Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 5.258

IJCSMC, Vol. 5, Issue. 6, June 2016, pg.87 – 94

A New Framework and Algorithms for Secure Cloud Transactions

Naveen Bugga[1], A. Siva Kumar[2]
Computer Science & Engineering, JNTU Hyderabad
naveenbugga@gmail.com
TUDI RAM REDDY INSTITUTE OF TECHNOLOGY & SCIENCES
aluri.kumar@gmail.com

Abstract: Transactions over communication networks play a vital role. It does mean that these transactions are distributed transactions pertaining to relational databases. Cloud computing is in distributed environment where the database transactions are to be carried out with high level of integrity. In this paper we investigate the problem of security in distributed transactions and provide a framework for data integrity and security. The concept of trusted transactions is given importance. The security policies and other constraints are to be considered. The proposed framework takes care of integrity of transactions besides security. Towards this end, two phase commit protocol is implemented which takes care of ACID properties to ensure integrity of data. We built a prototype application that demonstrates the proof of concept. The empirical results revealed that our two phase commit protocol and other mechanisms are able to support data integrity and security in distributed environment in cloud computing.

Keywords: Cloud, database transactions, two phase commit, security

I. INTRODUCTION

Cloud computing is a new phenomenon which is in distributed environment. It provides a shared pool of resources that can be shared. Before cloud came into existence, client/server computing was prevailing. The applications of different kinds depend on the data stored in database server. Database servers are responsible to store data permanently. There are different kinds of databases in the world. However, a popular database model is Relational Model. On top of this model all relational database management systems are. For instance Oracle is an RDBMS which is meant for storing data permanently.

RDBMS provides security to the data in the form of access control mechanism. It has abilities to support users, roles and access privileges. Thus only authorized users can access the database. When applications gain access to database, they actually use Structure Query Language (SQL) as an underlying language to interact with relational databases. This language is used along with frond end so as to interact with relational databases.

Later on distributed databases and distributed computing came into existence. Cloud computing is one of the examples for distributed computing. The computing resources are commoditised effectively using cloud computing technology. The cloud computing is based on the concept of virtualization. Virtualization is responsible for optimal utilization of resources and cloud computing is complemented with this technology. As explored in [1] and [2] the cloud computing took time to come into existence. However, now it is a reality and we are able to use cloud computing platforms from different vendors. Some of the popular cloud platforms are Amazon's EC2, AMAZON web server, Google Apps and Microsoft Azure [1].

In cloud computing resources are rendered on demand. It provides support for IT with low cost [2], [3]. There are many advantages to cloud users as it provides many services at low cost with unlimited storage and computing resources [2], [4].

