Implementing CaaS Model for Cloud Consistency

V.S.Padmini

Asst. Professor

Department of Computer Science & Engineering

G.N.D.E.C, Bidar, Karnataka, India.

***Abstract*— Cloud storage services are commercially more popular due to their amount of advantages. Cloud storage services with larger number of benefits have become commercially popular now- a- days. And they provide services like data storage services, and infrastructure management 24/7 through any device and from anywhere. The cloud service provider popularly known as CSP provides generally ubiquitous always-on-service by maintaining a single piece of data on different servers, which are geographically located in different places. The problem is that it is very expensive and failed to provide highly required consistency of service. Hence to overcome this issue, we going to propose a new fresh approach of service that is Consistency as as Service know as CaaS. The Consistency as a Service (CaaS) model concentrates on; in this we have large data cloud and small multiple audit clouds. Now firstly in the CaaS model, the main data cloud is created by a CSP, and a small number of group of users form an audit cloud part that can check whether the data cloud assures the guaranteed level of consistency that is whether it provides quality of service or not. To perform such operation on cloud we are going for two-level auditing strategy which makes use of loosely synchronized clock for calling 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. Then randomly choosing an auditor from an audit cloud to perform global auditing operations i.e. to perform global trace of operations. And then finally, making use of Heuristic auding strategy (HAS), which display the possible violations.**

***Keywords*— Cloud storage, consistency as a service (CaaS), two-level auditing, and heuristic auditing strategy (HAS).**

I. INTRODUCTION

Clouds computing has become more popular choice, because it has succeeded in giving guaranteed basic services like virtualized infrastructure system and providing data storage, etc. e.g. Amazon, SimpleDB are example of such systems. The customers or end users by making use of these services, become authorized users and able to access the data from anywhere and at any time using any device and getting confidence that the capital investment is going to less. The cloud service provider popularly known as CSP promising the users data is going to be available as 24/7, and they can access it efficiently. The CSP stores the different copies of data in a distributed fashion on different servers, which geographically present in different places. The main issue with distributing multiple copies of data called as replication technique is resultant into a very expensive process to provide strong consistency operation. In the coming days user is assured to see the latest updates about this service or operation.

Many cloud service providers provide week consistency, we call such consistency as eventual consistency, where a user can read the data for particular time. Now-a-days stronger consistency assurance is getting importance. Consider the following figure.

In the above figure data is stored in multiple copies on five cloud servers (CS1, CS2…, CS5), users specified in the figure share data through a cloud storage service. Here the cloud should provide casual consistency service, where a user Alice uploads a data on the cloud server CS4. Here the user update should be reflected in all the servers. If cloud service provider provides only eventual consistency then receiver user is going to receive the old version of data. Such a integrated design based on traditional version may not satisfy customer requirements.

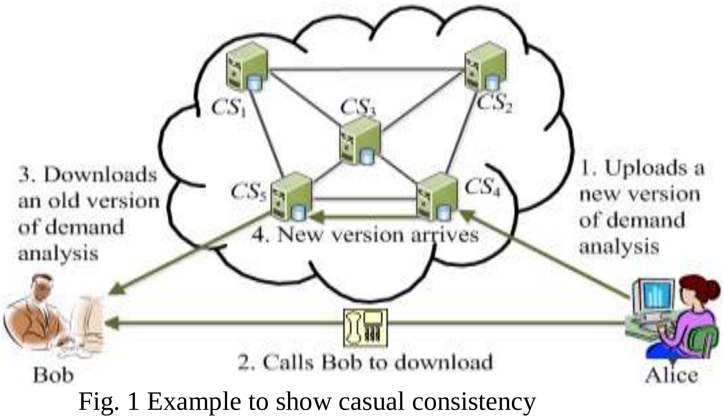
CS_1
CS_2
CS_3
CS_4
CS_5
3. Downloads an old version of demand analysis
1. Uploads a new version of demand analysis
4. New version arrives
2. Calls Bob to download
Bob
Alice

