ISSN 2395-1621

AN INNOVATIVE CONSISTENCY MODEL FOR CLOUD SERVICES

[#1]Sagar Kasar, [#2]Nisarg Jayvant, [#3]Shubham Dhake, [#4]Monika Jagtap,
[#5]Prof.Madhavi Darokar

[1]kasar.sagar3@gmail.com
[2]nisargjayvant@gmail.com
[3]shubhamdhake9028@gmail.com
[4]monikajagtap555@gmail.com

[#1234]Student, Department of Computer
[#5]Professor, Department of Computer
JSPM's ICOER, Pune, India - 412207.

ABSTRACT

Cloud computing storage service popularity has increased due to its great benefits. Cloud computing is the most important part of our life. Cloud is most popular because of huge advantages such as it is portable we can able to access the cloud anywhere globally and it is used in different business purpose. Most of the cloud service provider provides services like infrastructure management, data storage services on 24/7 through any devices at anywhere. To provide this ubiquitous always on service most of the cloud service provider (CSP) maintains each piece of data on geographically distributed servers. The main key problem with this technique is that, it is very expensive and some to fail to provide required consistency of service. To overcome this problem, we propose to use a new approach of service (i.e. Consistency as a Service (CaaS) this paper, firstly concentrate on a consistency as a service (CaaS) model, which has a large data cloud and multiple small audit clouds. In the CaaS model, a data cloud is formed by a CSP, and a group of users form an audit cloud that can verify whether the data cloud provides the promised level of consistency i.e. quality of service or not, for that make use of two- level auditing strategy which require loosely synchronized clock for ordering operations in an audit cloud. Then perform global auditing by global trace of operations through randomly electing an auditor from an audit cloud. Finally, use a heuristic auditing strategy (HAS) to display as many violations as possible.

Keywords: Cloud Storage, consistency as a service(Caas), two-level auditing, Global Consistency Auditing, Local Consistency Auditing, heuristic Auditing Strategy (HAS).

ARTICLE INFO

Article History

Received: 25th May 2017

Received in revised form : 25th May 2017

Accepted: 27th May 2017

Published online : 30th May 2017

I. INTRODUCTION

Now a day's Cloud computing has become more popular, as it provide advantages like security, scalability, elasticity and high availability at lower cost [6][8] .Cloud storage service has become more accepted due to their very great advantages .Cloud service providers retain many replicas for every piece of data on physically distributed server. Replication method is used to improve performance and increase reliability .Replica it allows remote sites to go on working in the event of local failure. To maintain continuous accessibility the file is replicated at many different places in cloud so even if one of the site is down still you can retrieve the data from another place .Cloud storage services which involves the transfer of data storage as a service including data base like services and NAS(network attach storage)frequently billed on service computing basis. Example Amazon simple database it is non relational data store. Support store and query function usually provided only by relational database and it also hold to increase performance web application. User can store and query data item by means of web service request it manages manually the infrastructure provisioning and hardware, software maintenance, replication and indexing of data items. By use the Cloud storage services, the clients can access data stored in a cloud anytime and anywhere using any device, without any capital investment when they are deploying the underlying hardware infrastructures.

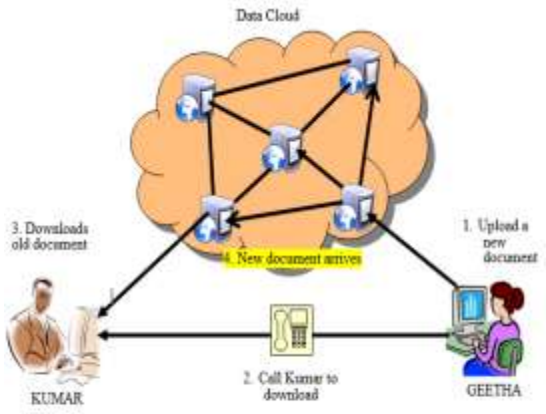

Fig 1. System Architecture.

Fig 1. Shows system architecture. In this there are two users Madhuri and Nikita they both are working on Project using a Cloud Storage Service. Data cloud that has data that data is replicated at many places. Data is replicated to five cloud server that is CS1,CS2,CS3,CS4,CS5 .After uploading a new document to CS4 ,Nikita calls Madhuri to download the latest version for integrated design .Here ,after Nikita calls Madhuri, the causal relationship is accepted between Nikita's update and Madhuri's read . Therefore, the cloud should give causal consistency, which ensures that Geetha's update is committed to all of the replicas before Kumar's read. If the cloud provides only eventual consistency , then Kumar is allowed to access an old version of the requirement analysis from CS5 .In this case ,the integrated design that is based on an old version may not satisfy the real requirements of customers. Different applications have different consistency requirements. The consistency properly ensures that any transaction will bring the database from one valid state to another. Any data written to database must be valid according to all defined rules, including constraints, cascades, triggers and any combination thereof. Different application have different consistency requirement .For example mail services require both monotonic-read consistency and read-your-write consistency and social network services need casual consistency [7] .In cloud storage, consistency not only determines accuracy but also the real cost per transaction. A novel consistency as a service (CaaS) model consists of big data cloud and several small audit clouds. The implementation of the data cloud is not clear to all users due to the virtualization technique. So it is very difficult for users to verify whether each replica in the data cloud is the latest one or not. 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 casual consistency which is performed by constructing a directed graph.