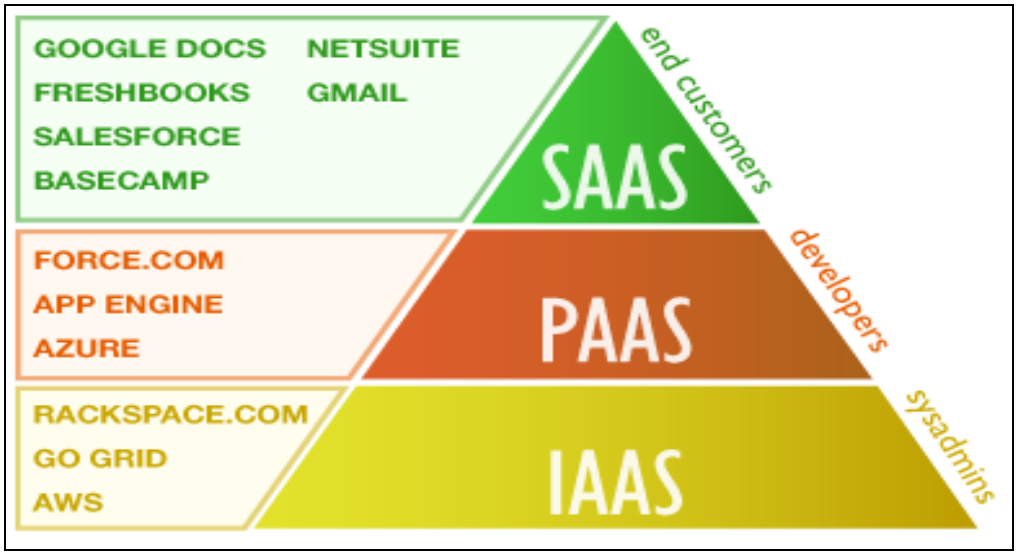

Figure 1 – Cloud computing services

As shown in Figure 1, cloud computing offers three services such as Software as a Service (SaaS), Infrastructure as a Service (IaaS) and Platform as a Service (PaaS). These layers offer intended services. The cloud computing is made affordable with virtualization technology. Virtualization is the "creation of a virtual (rather than actual) version of something, such as a server, a desktop, a storage device, an operating system or network resources". Virtualization can help in providing IT resources so that we can share these IT resources in order get benefits in the business [5], [6]. This paper throws light into proposing a better approach for transaction processing in distributed environment for cloud computing security. Consistent transaction management is explored in the paper. Our contributions in this paper are described here. We proposed a safe transaction mechanism for secure cloud computing operations. We implemented it using a prototype application that could demonstrate the proof of concept. The remainder of the paper is structured as follows. Section II provides review of relevant literature. Section III throws light into the proposed approach for safe and secure distributed transactions in cloud computing. Section IV presents experimental results while section V concludes the paper besides providing recommendations for future enhancements.

II. PROPOSED SOLUTION

We proposed architecture for secure distributed transactions in cloud computing environment. The architecture is shown in Figure 2. It throws light into the basic flow of transactions in the distributed environment. We proposed algorithm that focuses on the secure transactions with ACID properties ensured. Two phase validation commit protocol proposed is responsible for many things. It ensures authentication, authorization, security policies, transaction management, inconsistencies and importantly transactions. Transaction is a set of activities that are carried out in a server. They are to be carried out as a single unit. We enhanced two phase commit protocol that provide secure transactions in distributed environment.

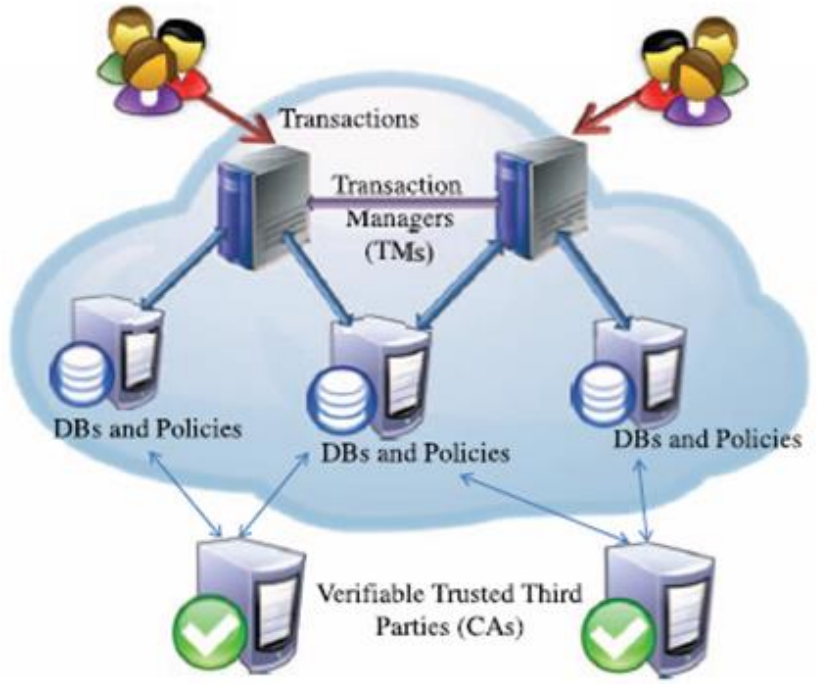

Figure 2 – Our solution is on top of this architectural overview

The solution proposed in the framework takes care of secure and trusted transactions besides following ACID properties of transaction in distributed environment. It ensures that the transactions are executed in different cloud servers securely and the data integrity is not lost. If any operation in the transaction in one of the servers is not carried out successfully, the underlying algorithm takes necessary steps to undo the whole transaction to ensure data integrity. The transactions are carried out with policy checking, credentials and data consistency. As explored in [8], [9] and [10] two phase commit is widely used for secure database transactions in the distributed environment. In this paper we improved it and proposed to ensure highly secure transactions.