Fig. 1 Example to show casual consistency

Hence we can conclude that different applications need different level of consistency operation. In this work, we propose CaaS model as a ideal consistency operation, which the applications of today's date are demanding. The standard CaaS model consists of large cloud data created by CSP and complete the operation scheme it contains many audit clouds which are formed by authorized group of users. These group of users working on a project and making a document, which constantly checking whether data cloud offers a guaranteed level of consistency or not. This standard model incorporating two-level auditing structure, which involves a synchronized clock assigning tasks to audit cloud and then performing global auditing with global trace of standard operations periodically by auditor chosen from an audit cloud. Local auditing is going to be performed and focuses on continuous read and read-your write process consistencies, which is going to be done online by a light-weight algorithm, while global auditing operation focuses on casual consistency, where it constructs a directed graph. Whatever graph has been constructed, if it directed acyclic graph, which is also called as precedence graph, we can say that casual consistency is maintained. We confirm the severity of violations by calculated two metrics for the standard CaaS model: One is called commonality of violations and other is staleness of value of read operation. Finally in this work, we propose a approach called Heuristic Auditing Strategy(HAS), which proves cloud consistency and required cost i.e. actual cost per transaction.

The two level auditing structure basically contain 2 auditing
1. Local Auditing
2. Global Auditing

Local Auditing: structure each user can perform local auditing with local trace operation periodically .this auditing focuses on monotonic read and read your write consistency .which can be perform by light-weight online algorithm the local auditing algorithm is online algorithm.

Global Auditing: the auditor can be selected from audit cloud .the main works of the auditor is to perform global auditing with global trace operation .this auditing focuses on causal consistency because causal consistency perform by constructing directed graph .the directed acyclic graph is constructed then causal consistency is obtain .Finally we propose analytical auditing strategy which appropriate reads to reveal many unsuccessful result.

We illustrate the consistency service model. Then, we describe the structure of the user operation table (UOT), with which each user records his operations. Finally, we provide a two-level auditing structure and related definitions.

A) **Consistency Service Model:** Consistency service model contain data cloud and multiple audit cloud as shown in fig3

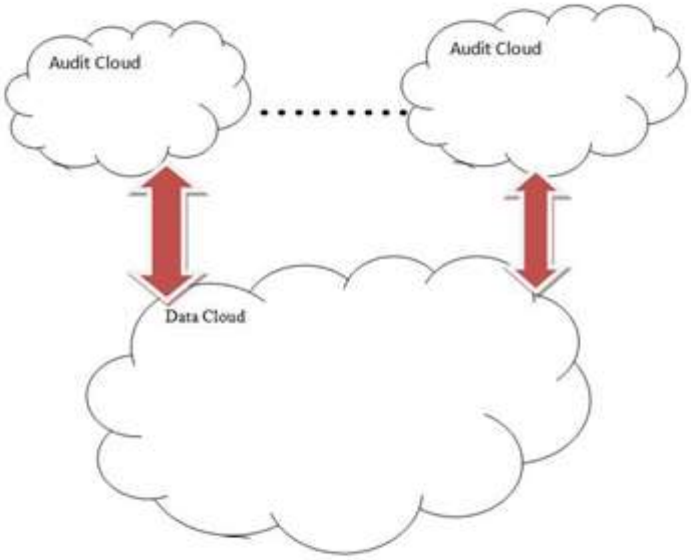

Figure 2 Consistency as a service model.

As shown in figure 2 the Cloud Service Provider maintain Data cloud .data cloud is key value data storage system hence unique key is assign to each piece of data ,cloud service provider maintain data cloud and audit cloud contain a group of users that working on that project And service level Agreement will be form between audit cloud and data cloud .which will decide how much will be charged if the data cloud failed to SLA and what type of consistency the data cloud should provide .the implementation of data cloud is not visible to all user due to virtualization technique. it is very difficult for user to check whether each replica in data cloud is newest one or not . we permit the user in audit cloud to check cloud consistency by analyzing the trace interactive operation .we don't require a global clock among all user for total ordering of operation so we use loosely synchronized clock for our solution. For partial order of operation each user maintain logical vector .so here we develop 2 level of auditing Structure .The two level auditing structure basically contain 2 auditing.

