companies on their cloud servers for almost any task, like word processing, designing or even media editing. This enables users to carry out their work without having to purchase software that can be installed and run only on a single machine. You no longer need to worry about expiration dates, downloading updates or installing software on multiple machines. SaaS reduces the user's expenses by allowing you to 'rent' a service and use only specifi c features rather than buying complete packages of programs, some of which you might not use. For instance, to use Microsoft Word, a user must purchase the Microsoft Offi ce auite, which gives you Word bundled with Excel, Powerpoint,etc. These packages can be fairly expensive.Instead, you can use a SaaS product that allows you to sign up for free or a monthly fee and lets you perform your word processing tasks without any installations on your machine's local drive.

Platform as a service (paas)

PaaS is a form of SaaS that provides development environments as a service. Rather than designing software for the customers to use, companies provide clients with the infrastructure required to design and run software specifi c to their needs. Clients can then use this infrastructure and develop software applications for their use that run on the host's cloud servers. This form of cloud computing gives the clients the freedom to develop software to suit their requirements without having to worry about the hardware. PaaS provides clients with facilities for designing and developing applications, hosting their applications as well as hardware for scalability and storage.

Information Security

- Security related to the information exchanged between different hosts or between hosts and users.
- This issues pertaining to **secure communication**, **authentication**, and issues concerning **single sign on** and **delegation**.
- Secure communication issues include those security concerns that arise during the communication between two entities.
- These include confidentiality and integrity issues. Confidentiality indicates that all data sent by users should be accessible to only "legitimate" receivers, and integrity indicates that all data received should only be sent/modified by "legitimate" senders.
- **Solution:** public key encryption, X.509 certificates, and the Secure Sockets Layer (SSL) enables secure authentication and communication over computer networks.

Cloud Computing has become commercially popular, as it promises to guarantee scalability, elasticity, and high availability at a low cost. Guided by the trend of the everything-as-a-service (XaaS) model, data storages, virtualized infrastructure, virtualized platforms, as well as software and applications are being provided and consumed as services in the cloud. Cloud storage services can be regarded as a typical service in cloud computing, which involves the delivery of data storage as a service, including database-like services and network attached storage, often billed on a utility computing basis, e.g., per gigabyte per month. Examples include Amazon SimpleDB[1] , Microsoft Azure storage, and so on. By using the cloud storage services, the customers can access data stored in a cloud anytime and anywhere using any device, without caring about a large amount of

ISSN 2348 – 7968

capital investment when deploying the underlying hardware infrastructures.

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, where a user is ensured to see the latest updates. Actually, mandated by the CAP principle, many CSPs (e.g., Amazon S3) only ensure weak consistency, such as eventual consistency, for performance and high availability, where a user can read stale data for a period of time. The domain name system (DNS) is one of the most popular applications that implement eventual consistency. Updates to a name will not be visible immediately, but all clients are ensured to see them eventually.

Actually, different applications have different consistency requirements. For example, mail services need monotonicread consistency and read-your-write consistency, but social network services need causal consistency. In cloud storage, consistency not only determines correctness but also the actual cost per transaction. In this paper, we present a novel consistency as a service(CaaS) model for this situation. The CaaS model consists of a largedata cloudand multiple small audit clouds. The data cloud is maintained by a CSP, and an audit cloud consists of a group of users that cooperate on a job, e.g., a document or a project. A service level agreement (SLA) will be engaged between the data cloud and the audit cloud, which will stipulate what level of consistency the data cloud should provide, and how much (monetary or otherwise) will be charged if the data cloud violates the SLA.

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, we allow the users in the audit cloud to verify cloud consistency by analyzing a trace of interactive operations. Unlike their work, we do not require a global clock among all users for total ordering of operations. A loosely synchronized clock is suitable for our solution. Specifically, we require each user to maintain a logical vector for partial ordering of operations, and we adopt a two-level auditing structure: each user can perform local auditing independently with a local trace of operations; periodically, an auditor is elected from the audit cloud to performglobal auditingwith 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. Global auditing focuses on causal consistency, which is performed by constructing a directed graph. If the constructed graph is a directed acyclic graph (DAG), we claim that causal consistency is preserved. We quantify the severity of violations by two metrics for the CaaS model: commonality of violations and staleness of the value of a read.

