International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org
Volume 4, Issue 1, January-February 2015 **ISSN 2278-6856**

A Survey on Improving Cloud Consistency Using Audit Cloud

Vasanti Kulkarni[1], Megha Bansode [2], Gaurav Mirje[3], Sanket Solanke[4], Prof. Spurti S. Shinde[5]

Modern Education Society's College of Engineering, University of Pune

Abstract

Cloud storage services have become commercially popular now a day. To provide unique and updated access of file on cloud, a cloud service provider (CSP) maintains multiple replicas for each piece of data on geographically distributed servers. The main problem of using the replication technique in clouds is that it is very costly to maintain consistency among data on clouds. In this paper, we first present a novel consistency as a service (CaaS) model, which consists of a large data cloud and multiple small audit clouds. In this model because of these small audit clouds it will be easy to maintain consistency in cloud data globally. In our model we apply two-level auditing architecture, which only requires a loosely synchronized clock in the audit cloud. Also we use algorithms to quantify the severity of violations with two metrics: the commonality of violations, and the staleness of the value of a read. Then we are going to implement these algorithms in java. Also Java is platform independent language well as we can maintain much more security aspect, so this will make our model more flexible and secure. Additionally we devise a heuristic auditing strategy (HAS) to reveal as many violations as possible.

Keywords: Cloud storage, consistency as a service (CaaS), two-level auditing, heuristic auditing strategy (HAS), Cloud Service Provider (CSP).

1. INTRODUCTION

Our model consist of large data cloud and 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. 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. 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.

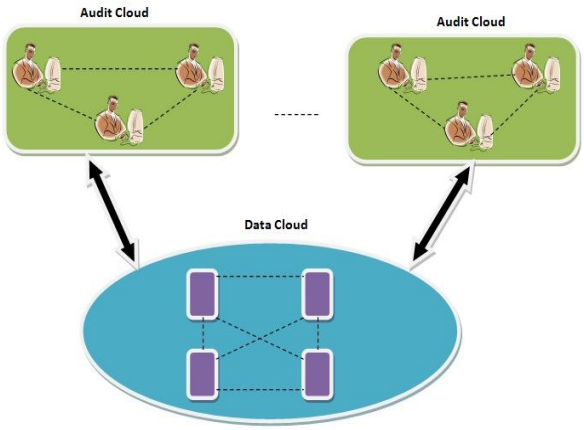

Figure 1. Consistency as a service model

2. RELATED WORK

As per our survey there are many previously worked done in the field of cloud data consistency. In paper [1] they reduce that confusion by clarifying terms, providing simple figures to quantify comparisons between of cloud and conventional Computing, and identifying the top technical and non-technical obstacles and opportunities of Cloud Computing. We believe the only plausible solution to very high availability is multiple Cloud Service Providers (CSP).We predict Cloud Computing will grow, so developers should take it into account. Regardless whether a cloud provider sells services at a low level of abstraction like EC2 or a higher level like AppEngine, we believe that computing, storage and networking must all focus on horizontal scalability of virtualized resources rather than on single node performance. In paper [2] they provide the better way to store any file on cloud storage. 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 previous systems. But storing a file on a cluster is producing huge problem for providing consistency. Measuring consistency is a very important task in this system because monitoring and controlling consistency is major goal of proposed system. By the time various benchmarking techniques are offered. In [3] they are providing whole new perspective to see the need of consistency as a service (CaaS).In cloud

computing storage services, every service request has an associated cost. In particular, it is possible to assign a very precise monetary cost to consistency protocols (i.e., the number of service calls needed to ensure the consistency level times the cost per call). Therefore, in cloud storage services, consistency not only influences the performance and availability of the systems but also the overall operational cost. In [4] to know the consistency models we have studied local and global consistency model of Dengyong Zhou, Olivier Bousquet, and Thomas Navin Lal. The key to semi-supervised learning problems is the consistency assumption, which essentially requires a classifying function to be sufficiently smooth with respect to the intrinsic structure revealed by a huge amount of labeled and unlabeled points. We proposed a simple algorithm to obtain such a solution, which demonstrated effective use of unlabeled data in experiments including toy data, digit recognition and text categorization.

3.Proposed System Design

3.1. Design and Implementation Constraints

Main Objective of this project is to upload and download a file from cloud. While downloading file from cloud, we are checking whether the file is updated version or not. Also finding out time require updating the file within multiple servers. 1] providing the upload and download tab with brows feature.2] Button to submit the file on cloud.3] Text Label to display warnings, exceptions and errors.4] Message boxes to display messages to user such as operation successful or not.

