INTERNATIONAL JOURNAL OF ADVANCE RESEARCH, IDEAS AND INNOVATIONS IN TECHNOLOGY

ISSN: 2454-132X
Impact factor: 4.295
(Volume 4, Issue 2)
Available online at: www.ijariit.com

Deadlock recovery and mutual exclusion with Atomic operations of cloud storage

N. Munesswari
munish0697@gmail.com
Tejaa Shakthi Institute of Technology for Women, Coimbatore, Tamil Nadu

S. Ishwarya
ishukarthik0620@gmail.com
Tejaa Shakthi Institute of Technology for Women, Coimbatore, Tamil Nadu

M. Karthikpriya
riyarinu.m@gmail.com
Tejaa Shakthi Institute of Technology for Women, Coimbatore, Tamil Nadu

A. M. Senthilkumar
senthil1185@gmail.com
Tejaa Shakthi Institute of Technology for Women, Coimbatore, Tamil Nadu

M. Saravanakumar
mskumarinfotech@gmail.com
Tejaa Shakthi Institute of Technology for Women, Coimbatore, Tamil Nadu

ABSTRACT

While using consumer electronic devices consumers continue to produce and spend a large amount of data. To store consumer's data, a huge volume of storage is needed. A cloud -of- a cloud which is a virtual cloud storage service using heterogeneous public cloud storage services can be a solution. To support concurrent accesses for multiple consumers and devices, a mutual exclusion in a cloud-of-cloud is necessary. From the base paper, they proposed a mutual exclusion method for concurrent access in a cloud-of-clouds. The proposed method includes an atomic operation for cloud-of-cloud a deadlock recovery and a race condition control. To evaluate the proposed method, we modified two modules of saving me which is a cloud-of-clouds storage service system with experiments on the implemented system, we found out to the optimal number of files in a file table and the deadline time for deadlock avoidance. The proposed method does need not only any server system but also any communication between clients. We conform that the proposed method provides mutual exclusion features to deadlock avoidance.

Keywords: *Cloud Computing, Data Storage, Files, Cloud-of-Clouds.*

1. INTRODUCTION

Cloud computing is an umbrella term used to refer to Internet-based development and services. The cloud is a metaphor for the Internet. A number of characteristics define cloud data, applications services & infrastructure(Remotely hosted) Services or data are hosted on someone else's infrastructure.(Ubiquitous)Services or data are available from anywhere.(Commoditized)The result is a utility computing model similar to traditional that of traditional utilities, Like gas and electricity. You pay for what you would like. Several large Web companies are now exploiting the fact that they have data storage capacity which can be hired out to others. This approach, known as 'cloud storage' allows data stored remotely to be temporarily cached on desktop computers, mobile phones or other Internet-linked devices. Amazon's Elastic Compute Cloud (EC^2) and Simple Storage Solution (S3) are well-known examples. Cloud Services can also be used to hold structured data. There has been some discussion of this being a potentially useful notion possibly aligned with the Semantic Web, though concerns, such as this resulting in data becomes undifferentiated, have been raised.

Many of the activities loosely grouped together under cloud computing have already been happening and centralized computing activity is not a new technique. Grid Computing was the last research-led centralized approach. However, there are concerns that the mainstream adoption of cloud computing could cause many problems for users. Whether these worries are grounded or not has yet to be seen.

To solve this problem in an existing system we propose a mutual exclusion method in the client-side aggregation of cloud storage. The proposed method provides not only mutual exclusion based on a Save Me for cloud-of-clouds from analysis APIs of public clouds. Since public cloud storage providers do not have computing resources, the proposed method works on only client-side with no communication between clients. To evaluate the proposed method, we modified two modules of Save Me which is a cloud-of-clouds storage service system. With experiments on the implemented system, we found out the optimal number of files in a file table and the deadline time for deadlock recovery. The proposed method does need not only any server system but also any Communication between clients. We confirm that the proposed method provides mutual exclusion features with deadlock avoidance.

2. EXISTING SYSTEM

In the existing system has discovered the "Save Me" for client-side aggregation of cloud storage. Multiple consumers to share their data in the mutual exclusion method is necessary there are many cloud service system has been researched. This work based on the individual public cloud without integrated storage between clients. It expands data with replication coding across multiple public clouds. The communication is necessary between clients. The online cloud storage system allows the private data for storing by the combination of private and public cloud which is available to each user. Clients have unstable connectivity with clouds and client which is disconnected between the lock and unlock and the other clients cannot write this situation is called as deadlock. If a client while detecting deadline which is missed lock, the client tries the wait file and check all the update time for all the waiting files and it performs the deadlock recovery method.

3. DISADVANTAGES IN EXISTING SYSTEM

- It has a lack of security system and have a to store consumers data large amount of storage is needed.
- Public cloud are not always available because of unavailability of the internet.
- No processing resource is available.
- Race condition control.