2. Description of consistency Algorithms

Local Consistency Auditing

Local consistency auditing is an online algorithm (Alg. 1). In Alg. 1, each user will record all of his operations in his UOT. While issuing a read operation, the user will perform local consistency auditing independently. LetR(a)denote a user's current read whose dictating write is W(a), W(b) denote the last write in the UOT, andR(c) denote the last read in the UOT whose dictating write isW(c). Read-your-write consistency is violated ifW(a) happens beforeW(b), and

ISSN 2348 – 7968

monotonic-read consistency is violated if W(a) happens beforeW(c). Note that, from the value of a read, we can know the logical vector and physical vector of its dictating write. Therefore, we can order the dictating writes by their logical vectors.

Algorithm 1Local consistency auditing

Algorithm 1 Local consistency auditing

Initial UOT with $\emptyset$
while issue an operation op **do**
  if $op = W(a)$ **then**
    record $W(a)$ in UOT
  if $op = r(a)$ **then**
    $W(b) \in$ UOT is the last write
    if $W(a) \rightarrow W(b)$ **then**
      Read-your-write consistency is violated
    $R(c) \in$ UOT is the last read
    if $W(a) \rightarrow W(c)$ **then**
      Monotonic-read consistency is violated
    record $r(a)$ in UOT

Global Consistency Auditing

Global consistency auditing is an offline algorithm (Alg. 2). Periodically, an auditor will be elected from the audit cloud to perform global consistency auditing. In this case, all other users will send their UOTs to the auditor for obtaining a global trace of operations. After executing global auditing, the auditor will send auditing results as well as its vectors to all other users. Given the auditor's vectors, each user will know other users' latest clocks up to global auditing.

Algorithm 2 Global consistency auditing

Each operation in the global trace is denoted by a vertex
for any two operations op_1 and op_2 **do**
  if $op_1 \rightarrow op_2$ **then**
    A time edge is added from op_1 to op_2
  if $op_1 = W(a)$, $op_2 = R(a)$, and two operations com
  from different users **then**
    A data edge is added from op_1 to op_2
  if $op_1 = W(a)$, $op_2 = W(b)$, two operations come fro
  different users, and $W(a)$ is on the route from $W(b)$ t
  $R(b)$ **then**
    A causal edge is added from op_1 to op_2
Check whether the graph is a DAG by topological sorting

3. Proposed work

We assume that each user in the audit cloud is identified by a unique ID. Before outsourcing the job to the data cloud, the audit cloud and the data cloud will engage in a Service Level Agreement (SLA), which stipulates the promised level of consistency that should be provided by the data cloud.

A service level agreement (SLA) will be engaged between the data cloud and the audit cloud, which will stipulate what level of consistency the data cloud should provide, and how much (monetary or otherwise) will be charged if the data cloud violates the SLA.

In our system, a two-level auditing model is adopted: each user records his operations in a User Operation Table (UOT), which is referred to as a local trace of operations. Local auditing can be performed independently by each user with his own UOT; periodically, an auditor is elected from the audit cloud.

The users will communicate to exchange messages after executing a set of reads or writes, rather than communicating immediately after executing every operation. Once two users finish communicating, a causal relationship on their operations is established.

One of the important concerns that need to be addressed is to assure the customer of the integrity i.e. correctness of his data in the cloud. As the data is physically not accessible to the user the cloud should provide a way for the user to check if the integrity of his data is maintained or is compromised. In this paper we provide a scheme which gives a proof of data integrity in the cloud which the customer can employ to check the

ISSN 2348 – 7968

correctness of his data in the cloud. This proof can be agreed upon by both the cloud and the customer and can be incorporated in the Service Level Agreement (SLA). It is important to note that our proof of data integrity protocol just checks the integrity of data i.e. if the data has been illegally modified or deleted.

We propose a two-level auditing architecture, which only requires a loosely synchronized clock in the audit cloud. Then, we design algorithms to quantify the severity of violations with two metrics: the commonality of violations, and the staleness of the value of a read. Finally, we devise a heuristic auditing strategy (HAS) to reveal as many violations as possible. Extensive experiments were performed using a combination of simulations and real cloud deployments to validate SLA.

Finally, we propose a Heuristic Auditing Strategy *(HAS*) which adds appropriate reads to reveal as many violations as possible.

Our key contributions are as follows:

1) We present a novel consistency as a service (CaaS) model, where a group of users that constitute an audit cloud can verify whether the data cloud provides the promised level of consistency or not.