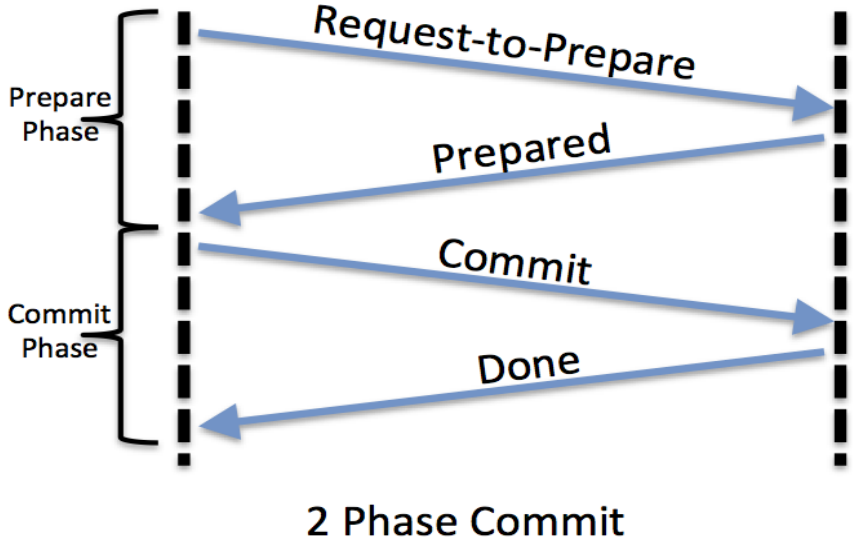

Figure 3 – Illustrates flow of 2 Phase Commit

The proposed 2 phase commit has different phases. The process illustrated in Figure 3 is explained here. The algorithm is very much similar to the 2PV but it contains additional integrity constraint validation. The decision of COMMIT or ROLLABCK is based on the constraints. The TM takes care of normal 2PV protocol behavior and then makes use of policies to ensure consistency. The TM is responsible to enforce global consistency by consulting all policies that are up to date.

Two-Phase Validation Commit - 2PVC (TM)

Step1: Send "Prepare-to-Commit" to all participants

Step2: Wait for all replies (Yes/No, True/False, and a set of policy versions for each unique policy).

Step3: If any participant replied No for integrity check

Step4: ABORT

Step5: Identify the largest version for all unique policies

Step6: If all participants utilize the largest version for each unique policy

Step7: If any responded False

Step8: ABORT

Step9: Otherwise

Step10: COMMIT

Step11: Otherwise, for participants with old policies

Step12: Send "Update" with the largest version number of each policy

Step13: Wait for all replies

Step14: Go to **Step5**

Figure 4 – Proposed algorithm

Our algorithm is in the similar lines of the algorithm presented in [12] and [13]. It is presented in Figure 4. Their algorithm and our algorithms are almost same with small differences. The validations are more in the proposed algorithm. When transactions are taken place in a single machine, no such protocol is required. When transactions take place in multiple servers then this protocol plays a vital role in ensuring secure and complete transactions in distributed environment. The proposed protocol takes confirmations from the participating servers. The time is also considered and both servers where transactions are taken place are to provide similar kind of response. If not the whole transaction is rolled back. Incremental punctual proofs provide additional means of ensuring consistency in distributed transactions.

The TM checks version number as well when it is received from participating servers. This is important for view consistency in distributed environment where multiple users are presented the data. Policies are enforced when transactions are committed or rolled back. The case of global consistency is ensured by the TM with policy revisions and validates and latest versions available in the master policy servers. Data integrity constraints are verified and continuous proof of inconsistencies are taken into consideration.

III. RELATED WORKS

This section provides review of literature on data integrity and security transactions in cloud computing. It throws light into different categories of cloud services such as IaaS, PaaS and SaaS.

3.1 Security Challenges in IaaS

There are many things involved in the infrastructure. It includes both hardware and software. In this service layer of cloud there are security challenges. However, it has much importance as it can store data and allows data retrieval. Since cloud servers are untrusted, the security issues arise when data is outsourced. There are issues when data is illegally modified or tampered. This needs to be prevented in order to ensure data integrity. The intrusions to databases are to be avoided and thus the distributed transactions that take place in the confines of IaaS. The security issues in the service layer include data intrusion, service availability, and data integrity [1]. Trust is one of the concerns in IaaS. The trust levels provided by the cloud service providers are different. There are many issues with respect to stakeholder rights and security issues in safeguarding the interests of cloud users [3]. Moreover there are many kinds of attacks that can occur on cloud services. In this context user access policies are very important as they play vital role in it [4]. As cloud is based on virtualization, the virtualization manager is in IaaS layer and it causes many security issues [6].