B) User Operation Table(UOT) : Every user maintains a User Operation Table to record logical operation elements logical vector ,physical vector as well as operation are inserted into user operation table. Every operation has write operation or read operation .let us consider operation as op , write W (K, a) ,read R(K, a) . where W(K , a) is nothing but writing the value a to data which is identified by key K. R(K, a) stands reading data which is identified by key K and whose value is a. let us consider W (K, a) as R(K, a)'s dictating write, and R(K, a) as W (K, a)'s dictated read. we have the following properties: A read must have a unique dictating write. A write may have either zero or more dictated reads. From the value of a read, we can know the logical and physical vectors of its dictating write. Let there are N users in the audit cloud and A logical per physical vector is a vector of N logical per physical clocks, 1 clock / user, sorted in ascending order of user ID. For a user with IDi where 1 ≤ i ≤ N. logical vector is < LvC1 , LvC2 ,..., LvCN >, where LvCi is logical clock, and LvCj is the latest logical clock of user j to his best knowledge; his physical vector is < Pv C1 ,Pv C2 ,...,Pv CN >, where Pv Ci is his physical clock, and Pv Cj is the latest physical clock of user jto the best of his knowledge. Logical vector is modernize by using vector clock algorithm and physical vector also gets modernize in the similar way as logical vector excluding physical clock rises as time passes . regardless of execution of event . Update process is given below: Initially all clocks are zero for two vector .the users continuously rises his own physical clock in physical vector as sell as rises his one logical clock in logical vector ,by one the moment event take place . two vector will be sent with message ,as soon as user receive message he modernize every elements in the vector with maximum value in his own vector along with value in receive vector .consider there are three user in audit cloud A,B,C respectively where IDA < IDB < I Dc

Each user update vector the details working of vector is shown in the fig 4

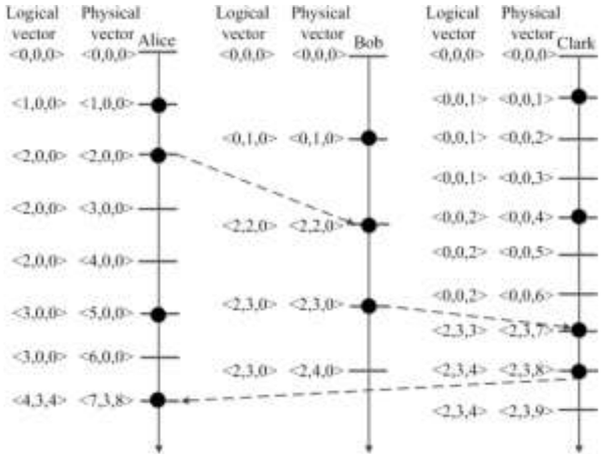

Figure 3: Logical and Physical Vector

As shown in figure 3 as A w(k,a)is <2,0,0><2,00>hence here logical and physical vector.

Following table 1 shows details regarding operations performed on user.

Table 1: User operation table

Alice operation log

Operation	Logical Operator	Physical vector
W(a)	<1,0,0>	<1,0,0>
W(b)	<3,0,0>	<5,0,0>
R(b)	<5,3,5>	<8,3,7>

Bobs Operation log:

Operation	Logical Operator	Physical vector
W(c)	<0,1,0>	<0,1,0>
R(c)	<2,4,0>	<2,5,0>
W(d)	<2,5,0>	<2,6,0>

Clarks Operation log:

Operation	Logical Operator	Physical vector
W(c)	<0,1,0>	<0,1,0>
R(c)	<2,4,0>	<2,5,0>
W(d)	<2,5,0>	<2,6,0>

General review of Two Level Auditing Structure

In this part local consistency is verified .every user perform local auditing separately with his own user operation table

Here we discuss three consistencies

- Monotonic read Consistency
- Read your Write consistency
- Causal Consistency

Monotonic read Consistency:

If any process read the value of data X as well as successive read on data X then same value or more recent value is obtain.

Read your Write Consistency:

If write of process on data X will be seen by successive reads on data X by the same process.

Causal Consistency:

Write which are causally related then it must be seen to all processes in the same arrangement Concurrent writes may be seen in different arrangement and different machines.

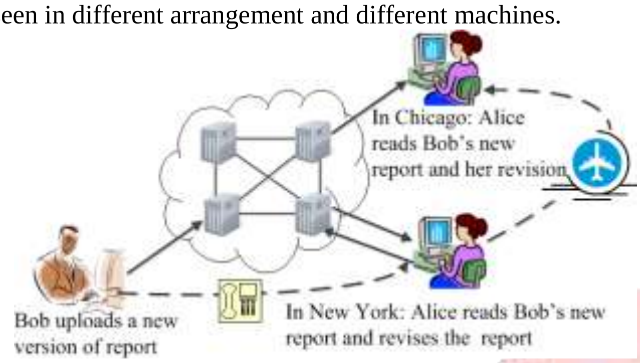
In Chicago: Alice reads Bob's new report and her revision
Bob uploads a new version of report
In New York: Alice reads Bob's new report and revises the report

Figure 4. An application has various consistencies

PROBLEM STATEMENT

The traditional system follows eventual consistency, that is data uploaded on the cloud didn't reflect on the cloud servers dynamically on the cloud system maintained by cloud service provider,where as today's customers requirements need casual consistency.

