ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2015

Providing Consistency in Cloud Using Read after Write Technique to Endusers

R.Jeena[1], Dr.S.Saravanakumar[2], B.Poornima Bharathi[3], R.P.Priyancaa [3]

Assistant Professor, Department of Information, Panimalar Institute of technology, Chennai, India[1]

Professor, Department of Information, Panimalar Institute of technology, Chennai, India[2]

UG Student, Final Year, Department of Information, Panimalar Institute of technology, Chennai, India[3]

ABSTRACT: A cloud service provider maintains multiple replicas for each piece of data on geographically distributed servers. In Existing system ,there occurs conflict in shared files,if one member in the group opens the file and editing or do some operations,if another member in the group opens and modifies the same file which is being used by another one then there occurs a conflict. In existing the algorithm used is HAS,heuristic auditing strategy but in proposed we are following Read-after-write consistency allows you to build distributed systems with less latency. As touched on above, without read-after-write consistency you'll need to incorporate some kind of delay to ensure that the data you just wrote will be visible to the other parts of your system.

KEYWORDS: heuristic auditing strategy, Read-after-write consistency

I. INTRODUCTION

Cloud computing is a computing term or metaphor that evolved in the late 2000s, based on utility and consumption of computing resources. Cloud computing involves deploying groups of remote servers and software networks that allow centralized data storage and online access to computer services or resources. Clouds can be classified as public, private or hybrid. Cloud computing relies on sharing of resources to achieve coherence and economies of scale, similar to a utility (like the electricity grid) over a network.At the foundation of cloud computing is the broader concept of converged infrastructure and shared services.Cloud computing, or in simpler shorthand just "the cloud", also focuses on maximizing the effectiveness of the shared resources. Cloud resources are usually not only shared by multiple users but are also dynamically reallocated per demand. This can work for allocating resources to users. For example, a cloud computer facility that serves European users during European business hours with a specific application (e.g., email) may reallocate the same resources to serve North American users during North America's business hours with a different application (e.g., a web server). This approach should maximize the use of computing power thus reducing environmental damage as well since less power, air conditioning, rack space, etc. are required for a variety of functions. With cloud computing, multiple users can access a single server to retrieve and update their data without purchasing licenses for different applications.

II. PROBLEM STATEMENT

SYSTEM MODEL

In our paper we are going to develop a model which avoids theconflict in the group where files are shared betweenthe group members.This is achieved by passing the alert message in the group so that the second user must wait until the file is closed by the first user.Secondly the modified part in the file will be immediately visible after the updation achieved by read after write consistency. In our paper we use local consistency auditing.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2015

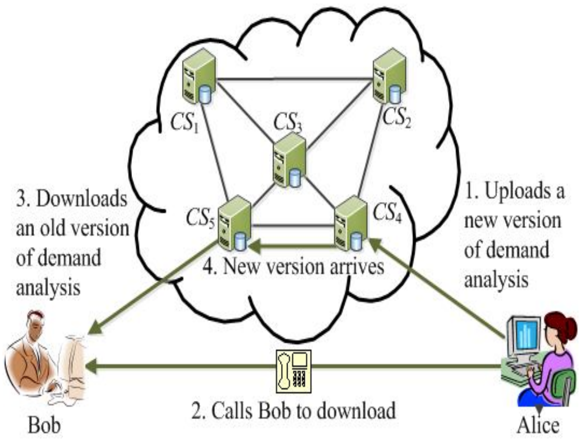
CS_1
CS_3
CS_2
CS_5
CS_4
3. Downloads an old version of demand analysis
1. Uploads a new version of demand analysis
4. New version arrives
2. Calls Bob to download
Bob
Alice

Fig. ARCHITECTURE DIAGRAM FOR EXISTING SYSTEM

DESIGN GOALS

We propose read after write auditing structure, which only requires a loosely synchronized clock for ordering operations in an audit cloud.In proposed system ,we are going to overcome the conflict . In a way if a file in the share is opened by particular member and if someone in the group tries to opens the same file ,the alert message will sent to the user and then others can see the updation made to file immediately. This provides consistency to the end users in cloud. This will avoid duplicate files in the cloud storage and conflict between the users.

