International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

CaaS Model for Cloud Consistency

Neethu Krishnan[1], Pyarimol Nair[2]

[1]M.tech student, Department of Computer Science and Engineering, Mount Zion College of Engineering, Kadammanitta, Pathanamthitta, Kerala, India

[2]M.tech student, Department of Computer Science and Engineering, Mount Zion College of Engineering, Kadammanitta, Pathanamthitta, Kerala, India

Abstract: ***Now a day, due to the overwhelming advantages cloud storage services become popular. Multiple replicas for each piece of data are stored on geographically distributed servers' inorder to provide always on access. The main drawback of keeping such replicas is that it requires high cost for providing strong consistency. In this paper proposing a CaaS model, which is comprised of a data cloud and multiple audit clouds. We are implementing a two level auditing structure. At last heuristic auditing strategies are to reveal violations.***

Keywords: cloud storage**,** consistency as a service, CaaS model, two-level auditing, heuristic auditing strategy.

1. Introduction

Due to elasticity, scalability, and high availability at low cost, cloud computing is commercially popular. In cloud computing cloud storage is considered as a typical service, where the data is delivered as a service. The users can access the data from anywhere and at any time using the cloud storage services.

Inorder to provide always on access, the cloud service provider maintains replicas for each data on geographically distributed servers. One of the main issues for storing such replicas is that it is very expensive while considering the strong consistency. Many CSP's only provides weak consistency for the high performance and availability. Eventual consistency is provided by the domain name system.

Have different consistency requirements for different applications. The consistency not only defines the correctness of the data but also the actual cost per transaction. In this paper presenting a consistency as a service model, which constitutes a large data cloud which is maintained by CSP and multiple small audit clouds which may be the users working on a project or a document or any other jobs. Between the large data cloud and the multiple audit cloud there exist a service level agreement which determines the level of consistency. Here adopting a two level auditing structure called local level auditing and global level auditing. Local level auditing is performed by the individual users and is focusing on monotonic read consistency and read your write consistency. Inorder to perform global level auditing an auditor is elected from the audit cloud and is focusing on casual consistency. Finally, proposing a heuristic auditing strategy inorder to reveal as many violations as possible.

2. Related Work

Cloud is a nebulous assemblage of computers connected via network and now a days it become commercially popular due to its high availability, scalability etc. at low cost. A. TANENBAUM AND M. VAN STEEN [2] proposes two types of consistency models called data centric consistency model and client centric consistency model. The data centric model focuses on the internal storage of a system. The main disadvantage of the data centric model is that for a customer its really doesn't need to know whether the internal storage contains any stale copies or not so the client centric model. It focuses on the specifications that the customer wants but it doesn't satisfy the monotonic read consistency.

W. VOGELS [3] says that strong consistency is not required in practice and it is very expensive to achieve strong consistency. Then, followed the work on achieving different levels of consistency in a cloud and found the consistency properties provided by commercial clouds and had many useful findings. Existing commercial clouds doesn't provide strong consistency (Google's MegaStore and Microsoft's SQL Data Services), and provides only weak consistency called eventual consistency (Amazon's simpleDB and Google's BigTable) also described several solutions to achieve different levels of consistency while deploying database applications on Amazon S3. But the consistency requirement depends on time and may vary according to time depending on actual availability of the data. So the Amazon.com technology developed a set of advanced business and infrastructure services that are implemented using scalable distributed systems. In that environment we can analyse a large number of particular data access patterns, based on their own consistency requirements. Thus to provide a collection of more diverse business processes the different patterns are provided.