METHODOLOGY

Different applications over the network have different consistency requirements. For example real time services like mail have monotonic-read consistency. But today popular services like social-network services need casual consistency. In the cloud system consistency not only defines correctness of data but also correct cost per transaction.

Cloud storage services have become very useful due to their best advantages .To provide good access facility, a cloud service provider (CSP) maintains many copies of every on distributed servers. A main problem of using this technique which distributes the multiple copies of data called replication Technique in clouds, it results into a costly process to get information like about the good consistency on a worldwide .We first present a normal consistency as a service (CaaS) model, which has considered as large data cloud and multiple small audit clouds. In the CaaS model, basically a data cloud is going to manage by a CSP, and generally a group of users that form an audit cloud can check whether the data cloud assuring the guaranteed level of consistency service or not. In this work we are going to propose a two-level auditing standard architecture, where synchronized clock is the requirement in the audit cloud. After this, we develop algorithms or methods to check the level of the severity of violations with the presence of basic two metrics: first one is the commonality of violations, and another one is staleness of the calculated value of a read operation. And finally, we formulate a strategy called heuristic auditing strategy (HAS) to provide information about as many violations as possible. Very complex operations were performed using a combination of simulation operations and real cloud operation that is deployments to validate HAS method.

OBJECTIVE OF PROJECT

The objective of the project is to implement a Consistency-as-a-service(CaaS) model which gives solutions to the above said problems.

LITERATURE SURVEY

"Quality-of-service for consistency of data geo-replication in cloud computing,"

AUTHORS: S. Esteves, J. Silva, and L. Veiga

Now-a-days we are dependent on the critical data information, which is stored in cloud data centers across the globe, and getting increased in great way. And we are using different replication methods or approaches to deliver high-availability of services, demand of high performance, and mainly to maintain the consistency among multiple copies of data i.e. replicas. And the performance is degraded gradually to a great extent with traditional consistency techniques, and due to which highly scalable data centers provide lower latency services to end-users, with sacrificing some proportion of consistency in the services. And the cloud system make the stale data to be present in the system, without updating it dynamically, allowing the users to access such unwanted data, which results into a ambiguity. In such cases the requirement of consistency methods is going to be high. If possible to we can combine the stronger and weaker consistency levels. Hence to tackle such in intrinsic and well studies process with trade-off between consistency process and availability of data, the authors have proposed use of a method called VFC 3, a unique consistency technique for duplicated data present across different data centers with great support of library and framework to support improvement in the level of consistency for different data based on their meaning. It targets cloud tabular data stores, offering rationalization of resources (especially bandwidth) and improvement of QoS (performance, latency and availability), by providing strong consistency where it matters most and relaxing on less critical classes or items of data. The proposed technique targets or focuses on data stored in tabular format, provides rationalization of resources, here bandwidth means bandwidth and which also requires improvement in the QOS parameters like latency value, performance in the network and availability of resources. The method provides strong consistency service which actually it is required for the operation, thinking less about applications where it is not critical.

“Data consistency properties and the trade-offs in commercial cloud storages: the consumers’ perspective,”
AUTHORS: _H. Wada, A. Fekete, L. Zhao, K. Lee, and A. Liu,
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.
Here the authors have brought a new area of for study, where a new class has been introduced in the data storage system, which is called as NoSQL (Not Only SQL), which have been introduced to support the traditional or general classic database systems, where it come across the removal of ACID transactions properties from these systems, considered as the one common general feature. This method is offering us different platforms to work, primitives with different features with one package of NoSQL platform, and which is offering different consistency properties. In this work, the authors have also considered a discussion about weaker consistency, which should provide benefits, provide lower latency, and provides better availability for the platform provider. In this paper the authors have investigated the consistency and considered performance properties of different applications. The authors have found that many platforms provide consistency in large extent, with the value what they promise in actual practice. And it also provides a platform for the consumers, where they will get options to chose between weaker and stronger consistencies.