3.2 Security Challenges in SaaS

The SaaS layer is also subjected to issues. It has problems with malware, SQL injection attack and other security attacks. Many third party cloud services are providing enterprise level security specifications for protecting SaaS from malicious attacks such as SQL injection. There are other architecture level security issues. Many security problems are related to plain text communications and the lack of encrypted data flow [5]. Data security, identity security and trust related security issues arise in SaaS. The cloud service providers might try to gain access to sensitive information. They can also gain access to software and misuse the software uploaded [5].

3.3 Security Challenges in PaaS

This service provides the required cloud platform with API for application development. It throws many challenges related to security. They are service life cycle optimization, market and legislative issues, multi-cloud architectures, adaptive self-preservation, and dependable sociability [2]. The usage of web services and the applications that are built on the cloud platform API are vulnerable to many security threats. Earlier Cloud Security Alliance (CSA) reported around 10 important security issues in cloud computing.

Many solutions came into existence to overcome all these security issues. They are related to data integrity which focused on the consistency of data in the data dynamics. These solutions were explored in [7], [8], [9], [10], [11], [12] and [13] and [14]. In this paper, based on the solution in [14] we designed and implemented a security mechanism for data integrity in cloud computing. We also built an algorithm for safe and secure cloud transactions. We built a prototype application that show how the proposed framework ensures secure transactions in the distributed environment.

IV. EXPERIMENTAL RESULTS

Our prototype application is used to perform experiments with database transactions in distributed environment. Experiments are made in terms of safe and secure transactions. The observations in the transactions include the consistency of data in a distributed environment.