II. LITERATURE SURVEY

It present a novel approach to benchmark staleness in distributed datastores and make use of the approach to assess Amazon's Simple Storage Service (S3) [4] There are two main classes of consistency :data-centric and client-centric consistency. Data-centric consistency model generally focus on the internal state of the storage system that is consistency have been reached as soon as all replica of given data item are the same .How updates flow through the system and what guarantees the system be able to provide with respect to update. Here in this customer does not matter whether or not a storage system internally contains any stale copies. There is no stale data is observed from the client point of view the customer is satisfied. In a client-centric consistency model focus on what specific customer wants that is how the customer observe data update. It was describes different level of consistency in distributed system, from strict consistency to weak Consistency. High consistency implies high cost and reduced availability .Client-centric consistency model they do not care about the internal state of a storage system. They explained how these two communicate to each other and introduced an approach which allows computing the staleness of data, or how soon 'eventual' in eventual consistency is.

A.Don't settle for eventual: scalable causal consistency for wide-area storage with COPS:

Geo-replicated, distributed data stores that support complex online applications, such as social networks, must provide an "always-on" experience where operations always complete with low latency. Today's systems often sacrifice strong consistency to get these goals, exposing inconsistencies to their clients and necessitating complex application logic.

This system presentimplementation and the design of COPS, a key-value store that delivers this consistency model across the wide-area and a key contribution of COPS is its scalability, which can enforce causal dependencies between keys stored across an entire cluster, rather than a single server like as previous systems. Furthermore, in COPS- GT, this system introduces get transactions in order to obtain a consistent view of multiple keys without locking or blocking. The assessment shows that COPS completes operations in less than a millisecond, provides throughput similar to previous systems when using one server per cluster, and scales well as this system increase the number of servers in each cluster. It also shows that COPS-GT facilitates similar latency, throughput, and scaling to COPS for common workloads.

***B.* Axioms for memory access in asynchronous hardware systems:**

Misra [2] is the first to present an algorithm for verifying whether the trace on a read/write register is atomic. Following his work, Ref. [3] proposed offline algorithms for verifying whether a key-value storage system has safety, regularity, and atomicity properties by constructing a directed graph. He presented an elegant algorithm for checking atomicity. Misra's algorithm works by reasoning about the values of the register. The observation is that, at some point during the span of an operation, the register assumes the value of the operation, (either write or read). Atomicity convenants that if a value is replaced by another value, then the old value is not allowed to re-appear in the future. Therefore, if a trace violates this situation, then it is not atomic. Some- what surprisingly, if a trace does not violate this condition, then it is atomic. In contrast, our algorithms reason about the operations. We choose to reason about operations but not values because we aim to provide a common framework to check a variety of semantics, many of which (e.g., regularity and safety) were introduced after Misra's paper. It's notinstantly clear to us how to extend Misra's algorithm to check, say, regularity, because for regularity, a replaced value is allowed to re-appear.

C. Two Level Auditing Architecture to Maintain Consistent In Cloud:

Secret and confidential data in an enterprise may be illegally accessed through a remote interface facilitated by a multiple cloud, or relevant data and archives may be lost or tampered with when they are stored into an uncertain storage pool outside the enterprise. To overcome these problems this system presents a Consistency as a service auditing cloud scheme. This system proves the security of my scheme based data fragmentation on multiple clouds. So that, the proposed system has data fragmentation, data security and storage on multiple cloud services. This system used trusted third party to store the data on multiple cloud and find the data access by un-trusted cloud service providers. In this system the client data divided into multiple pieces and send to the multiple clouds with help of trusted or believable third party. If any of the un-trusted cloud service providers try modify the data the alert will send to trusted third party about illegal access of un-trusted cloud service provider.

***D.* What consistency does your key-value store actuallyprovide:**

A different approach to calculating consistency of cloud storageplatforms is taken by Anderson et al [3], where theyrecord lengthy traces with interleaved operations, and afterthe fact they check for cycles in several conflict graphs tofind whether various properties hold. The propertiesthey analyze are those that are important in parallel hardware design, such as safe registers or regular, rather than theproperties usual in cloud storage platforms such as eventualconsistency with monotonic reads.There is also work on formally defining weak consistencyproperties. Usually eventual consistency is described in termsof internal properties such as the state of the replicas.