3.2. Assumptions and Dependencies

1. Assumptions:

- In general it has been assumed that the user has complete knowledge of the system that means user is not a naïve user.
- The software is as user friendly as possible but at the same time keeping in minds user requirements.
- All the PCs must have Windows OS with .NET framework on it and configured to form a cluster.

2. Dependencies:

- It depends that the one should follow the international standards for the generating the User ID & should fill the related information in the proper format.

3.3. User Classes and Characteristics:

This type of systems should be used by any type of users. Because of this, system shall include a clear documentation, which should be understandable by elementary computer users. There are various kinds of users for the product. Who are:-

1] Admin who will be acting as the controller.
2] Customer who will be using the above features by accessing the Software.

Both users make effective use of cloud service provided by service provider.

4.ALGORITHM USED

From paper [8] we find the efficient algorithm which we are going to implement in our model. We are using JAVA language to implement this algorithm. This algorithm is for Local Consistency Auditing. The steps in algorithm are as follows:

1] Initial User Operation Table (UOT) with Ø
2] While issue an operation op do
3] If op = W (a) then
4] Record W (a) in UOT
5] If op = r (a) then
6] W (b) ϵ UOT is the last write
7] If W (a) → W(b) then
8] Read-your-write consistency is violated
9] R(c) ϵ UOT is the last read
10] If W (a) → W(c) then
11] Monotonic-read consistency is violated
12] Record r (a) in UOT