E. BREWER [4] AND PUSHING THE CAP [5] states that the modern distributed systems adopted new types of data stores but that are not providing any strong consistency. That is the CAP theorem and their evolutions have a great influence on these systems which in turn scarifies the strict consistency but provides weak consistency with high availability.
E. ANDERSON, X. LI, M. SHAH, J. TUCEK, AND J. WYLIE [6] proposed some efficient algorithms. These algorithms analyzes the trace of interactive operations between the client machines and key value store i.e. the data

store reports weather the trace is safe , regular . Inorder to satisfy their goals they need to scarify the strong consistency and provides weak consistency with high availability. Many tools are used to verify the violations. After that concludes that the whether system is good or bad and provides promised level of consistency or not.

3. Implementation

In this paper it's mainly focused on four methods
1. CaaS model
2. UOT
3. Auditing(Two level auditing)
4. Heuristic auditing strategy(HAS)

3.1 CaaS model

The CaaS model consists of a large data cloud and multiple small audit clouds. The CSP manages the large data cloud. The small multiple audit cloud can be the users or customers working on a job such as a document or a project. The consistencies are verified on each audit cloud locally and globally then after that the data is transferred to the large data cloud by SLA. That is the Service Level Agreement which in turn consists of certain rules regarding the consistencies. Locally we check the monotonic read consistency and read your write consistency. And globally we are checking the casual consistency.

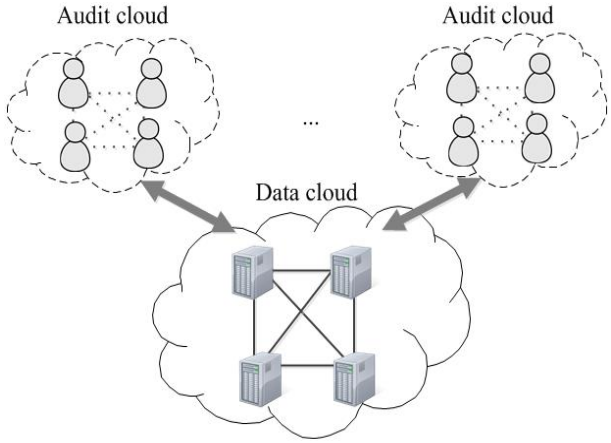

Figure 1: CaaS Model

The figure 1 shows the architecture of CaaS model which consists of different audit clouds and a large data cloud.

3.2 UOT (User operation Table)

One of the main methods is generating the UOT. For storing the local operations each user maintains their own UOT. The consistencies are verified using the User Operation Table. The UOT records all the operations and their corresponding logical vector and physical vector. The logical vector increments by one when an event happens that can be a read, write, send message, receive message etc. The physical vector is incremented as the time passes. And these two vectors are send along with the messages that has to be send. The physical vector and the logical vector are updated with its maximum value after receiving at the user side. Figure 2 shows the flow of logical vector and physical vector.

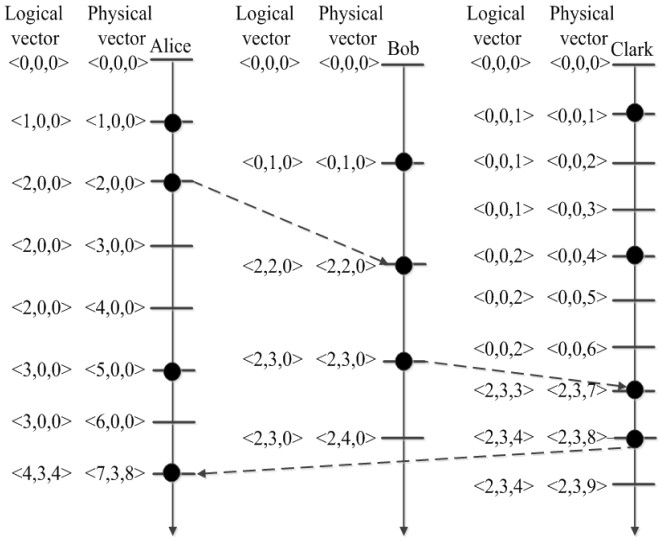

Figure 2: The update process of logical vector and physical vector. A black solid circle denotes an event (read/write/send message/receive message), and the arrows from top to bottom denote the increase of physical time.