“Toward a principled framework for benchmarking consistency,”
AUTHORS: _M. Rahman, W. Golab, A. AuYoung, K. Keeton, and J. Wylie

In this paper, the authors are highlighting the study of storage systems, these systems with large-scale key-value standard storage systems, compromise with consistency for the interest of dependability, i.e. availability of resources and partition tolerance systems, as well as performance of network, considering latency parameter. The system under this study provides eventual consistency, which is difficult to implement in real time systems. Considering the study of eventually-consistent systems, where we are considering their implementations and deployments. Example here to mention is NoSQL systems. The authors have attempted to measure such consistency empirically, but these systems suffer from some drawbacks. But their accuracy has been limited due to some state-of-the are systems considering consistency benchmarks. Where such systems exercise in restricted ways and workload has been distrupted. The authors in this paper, the consistency benchmark should draw a relationship between storage system, the failure patterns, the system workload, and consistency noticed by clients. To prove their point, the authors have done some surveys of previous efforts done to select the eventual consistency. Then the authors have presented a technique called benchmarking technique, worked on the limitations of existing system technique in measuring the consistency noticed by the clients. These measures are observed by the clients while running the workload considering system. The proposed method is going to be used in different applications, satisfy the requirements of end user.

“An evaluation of alternative architectures for transaction processing in the cloud,”
AUTHORS: _D. Kossmann, T. Kraska, and S. Loesing
Now-a-days the cloud computing systems have many advantages for deploying applications in real time systems, example for such applications are data-intensive applications.
Under this system, the user follows pay-as-you-go service model, under which user pay to the services, which he has used. The system provide a promising service with reduced cost. Another promising feature service to add here is, the system provides unlimited throughput by adding servers, if workload of the system increases. The authors in this paper suggests alternative architectures, to make the cloud computing getting used in database applications, and after evaluating the existing cloud services which have accepted these architectures for their operation, the results will be noted in the reports. They have focused on the transaction processing work, such as read and update workloads, instead of on the other processing like analytics operation or OLAP workloads operation. These operations recently have attracted great deal of attention. The results what they have got here are surprising in many ways. More importantly many vendors have adapted different architectures for their cloud related services. And therefore as a result of this cost and performance of services vary in great extent depending on the workload of the system.

Scope of the Project

The scope of this project is to upload and download a file from cloud. While providing cloud consistency, the following objectives are to be met:

1] Understanding the novel consistency as a service (CaaS) model provided by the cloud service provider.
2] The cloud computing solution should provide basic consistency as service.
3] Maintain synchronized clock at audit clouds that responsible for checking weather cloud provide promised consistency or not.
4] Service Availability.

Existing System

In the existing systems by using the specified cloud storage services in the system, the customers or end users able to access information stored in a cloud anytime and anywhere making use of or from any device. And here the user does not need to worry about or no need to care about large amount of capital investment during the deployment phase of the underlying hardware infrastructures systems. The cloud service provider (CSP) stores multiple copies of data i.e. replicas on different servers distributed geographically. Where a user can read unwanted data or the data which is not updated for a period of time. The general system known as domain name system (DNS) is considered to be one of the most famous application systems that is going to implement eventual consistency operation. Here the updates done to a name will not be able to see or visible immediately in the system, but the system to the clients working with the system are have make sure they going to see them eventually.

DISADVANTAGES OF EXISTING SYSTEM:

- Even though the infrastructure systems present under the cloud system are considered to be more powerful and more reliable than computing systems, but they are facing issue of data integrity considering both inter and external threats.
- And next secondly, there have been many possibilities or motivations for CSP, which make CSP to behave in an unfaithful manner, with the cloud user with respect to their outsourced data onto the cloud system.
- And practically it is not feasible to download all the data information for the integrity checking or verification, and it is not suggested to be a right practical solution, because it results into huge expenses related to I/O operations and also transmission operation cost over the network. And it has failed in detecting the data corruption, when user is accessing the data, and even it is unable to give correct assurance to the user that there is not going to data loss or data damage during the operation.
- From the security point of view, the encryption technology is failed in giving complete security and even in protecting the data privacy against the third party auditing system. And as the decryption keys are exposed openly, data leakage problem still is the open issue in such systems.