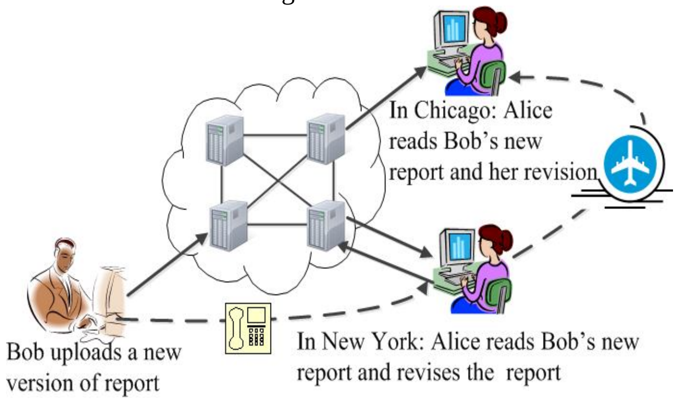
In Chicago: Alice reads Bob's new report and her revision
Bob uploads a new version of report
In New York: Alice reads Bob's new report and revises the report

FIG. ARCHITECTURE DIAGRAM FOR PROPOSED SYSTEM

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2015

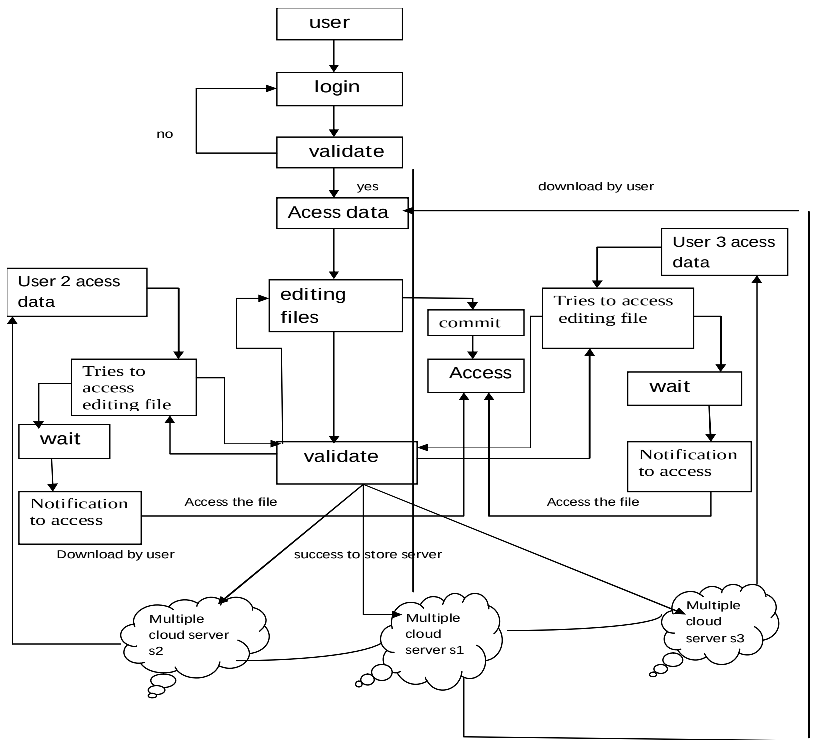

FIG. DETAILED DESIGN DIAGRAM

PRELIMINARIES

In this section we are going to see the detailed working and the modules used in the paper. In this paper we are going to implement three algorithms.

III. RSA ALGORITHM

The RSA scheme is a block cipher. Each plaintext block is an integer between 0 and $n - 1$ for some n, which leads to a block size $\leq \log2(n)$.The typical size for n is 1024 bits.

Key generation

Pick two large prime numbers p and q, $p <= q$

Calculate $n = p \times q$

Calculate $\Phi(n) = (p - 1)(q - 1)$

Pick e, so that $\gcd(e, \Phi (n)) = 1$, $1 < e < \Phi(n)$

Calculate d, so that $d\cdot e \bmod \Phi (n) = 1$, i.e., d is the multiplicative inverse of e in mod (n)

Get public key as Ku = {e, n}

Get private key as Kr = {d, n}

Encryption

For plaintext block $P < n$, its ciphertext C = Pe mod n.