Consider the figure 2, there shows the three audit cloud with three users Alice, Bob, and Clark. From that it's clear that initially the physical and logical vectors for each users is zero and as the time passes the physical vector is incremented, and logical vector is incremented by one when an event happens.

3.3 Auditing (Two level auditing)

Consistency is the main problem in cloud computing while replicating each piece of data for providing always on access. In this paper proposing a two level auditing structure called local level auditing and global level auditing. In local level auditing, each users in the audit cloud will perform the auditing individually with their own UOT. In the local level auditing we are focusing on monotonic read consistency and read your write consistency. When coming to the global level auditing, we need to select an auditor from the cloud. The selection of the auditor is carried out periodically. After selecting the auditor all the other users in the audit cloud need to transfer their UOT to the auditor and the auditor perform the global level auditing with that UOT. That is in short we can say that local level auditing is performed locally and global level auditing is performed globally.

Local level auditing is mainly focused on monotonic read consistency and read your write consistency. And the Global level auditing is focusing on casual consistency. Both the global level and local level auditing is based on the UOT.

3.4 Heuristic Auditing Strategy (HAS)

From the CaaS model it is found that only read operation can reveal the violations. So HAS mainly focusing how to reveal as many as violations possible. Inorder to reveal violation need to add appropriate reads called auditing reads.

As shown in figure 3 the physical time is divided into different time slices called L and *l* denotes an interval. Each time slice is associated with one of the two states called normal state or abnormal state. If the state is normal then we can regarded as the data maintains the consistency else not.

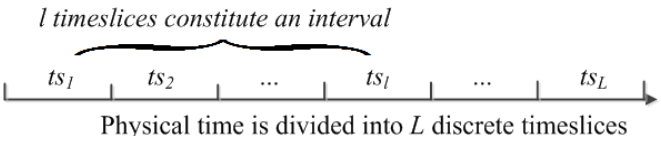

Figure 3: Physical time is divided into time slices.

4. Conclusion

In this paper, Consistency as a service model is presented, which provided promised level of consistency. The consistency level is verified by two level auditing structures. The users can check the quality of cloud services with the CaaS model. Most importantly the strong consistency can be achieved with low cost.

References

[1] Qin Liu, Guojun Wang, IEEE, Member, and Jie Wu, IEEE Fellow "Consistency as a service: Auditing cloud consistency", IEEE Vol 11 No.1, July 2014.
[2] A. TANENBAUM AND M. VAN STEEN, Distributed Systems: Principles and Paradigms. Prentice Hall PTR, 2002
[3] W. VOGELS, "Data access patterns in the Amazon.com technology platform," in Proc. 2007 VLDB.
[4] E. BREWER, "Towards robust distributed systems," in Proc. 2000 ACM PODC.
[5] Pushing the CAP: strategies for consistency and availability," Computer, vol. 45, no.2, 2012
[6] 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.
[7] W. GOLAB, X. LI, AND M. SHAH, "Analyzing consistency properties for fun and profit," in Proc. 2011 ACM PODC.
[8] 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.
[9] D. Bermbach and S. Tai, "Eventual consistency: how soon is eventual?" in Proc. 2011 MW4SOC
[10] M. Rahman, W. Golab, A. AuYoung, K. Keeton, and J. Wylie, "Toward a principled framework for benchmarking consistency," in Proc. 2012 Workshop on HotDep.

Author Profile

Neethu Krishnan received the B.Tech degree in Computer Science and Engineering from Mount Zion College of engineering in 2012. And currently she is pursuing her M.Tech degree in Computer Science and Engineering at Mount Zion College of Engineering.

Pyarimol Nair received the B.Tech. degree in Computer Science And Engineering from Mount Zion College of engineering in 2012. And currently she is pursuing her M.Tech degree in Computer Science and Engineering at Mount Zion College of Engineering.