PROPOSED SYSTEM

In this project work, we going to propose a standard model called as a consistency technique defined as **CaaS** model and here the proposed model follows a two-level auditing structure, which helps the users to check whether the cloud service provider (CSP) guaranteeing consistency service, and to express the severity of the violations, present if any.
Next with the CaaS model, the system users can able to assess the quality work of cloud services and take the decision to select a right CSP in the various candidates present. Example the cloud service which provides less expensive one operation but able to provide strong consistency for the applications of users.

Advantages:

- It is an emerging one, and cloud consistency is playing an important role in systems like decision support systems, which support the activity of everyday life.
- Going to get the efficient results based on the results provided by CaaS model.

Implementation

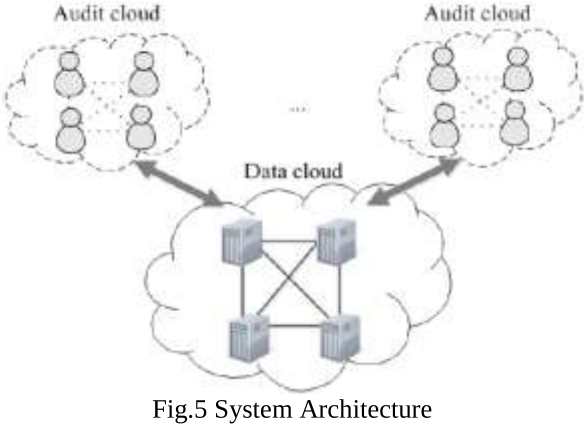

Fig.5 System Architecture

The proposed work contains the following modules:

1. System Module
2. User operation table
3. Local Consistency Auditing
4. Global Consistency Auditing

Modules Description

1. System Module:

In the first module, we develop the System Module with User Module, Admin Module, and Auditor Module.

In user module, the authorized user should undergo registration operation andregister their detailed information and collect the secret key information for login operation and user can able to upload the file regarding the operation like auditing. Next in user module, system user already uploaded files can be stored in system cloud database in systematic manner. Next Auditor can view the file or locate the file present in the database and it can be very secured.

In admin module admin will be able to view all the register user details; and also user uploads information details, and third party TPA activities regarding the auditing strategy.

In auditor module, the generally selected auditor can do the auditing operation based on the strategy operation called as heuristic auditing strategy. This operation is related to the basic operation of document verification. Then special unit Auditor can collect and check the auditing file, the he decide whether to reject or accept the file. On this he is going to make a report and enter all the details and about the decision like whether it’s good or bad. Also in this module an auditor can submit revision report. In this report information like accept or waiting. After the decision if status is present as a accept means then the user can view the file else if the status is waiting condition means then user cannot view the particular file.

2. User Operation Table:

In this module each user is going to maintain a UOT for the operation of recording local operations in systematic manner. Then each present record in the UOT unit is shown or explored by three components: first is an *operation parameter*, next the present *logical vector*, and finally more important one is *physical vector*. When the user is working on any operation, he is going to record his complete activity and also the current logical vector and final value as a physical vector, in his own UOT. In this module each user of the system is going to maintain a special logical vector and a basic physical vector to track the complete logical and physical time when an operation is going to take place correspondingly.

3. Local Consistency Auditing:

Local consistency auditing technique is an online algorithm. The operation in this module or unit, in which each user is going to record all his complete activities and store in his UOT. During the read operation, the authorized user is going to perform local consistency operation in an independent manner.

4. Global Consistency Auditing:

Global consistency auditing technique is going be considered as an offline algorithm. Next to consider is that an auditor periodically will be selected from the audit cloud system to perform the special operation like global consistency auditing technique. Hence in this case the auditor is going to collect all users’ UOTs for obtaining a special global trace of all activities. Then later executing global auditing technique, selected auditor is going to send results of auditing operation as well as its vectors values to all other authorized users . Now given the auditor’s vector values, then each user will come to know other users’ new clocks up to next global auditing.

