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 2, February 2015

Secure Data Transaction in Multi Cloud Using Two-Phase Validation

K.G.S. Venkatesan[1], N.G. Vijitha[2], R. Karthikeyan[3]

Research Scholar, Dept. of C.S.E., Bharath University, Chennai, India[1].

Department of computer Science & Engg., Bharath University, Chennai, Tamil Nadu, India[2].

Research Scholar, Dept. of C.S.E., Bharath University, Chennai, India[3].

ABSTRACT: In distributed transactional info systems deployed over cloud servers, entities work to make proofs of authorization that square measure even by collections of certified credentials. These proofs and credentials is also evaluated and picked up over extended time periods below the chance of getting the underlying authorization policies or the user credentials being in inconsistent states. It so becomes possible for policy-based authorization systems to form unsafe selections which may threaten sensitive resources. During this paper, we have a tendency to highlight the criticality of the matter. we have a tendency to then outline the notion of trusty transactions once addressing proofs of authorization. Consequently, we propose many progressively tight levels of policy consistency constraints, and gift completely different social control approaches to ensure the trustiness of transactions capital punishment on cloud servers. We have a tendency to propose a Two-Phase Validation Commit protocol as an answer, which is a changed version of the essential Two-Phase Commit protocols. We have a tendency to finally analyse the various approaches bestowed victimization each analytical evaluation of the overheads and simulations to guide the choice manufacturers to that approach to use.

KEYWORDS: Cloud databases, authorization policies, consistency, distributed transactions, atomic commit protocol.

I. INTRODUCTION

Cloud computing has recently emerged as a computing paradigm within which storage and computation is outsourced from organizations to next generation knowledge centres hosted by corporations like Amazon, Google, Yahoo, and Microsoft. Such corporations facilitate free organizations from requiring pricey infrastructure and experience in-house, and instead build use of the cloud suppliers to keep up, support, and broker access to high-end resources. From AN economic perspective, cloud customers will save huge IT capital investments and be charged on the idea of a pay-only-for-what you- use rating model. One of the foremost appealing aspects of cloud computing is its snap, that provides AN illusion of infinite, on demand resources creating it a beautiful atmosphere for highly-scalable, multi-tiered applications. However, this can produce further challenges for back-end, transactional database systems, that were designed while not snap in mind. Despite the efforts of key-value stores like Amazon's Simple DB, Dynamo, and Google's big table to supply climbable access to large amounts of knowledge, transactional guarantees remain a bottleneck.

To provide measurability and physical property, cloud services typically make serious use of replication build sure to confirm consistent performance and convenience. As a result, several cloud services trust on the notion of ultimate consistency once propagating information throughout the system. This consistency model could be a variant of weak consistency that permits information to be inconsistent among some replicas throughout the update method, however ensures that updates can eventually be propagated to any or all replicas. This makes it difficult to strictly maintain the ACID guarantees, as the 'C' (consistency) a part of ACID is sacrificed to supply reasonable convenience.

In systems that host sensitive resources, accesses square measure protected via authorization policies that describe the conditions under that users ought to be allowable access to resources. These policies describe relationships between the system principals, as well because the certified credentials that user should provide to attest to their attributes. In an exceedingly transactional information system that's deployed in an exceedingly extremely distributed and elastic

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 2, February 2015

system such as the cloud, policies would usually be replicated very much like data among multiple sites, typically following the same weak or ultimate consistency model. It so becomes attainable for a policy-based authorization system to make unsafe choices victimization stale policies [3].

Interesting consistency issues will arise as transactional database systems area unit deployed in cloud environments and use policy-based authorization systems to shield sensitive resources. Additionally to handling consistency problems amongst database replicas, we have a tendency to should conjointly handle 2 sorts of security in consistency conditions. First, the system might sure from policy inconsistencies throughout policy updates thanks to the relaxed consistency model underlying most cloud services. For example, it's doable for many versions of the policy to be observed at multiple sites among one dealing, leading to inconsistent (and seemingly unsafe) access selections throughout the dealings. Second, it's doable for external factors to cause user certification inconsistencies over the life of a transaction. for example, a user's login credentials may be nullified or revoked once assortment by the authorization server, however before the completion of the dealings. In this paper, we tend to address this confluence of information, policy, and written document inconsistency issues which will emerge as transactional database systems are deployed to the cloud [4].