2) We propose a two-level auditing structure, which only requires a loosely synchronized clock for ordering operations in an audit cloud.

3) We design algorithms to quantify the severity of violations with different metrics.

4) We devise a heuristic auditing strategy (HAS) to reveal as many violations as possible. Extensive experiments were performed using a combination of simulations and real cloud deployments to validate HAS.

User: In this module, user should register their details and get the secret key for login and user can upload the file regarding the auditing.

Auditor: In this module, auditor can do the auditing based on the heuristic auditing strategy. It relates with document verification. Auditor can check the auditing file he can neglate or accept the file he can revise the report and check whether its good or bad and auditor can give revision report like accept or waiting. If status in accept means user can view the file else status is waiting means user cant view the file.

Admin: In this module admin can view all the user details, user uploads details, and TPA activities regarding the auditing strategy.

Data upload module (in cloud database): In this module, the user uploaded files can be stored in cloud database... it can be very secure auditor can view the file from the database it can be very secure.

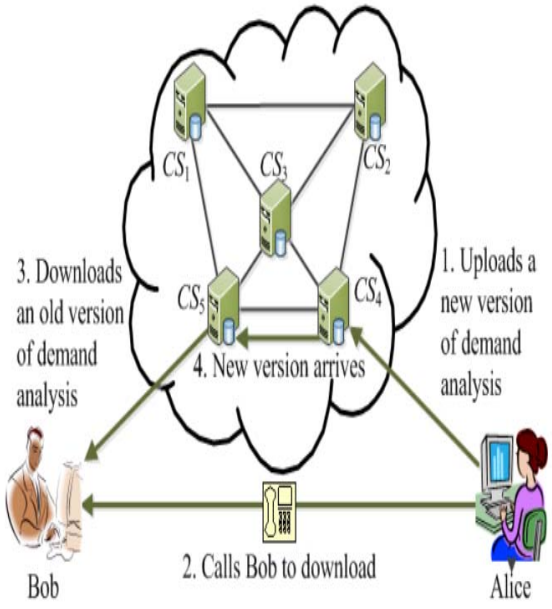

Fig 1-Auditing Cloud

ISSN 2348 – 7968

4. Conclusion and Future Enhancement

In this paper, we presented a consistency as a service (CaaS) model and a two-level auditing structure to help users verify whether the cloud service provider (CSP) is providing the promised consistency, and to quantify the severity of the violations, if any. With the CaaS model, the users can assess the quality of cloud services and choose a right CSP among various candidates, e.g, the least expensive one that still provides adequate consistency for the users' applications. For our future work, we will conduct a thorough theoretical study of consistency models in cloud computing.

5. References

[1] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica,et al., "A view of cloud computing,"Commun. ACM, vol. 53, no. 4, 2010.

[2] P. Mell and T. Grance, "The NIST definition of cloud computing (draft)," NIST Special Publication 800-145 (Draft), 2011.

[3] E. Brewer, "Towards robust distributed systems," in Proc. 2000 ACM PODC.

[4] ——, "Pushing the CAP: strategies for consistency and availability," Computer, vol. 45, no. 2, 2012.

[5] M. Ahamad, G. Neiger, J. Burns, P. Kohli, and P. Hutto, "Causal memory: definitions, implementation, and programming,"Distributed Computing, vol. 9, no. 1, 1995.

[6] W. Lloyd, M. Freedman, M. Kaminsky, and D. Andersen, "Don't settle for eventual: scalable causal consistency for wide-area storage with COPS," inProc. 2011 ACM SOSP.

[7] E. Anderson, X. Li, M. Shah, J. Tucek, and J. Wylie, "What consistency does your key-value store actually provide," in Proc. 2010 USENIX HotDep.

[8] C. Fidge, "Timestamps in message-passing systems that preserve the partial ordering," inProc. 1988 ACSC.

[9] W. Golab, X. Li, and M. Shah, "Analyzing consistency properties for fun and profit," in Proc. 2011 ACM PODC.

[10] A. Tanenbaum and M. Van Steen, Distributed Systems: Principles and Paradigms. Prentice Hall PTR, 2002.

[11] W. Vogels, "Data access patterns in the Amazon.com technology platform," in Proc. 2007 VLDB.

[12] M. Brantner, D. Florescu, D. Graf, D. Kossmann, and T. Kraska, "Building a database on S3," inProc. 2008 ACM SIGMOD.