***E.* Analyzing Consistency Properties for Fun and Profit:**

This system address two important problems related to the consistency properties in a history of operations on a write/read register (i.e., finish time,start time, argument, and response of every operation). First, this system considers how to detect a consistency violation as soon as one happens. To this end, this system formulates a specification for online verification algorithms, and this system presents such algorithms for several well-known consistency properties. Second, this system considers how to quantify the severity of the violations, if a history is found to contain consistency violations. This system investigates two quantities: first is the staleness of the reads, and the second is the commonality of violations. For staleness, this system further considers time-based staleness and operations-count-based staleness. These system present efficient algorithms that compute these quantities. This system have faith that addressing these problems helps both key-value store providers and users adopt data consistency as an important aspect of key-value store offering.

Proposed System:

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

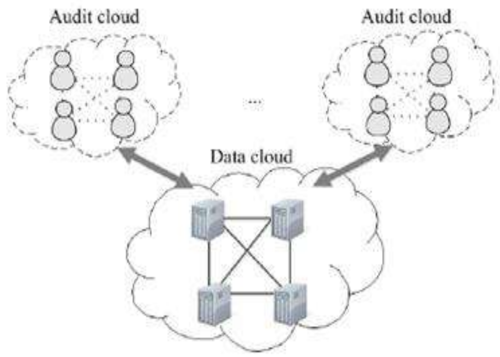

Figure.2 Proposed model

The Consistency as a service model consists of large data cloud and various audit cloud .A service level agreement (SLA) will be busy between the data cloud and the audit cloud ,which will tell what level of consistency the data cloud must provide, and how much will be charged if the data cloud violates the service level agreement .

In User Operation Table Each client maintains a User Operation Table for recording local operations. Each record in the User Operation Table is described by three elements: operation, logical vector, and physical vector. While doing an operation, a client will record this operation, as well as his current logical vector and physical vector, in his UOT.

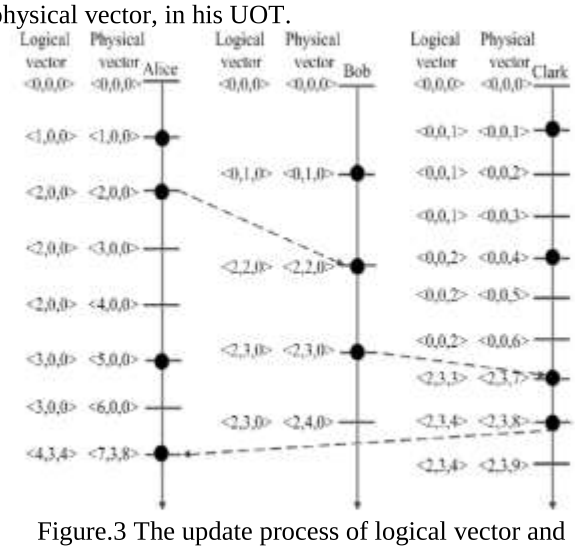

Figure.3 The update process of logical vector and physical vector.

A dark black circle shows an event (write/send/read/ message/receive message), and the arrows from top to bottom denote the increase of physical time. The physical vector is updated in the similar way as the logical vector, except that the user's physical clock keeps growing as time passes, no matter whether an event (read/write/send message/receive message) happens or not. The update process is as follows: All clocks are initialized with zero (for 2 vectors); The user increases his own physical clock in the physical vector continuously, and increases his own logical clock in the logical vector by one only when an event happens; Two vectors will be sent along with the message being sent. When a user receives a message, he updates each element in his vector with the maximum of the value in his own vector and the value in the received vector (for two vectors).Each user will maintain a logical vector and a physical vector to track the logical and physical time when an operation happens, resepectively. A two-level auditing structure, which only requires a loosely synchronized clock for ordering operations in an audit cloud. Here each client has to support a logical vector for limited ordering of operation and implement, a twolevelauditing structure. Each client perform local auditing separately with a local trace of operation; periodically an auditor is selected from the audit cloud to perform global auditing with global trace of operations.
At the first level, each client on your own perform local auditing with his own User operation table . In local consistency has two types Monotonic-read-consistency and Read-your-write consistency:

1) Monotonic-read-consistency:- If a process reads the price of data K , any consecutive read on data K by that process will always return that same value or extra recent value .
2) Read-your-write consistency:-The result of a write by a process on data K will always be a consecutive read on data K by the similar process.

Monotonic-read consistency requires that a user should read either a new value or the identical value, and read-your-write consistency need that a user all the time read his latest updates At the second level, an auditor be able to execute global auditing after obtaining a global trace of all users operations, global auditing concentrate on Causal consistency which can be performed by offline algorithm.