II. RELATED WORK

Relaxed Consistency Models for the Cloud, Several info solutions are written to be used at intervals the cloud surroundings. For instance, Amazon's generator info [14]; Google's Big Table storage system [15]; Face book's Cassandra [16]; and Yahoo!'s PNUTS [17]. The common thread between every of those custom knowledge models is BASE with a relaxed notion of consistency provided so as to support massively parallel environments.

Such a relaxed consistency model adds a replacement dimension to the quality of the look of enormous scale applications and introduces a replacement set of consistency issues [18]. In [19], the authors bestowed a model that permits queries to precise consistency and concurrency constraints on their queries that can be implemented by the software at runtime. On the opposite hand, [20] introduces a dynamic consistency parcelling mechanism which mechanically adapts the amount of consistency at runtime. Both of those works specialize in information consistency, while our work focuses on attaining each information and policy consistency.

A. *Reliable Outsourcing*

Security is taken into account one in all the major obstacles to a wider adoption of cloud computing. Particular attention has been given to shopper security because it relates to the correct handling of outsourced knowledge. As an example, proofs of information possession are planned as a method for shoppers to confirm that service suppliers truly maintain copies of the information that they're contractile to host [21]. In other works, knowledge replication has been combined with proofs of irretrievability to produce users with integrity and consistency guarantees once exploitation cloud storage [22], [23].

B. *Distributed Transactions*

Cloud TPS provides full ACID properties with a ascendable dealing manager designed for a No SQL surroundings [26]. However, Cloud TPS is primarily concerned with providing consistency and isolation upon information without relation to concerns of authorization policies. There has additionally been recent work that focuses on providing some level of guarantee concerning the connection between information and policies . This work proactively ensures that information. Stored at a selected website conforms to the policy hold on at that site. If the policy is updated, the server can scan the information items and throw out any that may be denied supported the revised policy. it's obvious that this can cause AN eventually consistent state wherever information and policy adapt, however this work only issues itself with native consistency of one node, not with transactions that span multiple nodes [6].

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 2, February 2015

C. Distributed Authorization

The consistency of distributed proofs of authorization has antecedent been studied, though not in a very dynamic cloud surroundings (e.g., [4]). This work highlights the inconsistency problems that may arise within the case where authorization policies are static, however the credentials used to satisfy these policies could also be revoked or altered. The authors develop protocols that alter numerous consistency guarantees to be enforced throughout the proof construction method to attenuate these styles of security problems.

III. AUTONOMOUS SYSTEM

System Assumptions and drawback Definition

A. *System Model*

Fig.1 illustrates the interaction among the elements in our system. We tend to assume a cloud infrastructure consisting of a collection of S servers, wherever every server is to blame for hosting a subset D of all information things D happiness to a particular application domain (D D). Users act with the system by submitting queries or update requests encapsulated in ACID transactions. A dealing is submitted to a dealing Manager (TM) that coordinates its execution. Multiple TMs might be invoked as the system employment will increase for load reconciliation, but each transaction is handled by just one metal [7].

We denote every group action as T = q1; q2…., qn, where qi 2 Q could be single query/update happiness to the set of all queries letter. The start time of every group action is denoted by (T), and the time at that the group action finishes execution and is prepared to commit is denoted by !(T). We have a tendency to assume that queries happiness to a group action execute consecutive, which a group action does not fork sub-transactions. These assumptions change our presentation; however don't act the correctness or the validity of our consistency definitions [8].