Where,
W (a) - Write operation
R (a) - Read operation

5.DETAILED DESIGN

5.1. State-Transition Diagram

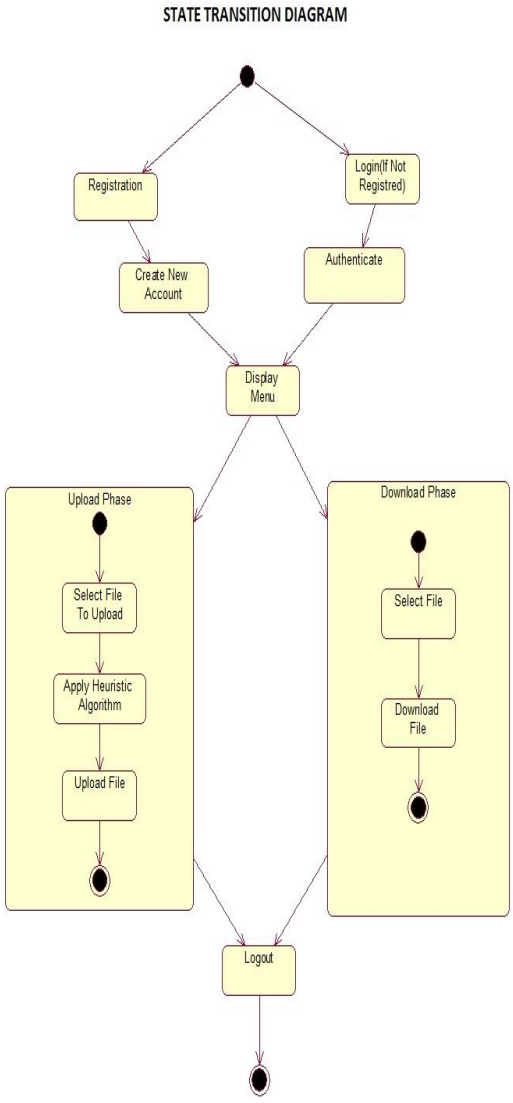

Figure 2. State Transition Diagram

5.2. Data Flow Diagram

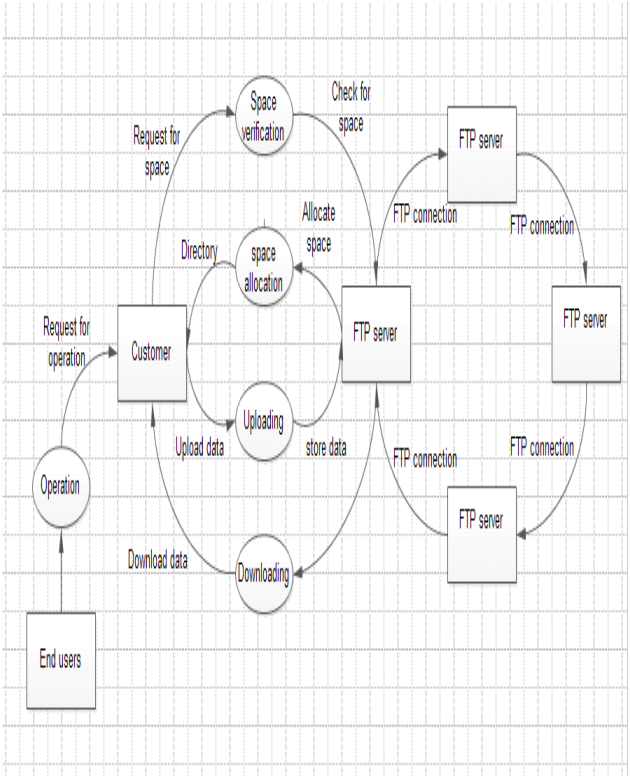

Figure 3. Data Flow Diagram

5.3. System Architecture Diagram

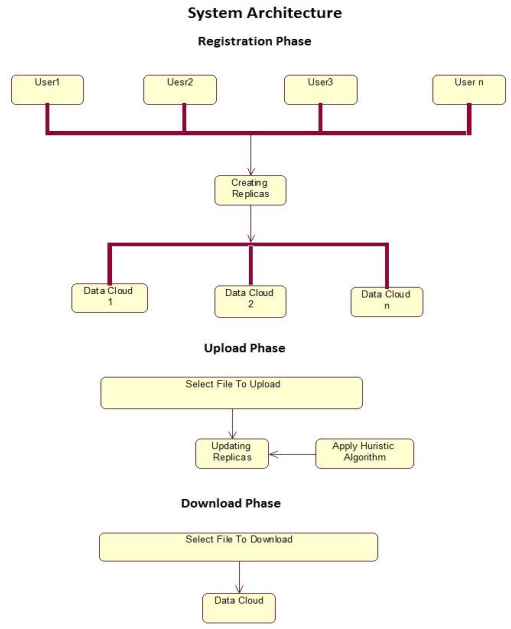

Figure 4. System Architecture

6.CONCLUSION

In this paper, we survey different technology and algorithms to implement CaaS base model and a two-level auditing structure to help users verify whether the cloud service provider (CSP) is providing valid security and consistency. With the CaaS model, the users can assess the quality of cloud services and choose a right CSP among various candidates. Also we are going to implement CaaS base model in Java which will be very important, portable and make better future technology in cloud storage.

References

[1] Static Verification of Data-Consistency Properties, by Nicholas A. Kidd University of wisconsin– Madison, 2009.

[2] Privacy-Preserving Public Auditing for Secure Cloud Storage Cong Wang, Student Member, IEEE, Sherman S.- M. Chow, Qian Wang, Student Member, IEEE, Kui Ren, Member, IEEE, and Wenjing Lou, Member, IEEE.

[3] Learning with Local and Global Consistency Dengyong Zhou, Olivier Bousquet, Thomas Navin Lal,Jason Weston, and Bernhard Sch¨olkopf Max Planck Institute for Biological Cybernetics, Tuebingen.

[4] Service Level Agreement (Sla) Based Scheduling Algorithms For Wireless Networks Mehdi Alasti, Farrokh R. Farrokhi, Masoud Olfat, And K. J. Ray Liu Electrical And Computer Eng. Department,University Of Maryland, College Park.

[5] Benchmarking Eventual Consistency: Lessons Learned From Long-Term Experimental Studies David Bermbach Karlsruhe Institute of Technology, Karlsruhe.

[6] P. Mell and T. Grance, "The NIST definition of cloud computing,"NIST Special Publication, 2011.

[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] Consistency as a Service: Auditing Cloud Consistency Qin Liu, Guojun Wang, IEEE, Member, and Jie Wu, IEEE Fellow IEEE Transactions On Network And Service Management, 2014.

AUTHOR

Gaurav Mirje pursuing BE in Computer Engineering from MES college of engineering, Pune.

Megha Bansode pursuing BE in Computer Engineering from MES college of engineering, Pune.

Sanket Solanke pursuing BE in Computer Engineering from MES college of engineering, Pune.

Vasanti Kulkarni pursuing BE in Computer Engineering from MES college of engineering, Pune.

Prof. S.S. Shinde received BE in Computer Engineering from PCCOE, Pune.