ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Implementation of stringent levels of policy consistency constraints for trusted transactions in distributed transactional database systems in cloud environment.

G.Vikas
M.Tech Student,
Department of CSE,
Sarada Institute of Technology and Science
Khammam, Telangana, India.

B.Laxmaiah
Assistant Professor
Department of CSE
Malla Reddy College of Engineering for Women
(MRCEW), Quthbullapur, Hyderabad – 500 055.

Abstract: ***Cloud computing involves application systems which are executed within the cloud and operated through internet enabled devices. Purely cloud computing does not rely on the use of cloud storage as it will be removed upon users download action. Clouds can be classified as public, private and hybrid. For transactions to be secure, we need to address various Constraints from an end-user and Cloud service provider's point of view. The end-user is primarily concerned with the provider's security policy, how and where their data is stored and who has access to that data. On the other hand, concerns for the Cloud service provider can range from the physical security of the infrastructure and the access control mechanism of cloud assets, to the execution and maintenance of security policy. In this paper, we analyze the methodologies used to authorize users who access distributed database systems and the risks faced by these methodologies. To increase the trustworthiness of the transactions and also to ensure its accuracy, a combination of Two-Phase Validation Commit Protocol and Blow-fish algorithm is proposed. We analyze this approach through simulation method and the results are shared.***

Keywords: ***Cloud Computing, Secure Transactions, User Authorization, Validation Commit Protocol, Blow-fish Algorithm***

Introduction:

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.

The term "moving to cloud" also refers to an organization moving away from a traditional CAPEX model (buy the dedicated hardware and depreciate it over a period of time) to the OPEX model (use a shared cloud infrastructure and pay as one uses it).

Proponents claim that cloud computing allows companies to avoid upfront infrastructure costs, and focus on projects that differentiate their businesses instead of on infrastructure. Proponents also claim that cloud computing allows enterprises to get their applications up and running faster, with improved

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

manageability and less maintenance, and enables IT to more rapidly adjust resources to meet fluctuating and unpredictable business demand. Cloud providers typically use a "pay as you go" model. This can lead to unexpectedly high charges if administrators do not adapt to the cloud pricing model.Cloud computing exhibits the following key characteristics:
Cloud computing exhibits the following key characteristics:
Agility improves with users' ability to re-provision technological infrastructure resources.

Cost reductions claimed by cloud providers. A public-cloud delivery model converts capital expenditure to operational expenditure. This purportedly lowers barriers to entry, as infrastructure is typically provided by a third party and does not need to be purchased for one-time or infrequent intensive computing tasks. Pricing on a utility computing basis is fine-grained, with usage-based options and fewer IT skills are required for implementation (in-house). The e-FISCAL project's state-of-the-art repository contains several articles looking into cost aspects in more detail, most of them concluding that costs savings depend on the type of activities supported and the type of infrastructure available in-house.

Device and location independence enable users to access systems using a web browser regardless of their location or what device they use (e.g., PC, mobile phone). As infrastructure is off-site (typically provided by a third-party) and accessed via the Internet, users can connect from anywhere.

Maintenance of cloud computing applications is easier, because they do not need to be installed on each user's computer and can be accessed from different places.
Multitenancy enables sharing of resources and costs across a large pool of users thus allowing for: centralization of infrastructure in locations with lower costs (such as real estate, electricity, etc.) peak-load capacity increases (users need not engineer for highest possible load-levels) utilisation and efficiency improvements for systems that are often only 10–20% utilised.

Performance is monitored, and consistent and loosely coupled architectures are constructed using web services as the system interface.

Existing System:

To provide scalability and elasticity, cloud services oftenmake heavy use of replication to ensure consistent performance and availability. As a result, many cloud services rely on the notion of eventual consistency when propagating data throughout the system. This consistency model is a variant of weak consistency that allows data to be inconsistent among some replicas during the update process, but ensures that updates will eventually be propagated to all replicas.

Disadvantages of Existing System:

- Consistency problems can arise as transactional database systems are deployed in cloud environments and use policy-based authorization systems to protect sensitive resources.
- The system may suffer from policy inconsistencies during policy updates.
- It is possible for external factors to cause user credential inconsistencies over the lifetime of a transaction.

Two-Phase Commit (2PC) Algorithm

The 2-phase commit (2PC) protocol is a distributed algorithm to ensure the consistent termination of a transaction in a distributed environment. Thus, via 2PC a unanimous decision is reached and enforced among multiple participating servers whether to commit or abort a given transaction, thereby guaranteeing atomicity. The protocol proceeds in two phases, namely the prepare (or voting) and the commit (or decision) phase, which explains the protocol's name.

The protocol is executed by a coordinator process, while the participating servers are called participants. When the transaction's initiator issues a request to

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

commit the transaction, the coordinator starts the first phase of the 2PC protocol by querying—via prepare messages—all participants whether to abort or to commit the transaction. If all participants vote to commit then in the second phase the coordinator informs all participants to commit their share of the transaction by sending a commit message. Otherwise, the coordinator instructs all participants to abort their share of the transaction by sending an abort message. Appropriate log entries are written by coordinator as well as participants to enable restart procedures in case of failures.

As long as a transaction is still executing ordinary operations, coordinators as well as all participants operate in the Initial state. When the coordinator is requested to commit the transaction, it initiates the first phase of the 2PC protocol: To capture the state of the protocol's execution (which needs to be available in case of protocol restarts as explained below), the coordinator first forces a begin log entry, which includes a transaction identifier as well as a list of the transaction's participants, to a stable log. Afterwards, the coordinator sends a prepare message to every participant, enters the Collecting state and waits for replies.