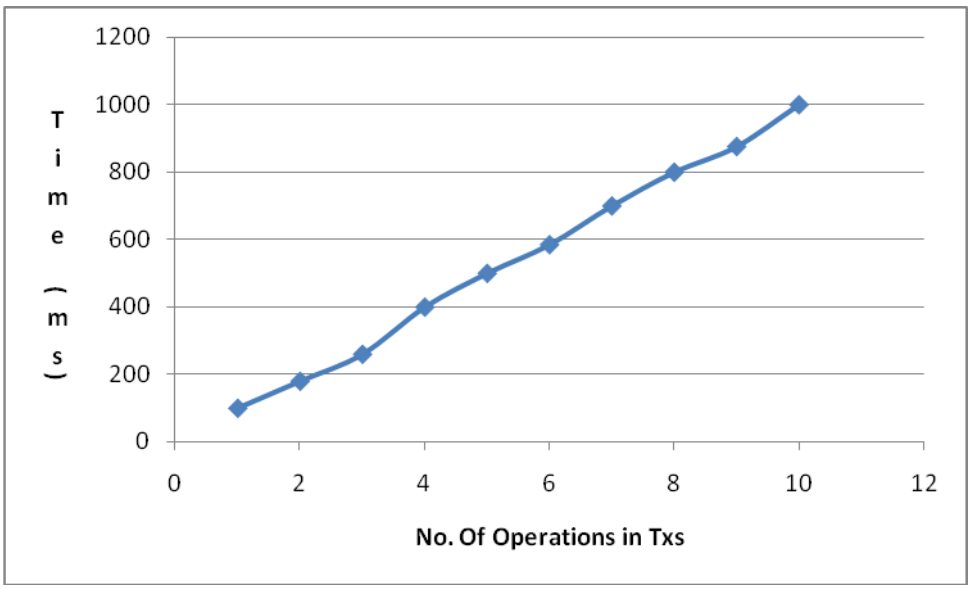

Figure 5 – Number of operations vs. time

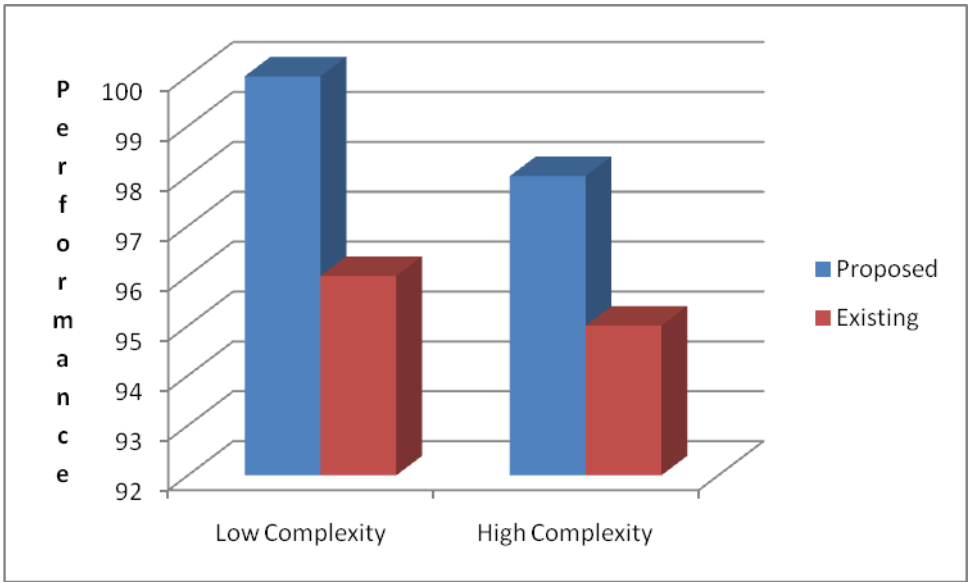

Figure 6 – Performance comparison in low and high complexity in distributed environments

As shown in Figure 5 and 6, it is evident that the proposed solution shows higher performance in terms of performance and the time taken with number of underlying operations in distributed transactions.

V. CONCLUSIONS AND FUTURE WORK

There are many security and data integrity concerns when data is outsourced to cloud. There are mechanisms and algorithms to ensure secure and data dynamics. The mechanisms enforce precision of authorization, accuracy, consistency and trustworthiness in transactions. Especially in distributed environment, it is essential to have consistency of transactions. In this paper we proposed a framework with underlying algorithm to ensure that ACID properties with additional validations are carried out in distributed environment like cloud computing. Safe and secure transactions are carried out with an improved two phase commit protocol. This protocol has two basic phases. In the first phase all participant servers provide the response. Based on the response of multiple servers, the second phase will take care of decisions like commit or rollback. In addition to this TM is responsible to ensure valid view and consistency in transactions. We built an application that demonstrates the concept and the empirical results are show that proposed algorithm is very useful. In future this research can be extended further to include more security policies and verify the distributed transactions.

REFERENCES

[1] John MacCharty. (1961). History of Cloud Computing.*http://www.javatpoint.com/history-of-cloud-computing*, p.45-56.

[2] techinmind. (2014). What Is Cloud Computing? What are It's Advantages and Disadvantages?. *Cloud computing security issues*, p.12-19.

[3] John MacCharty. (1961). What is Cloud Computing. *Cloud computing security issues*, p.20-30.

[4] John MacCharty. (1961). Advantages of Cloud Computing. *Cloud computing security issues*, p.90-101.

[5] MacCharty. (1961). Virtualization in Cloud Computing. *Cloud computing security issues*, p.45-56.

[6] ohn MacCharty. (2014). Virtualization and Cloud Computing. *ntel IT Center Planning Guide | Virtualization and Cloud Computing*, p.20-30.

[7] Binali Yıldırım. (2016). Cloud computing security.*https://en.wikipedia.org/wiki/Main_Page*, p.12-19.

[8]. Philip A. Bernstein, Vassos Hadzilacos, Nathan Goodman (1987): Concurrency Control and Recovery in Database Systems, Chapter 7, Addison Wesley Publishing Company, ISBN 0-201-10715-5.

[9].Gerhard Weikum, Gottfried Vossen (2001): Transactional Information Systems, Chapter 19, Elsevier, ISBN 1-55860-508-8.

[10].Philip A. Bernstein, Eric Newcomer (2009): Principles of Transaction Processing, 2nd Edition, Chapter 8, Morgan Kaufmann (Elsevier), ISBN 978-1-55860-623-4

[11].http://en.wikipedia.org/wiki/Two-phase_commit_protocol.

[12]. P.K. Chrysanthis, G. Samaras, and Y.J. Al-Houmaily, "Recovery and Performance of Atomic Commit Processing in Distributed Database Systems," Recovery Mechanisms in Database Systems, Prentice Hall PTR, 1998.

[13]. M.K. Iskander, D.W. Wilkinson, A.J. Lee, and P.K. Chrysanthis, "Enforcing Policy and Data Consistency of Cloud Transactions," Proc. IEEE Second Int'l Workshop Security and Privacy in Cloud Computing (ICDCS-SPCCICDCS-SPCC), 2011.

[14]. H. Guo, P.-A. Larson, R. Ramakrishnan, and J. Goldstein, "Relaxed Currency and Consistency: How to Say "Good Enough" in SQL," Proc. ACM Int'l Conf. Management of Data (SIGMOD '04), 2004.