1) Causal consistency:- Writes that are causally related should be seen by all processes in the similar order. Concurrent writes may be seen in a dissimilar order on different machines.

Global auditing concentrate on casual consistency, which is performed by constructing a directed graph . If the constructed graph is a directed acyclic graph (DAG) then casual consistency is preserved. Specify the severity of violations can be done by two metrics for the CaaS model: commonality of violations and staleness of the value of read. Finally it was propose a heuristic auditing strategy (HAS) which adds appropriate reads to reveal as several violations as possible.

Mathematical model:

System Description:

User Authentication
Set (C) = {c0, c1, c2, c3}
C0= Get User Id
C1=Get Cloud Id.
C2=Get Data Owner Info
C3=get the User Privilege Information
C4= Get key from hash table.

Encryption
Set (E) = {e0, e1, e2, c1, c2}
e0=get file to be encrypted
e1=get public key for encryption
e2=encryption of data

Consistency check
Set (D) = {d0, d1, d2}
d0= check global status
d1= maintain consistency status
d2= check out

Service module
Set (S) = {s0, s1, s2, d0, d1}
s0=get user id and file request
s1=get data to be uploaded or downloaded
s2=provide service

Union and Intersection of project
Set (E) = {e0, e1, e2, c1, c2}
Set (D) = {d0, d1, d2, d3}
Set (S) = {s0, s1, s2, d0, d1}
(C U E)= {c0,c1,c2,c3,c4, e0, e1, e2}
(C intersection E)= {c1, c2}
(D U S)= {d0, d1, d2, d3, s0, s1, s2}
(T intersection S) = {d0, d1}

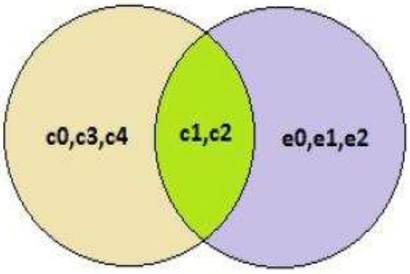

Figure 1.2: C intersection E

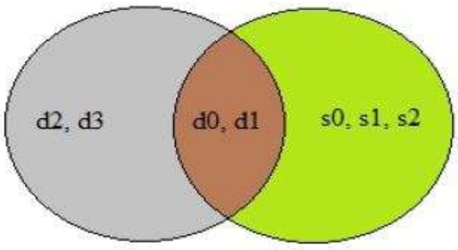

Figure 1.3: D intersection S

III. RESULTS

System results are as follows:

Figure4: Registration

Figure5: User Activation

Figure6: Toke generation

Figure7: Cloud user login

Figure9: User login

Figure10: Authentication

Figure11: File Uploading

Figure8: File Uploaded successfully

IV. CONCLUSION

Consistency as a service (CaaS) model and a two-level auditing structure to help users validate whether the cloud service provider (CSP) is providing the promised consistency and to quantify the severity of the violations is any . With the CaaS model, the users can assess the quality of cloud services and select a right cloud service provider among various candidates, for example the least expensive one that still provides adequate consistency for the users' application. In future work will determine dependencies between files on S3. The plan to publish these results such as bechmarking Apache Cassandra and the Google App Engine data store to extend our efforts to additional storage system. Future work, it will conduct a thorough theoretical study of consistency models in cloud computing.

REFERENCES

[1] Qin Liu,Guojun Wang,Jie Wu, "Consistency as a Service: Auditing Cloud Consistency," 2014.

[2] M.Rahman, W.Golab, A.AuYoung, K.Keeton, and J.Wylie, "Toward a principled framework for benchmarking consistency," 2012.

[3] 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," 2014.
[4] D.Bermbach and S.Tai, "Eventual consistency: how soon is eventual?" 2011.
[5] E.Anderson, X.Li, M.Shah, J.Tucek, and J.Wylie, "What consistency does your key-value store actually provide," 2010.
[6] 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 DFISAAECAcomputing," Commun. ACM, vol. 53, no. 4, 2010.
[7] W.Lloyd, M.Freedman, M.Kaminsky, and D. Andersen, "Don't settle for eventual: scalable causal consistency for wide-area storage with COPS," in Proc. 2011 ACM SOSP.
[8] P.Mell and T.Grance, "The NIST definition of cloud computing (draft),"NIST Special Publication 800-145 (Draft), 2011.
[9] S.Esteves, J.Silva, and L.Veiga, "Quality-of-service for consistency of data geo-replication in cloud computing," 2012.
[10] T.Kraska, M.Hentschel, G.Alonso, and D. Kossmann, "Consistency rationing in the cloud: pay only when it matters," in Proc. 2009.