4. PROPOSED SYSTEM

To evaluate the proposed method, we modified two modules of Save Me which is a cloud-of-clouds storage service system. In this can read and write the related features between clients. From the implemented system, we found out the optimal number of files in a file table and the deadline time for deadlock recovery. The proposed method does need not only any server system but also any communication between clients. That the proposed method provides mutual exclusion features with deadlock avoidance and race condition control using comparison the optimal scenario sequence and the real working sequence of the implemented system. Concurrent access file systems from multiple clients file systems, a mutual exclusion, and a race condition control are essential features. In a cloud-of-clouds storage service.

5. ADVANTAGES OF PROPOSED SYSTEM

- To provide mutual exclusion.
- To implement deadlock avoidance.
- To apply ACID properties (atomicity, consistency, isolation, durability).

6. IMPLEMENTATION

Implementation is the stage in the project where the theoretical design in the turned into a working system and is giving confidence on the new system for the user that it will work effectively. It involves careful planning, investigation of the current system and its constraints on implementation, the design of methods to achieve the changeover, an evaluation, of change over methods. Apart from planning major task of preparing the implementations are education and training of users. The more complex the system begin the effort, required just for implementation. An implementation coordination committee based on policies of the individual organization has been appointed. The implementation process begins with preparing a plan for the implementation of the system. After the system is implemented successfully, training of the user is one of the most important subtasks of the developer. For this purpose user manuals are prepared and handled over to the user to operate the developed system. Thus the users are trained to the operate the developed system. Both the hardware and software securities are made to run the developed systems successfully in future.

The implementation stage involves following Tasks.

- Careful planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.
- Training of the staff in the changeover phase.

The maintenance phase of the software cycle is the time in which a software product useful work. After a system is successfully implemented, it should be maintained in a proper manner. System maintenance is an important aspect of the software development life cycle. The need for the system maintenance is for it to make adaptable to change in the system environment. There may be social, Technical and other environmental changes, which affect a system, which is being implemented. Software product enhancements may involve providing new functional capabilities, improving user displays and mode of interaction, upgrading the performance characteristics of the system. The maintenance activity occurs because it is unreasonable to assume that software testing

will uncover all errors in a large software system. During the use of any large program, errors, will occur and be reported to the Developer. The process that includes the diagnosis and correction of one or more errors is called Corrective maintenance.

7. CONCLUSION

From the investigation of the problem of deadlock avoidance in cloud data storage, which is essentially a distributed storage system. To ensure the correctness of users' data in cloud data storage, we proposed an effective and flexible distributed scheme with explicit dynamic data support, including block update, delete. Whenever data corruption has been detected during the storage correctness verification across the distributed servers, we can almost guarantee the simultaneous identification of the misbehaving server(s).The most promising one we believe is a model in which public verifiability is enforced. Besides along with our research on dynamic cloud data storage, we also plan to investigate the problem of fine-grained data error localization.

8. REFERENCES

[1] C. Wang, Q. Wang, K. Ren, and W. Lou, "Ensuring data storage security in cloud computing," in Proc. of IWQoS‟09, July 2009
[2] Amazon.com, "Amazon web services (aws)," Online at http://aws.amazon.com/, 2009.
[3] C. Wang, K. Ren, W. Lou, and J. Li, "Towards publicly auditable secure cloud data storage services," IEEE Network Magazine,vol. 24, no. 4, pp. 19–24, 2010.
[4] K. D. Bowers, A.Juels, and A. Oprea, "Proofs of irretrievability: Theory and implementation," in Proc. of ACM workshop on Cloud Computing security (CCSW'09), 2009, pp. 43–54.
[5] Q. Wang, K. Ren, W. Lou, and Y. Zhang, "Dependable and Secure Sensor Data Storage with Dynamic Integrity Assurance," Proc. of IEEE INFOCOM, 2009.
[6] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable and Efficient Provable Data Possession," Proc. of SecureComm ‟08, pp. 1–10, 2008.
[7] Q. Wang, K. Ren, W. Lou, and Y. Zhang, "Dependable and Secure Sensor Data Storage with Dynamic Integrity Assurance," Proc. of IEEE INFOCOM, 2009.
[8] K. D. Bowers, A. Juels, and A. Oprea, "HAIL: A High-Availability and Integrity Layer for Cloud Storage," Cryptology ePrint Archive, Report 2008/489, 2008, http://eprint.iacr.org/.
[9] L. Carter and M. Wegman, "Universal Hash Functions," Journal of Computer and System Sciences, vol. 18, no. 2, pp. 143–154, 1979.
[10] Q. Wang, K. Ren, W. Lou, and Y. Zhang, "Dependable and secure sensor data storage with dynamic integrity assurance," in Proc. Of IEEE INFOCOM‟09, Rio de Janeiro, Brazil, April 2009