Upon receiving a prepare message, a participant decides whether it is able to commit its share of the transaction. In either case, suitable log entries for later recovery operations as well as a prepared log entry indicating the vote ("Yes" or "No") are forced to a stable log, before a response message containing the vote is sent back to the coordinator. In case of a No-vote, the participant switches into the Aborted state and immediately aborts the transaction locally. In case of a Yesvote, the participant moves into the Prepared state. In the latter case the participant is said to be in doubt or blocked as it has now given up its local autonomy and must await the final decision from the coordinator in the second phase (in particular, locks cannot be released yet).

Once the coordinator has received all participants' response messages it starts the second phase of the 2PC protocol and decides how to complete the global transaction: The result is "Commit" if all participants voted to commit and "Abort"
otherwise. The coordinator then forces a commit or aborts log entry to the stable log, sends a message containing the final decision to all participants, and enters the corresponding state (Committed or Aborted).

Upon receipt of the decision message, a participant commits or aborts the local changes of the transaction depending on the coordinator's decision and forces suitable log entries for later recovery as well as a commit or abort log entry to a stable log. Afterwards, it sends an acknowledgment message to the coordinator and enters the corresponding final state (Committed or Aborted).

Once the coordinator has received all acknowledgment messages it ends the protocol by writing an end log entry to a stable log to enable later log truncation and enters the final state, Forgotten. (For multiple participants, the actions simply have to be duplicated; in case of abort, at least one of the participants votes "No", which implies that all occurrences of "commit" are replaced with "abort".)

Proposed System:

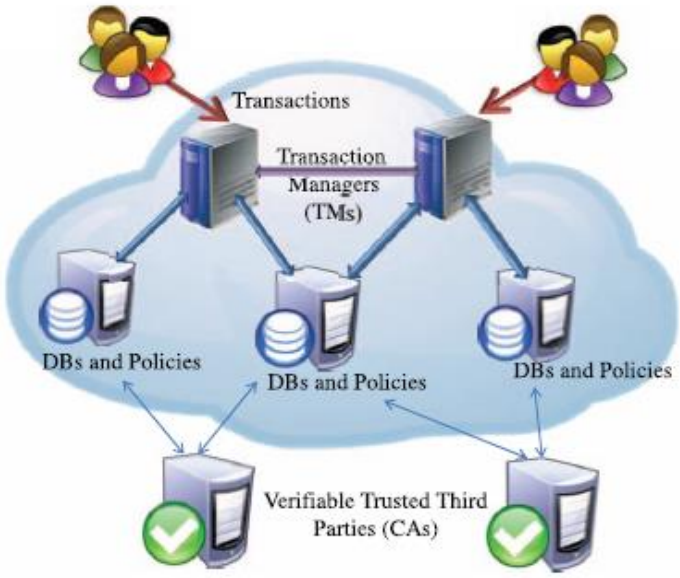
- We formalize the concept of trusted transactions.
- We define several different levels of policy consistency constraints and corresponding enforcement approaches that guarantee the trustworthiness of transactions executing on cloud servers.
- We propose a Two-Phase Validation Commit (2PVC) protocol that ensures that a transaction is safe by checking policy, credential, and data consistency during transaction execution.
- We carry out an experimental evaluation of our proposed approaches.

ISSN No: 2348-4845
International Journal & Magazine of Engineering, Technology, Management and Research
A Peer Reviewed Open Access International Journal

Advantages of Proposed System:

- Identifies transactions that are both trusted and conform to the ACID properties of distributed database systems.
- Guarantee the trustworthiness of transactions executing on cloud servers.
- A transaction is safe by checking policy, credential, and data consistency during transaction execution.
- Most suitable in various situations.

System Architecture:



Conclusion: Cloud computing poses privacy concerns because the service provider can access the data that is on the cloud at any time. A combination of algorithms that will enforce consistency, accuracy and precision of the authorization policies that increases the trustworthiness of the transactions has been identified. An attempt has been made to determine if the proposed approach will guarantee safe transactions.

References:

[1] Marian K. Iskander, Tucker Trainor, Dave W. Wilkinson, Adam J. Lee, and Panos K. Chrysanthis, "Balancing Performance, Accuracy, and Precision for Secure Cloud Transactions", VOL. 25, NO. 2, FEBRUARY 2014.

[2] Rabi Prasad Padhy, ManasRanjanPatra and Suresh Chandra Satapathy, (2011) Cloud Computing: Security Issues and Research Challenges

[3] BupeshMansukhani and Tanveer A. Zia, (2011) An empirical study of challenges in managing the security in cloud computing

[4] FarhadSoleimanianGharehchopogh and MeysamBahari, (2013) Evaluation of the Data Security Methods in Cloud Computing Environments

[5] Martin Kuehnhausen, Victor S. Frost and Gary J. Minden, (2012) Framework for Assessing the Trustworthiness of Cloud Resources, 142- 145

[6] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, and MateiZaharia, (2009) Above the Clouds: A Berkeley View of Cloud Computing

[7] Adam J. Lee and Marianne Winslett, Safety and Consistency in Policy Based Authorization Systems

[8] Dr.RaoMikkilineni and Vijay Sarathy, (2009) Cloud Computing and the Lessons from the Past, 57-62

[9] Abdullah Abuhussein, HarkeeratBedi and Sajjan Shiva, Evaluating Security and Privacy in Cloud Computing Services:A Stakeholder's Perspective

[10] Sang-Ho Na, Kyoung-Hun Kim and Eui-Nam Huh, (2013) A Methodology for Evaluating Cloud Computing Security Service-Level Agreements, 235-242