ALGORITHMIC STRATEGY

Algorithmic strategy contain following algorithms:

A. Local Auditing Algorithm:

```
1.Initially user_operation_table with null while issue an operation op do
2. if op = w(a) then record w(a) in user_operation_table
3. if op = r(a) then  w(b) Belongs to user_operation_table is the last write
4. if w(a) -> w(b) then read your write consistency is violated r(c) belongs to user_operation_table is the last read
5. if w(a) -> w(c) then monotonic consistency is violated
6. record r(a) in user_operation_table
```

B. Global Auditing Algorithm

```
1. for every operation in the global trace is represent by a vertex
2. for operation op1 and op2 do
3. if op1->op2
Then time edge is added between op1 and op2
4. if op1=w(a),op2=r(a) op1 and op2 comes from different user then data edge is inserted between op1 and op2
5. if op1=w(a) and op=(b) and op1 and op2 comes from different users and w(a)->w(b)->r(b) then causal edge is
    inserted between op1 and op2
```

6. Verify whether the graph is directed acyclic graph by topological sorting method

Global auditing algorithm contain all strategy describe in figure 6.

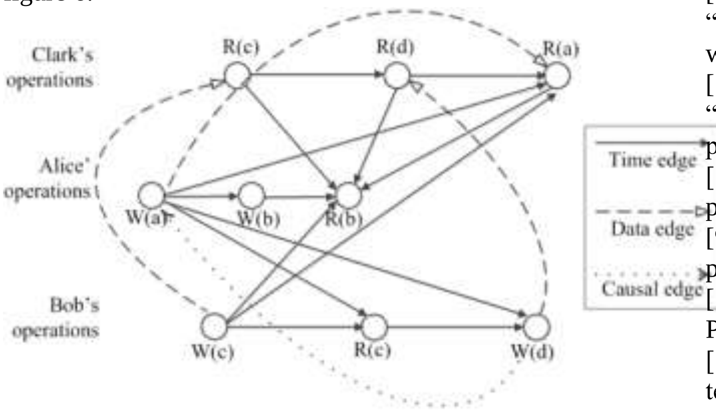

Figure 6. Global consistency

Conclusion

From this study we can conclude that the consistency service model is going to maintained by the system, and also we have come across a two levels of auditing structure technique. This technique helps the user to checks whether cloud service provider also called as CSP is going to provide a valid consistency operation or not with help of few violations if present in the system. And in this system the user also can understand which Cloud Service Provider genuine service provider from the other different Cloud service provider . Here we also want to consider that the Consistency is maintain or managed by Local Consistency Auditing and also by the Global Consistency Auditing Cloud.

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 cloudcomputing,” 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,” in Proc. 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,” in Proc. 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] ——, “Eventually consistent,” Commun. ACM, vol. 52, no. 1, 2009.

[13] M. Brantner, D. Florescu, D. Graf, D. Kossmann, and T. Kraska, “Building a database on S3,” in Proc. 2008 ACM SIGMOD.

[14] T. Kraska, M. Hentschel, G. Alonso, and D. Kossmann, “Consistency rationing in the cloud: pay only when it matters,” in Proc. 2009 VLDB.

[15] S. Esteves, J. Silva, and L. Veiga, “Quality-of-service for consistency of data geo-replication in cloud computing,” Euro-Par 2012 Parallel Processing, vol. 7484, 2012.

[16] 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.

[17] D. Bermbach and S. Tai, “Eventual consistency: how soon is eventual?” in Proc. 2011 MW4SOC.

[18] M. Rahman, W. Golab, A. AuYoung, K. Keeton, and J. Wylie, “Toward a principled framework for benchmarking consistency,” in Proc. 2012 Workshop on HotDep.

[19] D. Kossmann, T. Kraska, and S. Loesing, “An evaluation of alternative architectures for transaction processing in the cloud,” in Proc. 2010 ACM SIGMOD.

[20] L. Lamport, “On interprocess communication,” Distributed Computing, vol. 1, no. 2, 1986.

[21] A. Aiyer, L. Alvisi, and R. Bazzi, “On the availability of non-strict quorum systems,” Distributed Computing, vol. 3724, 2005.

[22] J. Misra, “Axioms for memory access in asynchronous hardware systems,” ACM Trans. Programming Languages and Systems, vol. 8, no. 1, 1986.

[23] P. Gibbons and E. Korach, “Testing shared memories,” SIAM J. Computing, vol. 26, no. 4, 1997.

[24] G. DeCandia, D. Hastorun, M. Jampani, G. Kakulapati, A. Lakshman, A. Pilchin, S. Sivasubramanian, P. Vosshall, and

W. Vogels, “Dynamo: Amazon’s highly available key-value store,” in Proc. 2007 ACM SOSP.

[25] T. Gormen, C. Leiserson, R. Rivest, and C. Stein, Introduction to Algorithms. MIT Press, 1990.