Decryption

For ciphertext block C, its plaintext is P = Cd mod n.

RABIN-KRAP ALGORITHM

The Rabin–Karp algorithm or Karp–Rabin algorithm is a string searching algorithm created by Richard M. Karp and Michael O. Rabin (1987) that uses hashing to find any one of a set of pattern strings in a text.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 3, March 2015

```
functionNaiveSearch(string s[1..n], string pattern[1..m])
        fori from 1 to n-m+1
        for j from 1 to m
        if s[i+j-1] ≠ pattern[j]
        jump to next iteration of outer loop
        returni
        return not found
```

FORMULA
s[i+1..i+m] = s[i+1….i+m-1] - s[i] + s[i+m]

READ AFTER WRITE TECHNIQUE
Read after write is a technique that is used in our paper to provide consistency. In this technique , when a particular user is viewing or editing a file and if suppose other user wants to access the same file, alert message that a particular file cannot be accessed message will be displayed to the user. This means that write operation can be done only after the read operation.

LOCAL CONSISTENCY AUDITING
In this auditing, all the user in the cloud is a auditor and everyone will be having the UOT (User Operation Table). All the read and write values of all the user in the cloud will be saved in the UOT.

```
Initial UOT with ∅
whileissue an operation op do
ifop = W(a) then
recordW(a) in UOT
ifop = r(a) then
W(b) ∈UOT is the last write
ifW(a) → W(b) then
Read-your-write consistency is violated
R(c) ∈UOT is the last read
ifW(a) → W(c) then
Monotonic-read consistency is violated
recordr(a) in UOT
```

IV. IMPLEMENTATION

In our proposed system, database is allocated for the users who can store their files. If a user is a registered user he can go to the login page and access the files. If the user is not registerd he has to go to the admin page and register the details.

LOGIN
A user has to initially login to the cloud to get access to the files. For that the user has to give the username and the password that he has registered for the first time. Or if the user is a new user then he has to go the admin page and register and get the access.

UPLOAD FILE
When the user logins he can upload his file in the cloud database from his personal database. This provides replication of files. These replicas allows us to access the file from nearby geographical locations.

FILE SHARING
When a user uploads a file he can share those files with the other user in the cloud. This is enabled by file sharing. A particular user can share those files with user to whom he wants to. And the other users in the cloud cannot access the file.

EDITING A FILE

When a user wants to edit a file, editor is provided in the cloud and he can edit a file there. A file that is being uploaded may be uploaded anytime and it might needupload. This can be done in the editor that is provided in the cloud.

PROVIDING CONSISTENCY

When a user tries to edit and if a particular user in that cloud who is a shared user tries to access those files then conflict occurs. Proper data will not be saved in the cloud that is the file may be incorrect. To provide consistency, this technique is used. An alert message will be sent to the other user who tries to access that file.

V. CONCLUSION

In this paper, we presented a consistency as a service (CaaS) model and read after write technique 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, and to quantify the severity of and to quantify the severity of the violations, an do make data visible immediately after modification. This provides the consistency that is needed by the users in the cloud.

ACKNOWNLEDGMENT

Our sincere thanks to our guide Mrs. R. Jeena who gave us proper support to do this project. Our special thanks to our Information Technology Head Of the Department Dr.A.Joshi who provide the infrastructure in our college. We are also thankful to Mrs. DhanalakshmiRajasekaran coordinator of our project. We also thankfull to all other staff members for their cooperation.

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 cloud of computing,” Commun. ACM, vol. 53, no. 4, 2010.
[2] “Pushing the CAP: strategies for consistency and availability,”Computer, vol. 45, no. 2, 2012.
[3] C. Fidge, “Timestamps in message-passing systems that preserve the partial ordering,” in Proc. 1988 ACSC.
[4] H. Wada, A. Fekete, L. Zhao, K. Lee, and A. Liu, “Data consistency
[5] properties and the trade-offs in commercial cloud storages: the consumers perspective,” in Proc. 2011 CIDR.
[6] J. Misra, “Axioms for memory access in asynchronous hardware systems,”ACM Trans. Programming Languages and Systems, vol. 8, no.1, 1986.