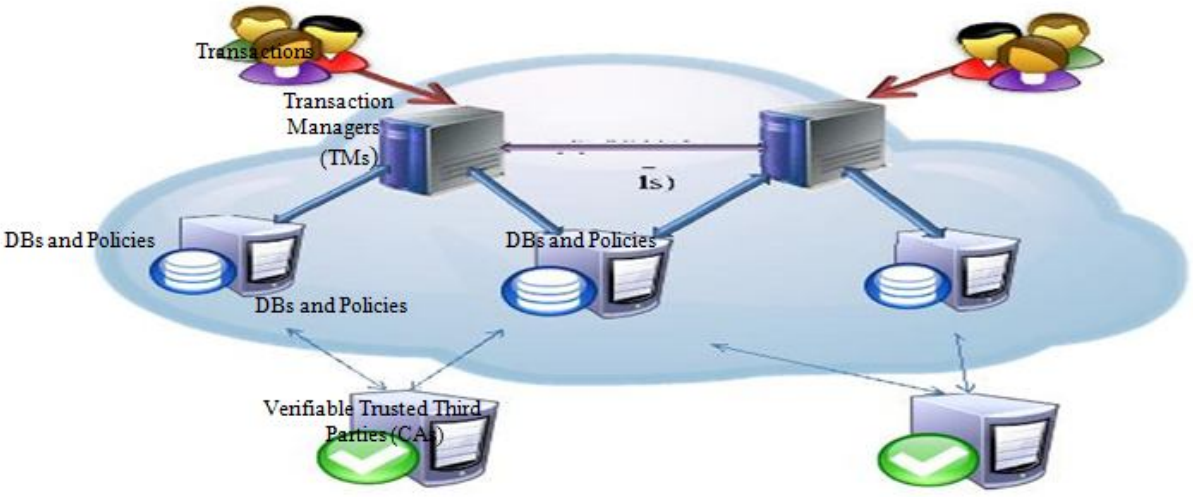

Fig. 1. Communication between the System Components

Let P denote the set of all authorization policies, and let P_{si} (D) denote the policy that server s_i uses to guard information item D. we tend to represent a policy P as a mapping $P : S \times 2^D \rightarrow 2R \times A \times N$ that associates a server and a group of knowledge things with a set of logical thinking rules from the set R, a policy administrator from the set A, and a version range. We have a tendency to denote by C the set of all credentials, that square measure issued by the

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 2, February 2015

Certificate Authorities (CAs) at intervals the system. We have a tendency to assume that every CA an internet methodology that permits any server to ascertain the current standing of credentials that it's issued [5]. Given a credential ck two C, (ck) and! (ck) denote issue and expiration times of ck, severally. Given a perform m : Q ! second that identifies the info things accessed by a selected question, a proof of authorization for question *qi* evaluated at server sj at time tk may be a tipple ‹qi; s j; Ps j (m(qi)); tk;C,› wherever C is that the set of credentials given by the querier to satisfy postscript j (m(qi)). In this paper, we have a tendency to use the perform eval : F × TS →B to denote whether a symptom f ∈F is valid at time t ∈TS. To enhance the final pertinence of the consistency models developed during this paper, the on top of formalism is by choice opaque with reference to the policy and credentials formats wont to implement the system. for example, this formalism may simply be wont to model the employment of XACML policies [6] because the set of logical thinking rules R, and ancient (e.g., X.509 [7]) credentials for the set C. On the opposite hand, it also can model the employment of a lot of advanced trust management policies (e.g., [8], [9]) for the logical thinking rules R, and therefore the use of privacy-friendly credentials (e.g., [10], [11]) for the set C.

B. Downside Definition

Since transactions area unit dead over time, the state data of the credentials and therefore the policies implemented by different servers are subject to changes at any instance of your time, thus it becomes vitalto introduce precise definitions for the different consistency levels that might be achieved at intervals a transactions period. These consistency models strengthen the trusted dealing definition by process the setting in which policy versions area unit consistent relative to the remainder of the system. Before we tend to try this, we tend to outline a transaction's read in terms of the different proofs of authorization evaluated throughout the period of a selected dealing [10].

Definition 1: (View) A transaction's read Vermont is that the set 3 of proofs of authorization determined throughout the period of a transaction [α(T), ω(T)] and outlined as VT = {f | f =hqi, si, Psi(m(qi)), ti,Ci ∧qi ∈T}
Following from Def. 1, a transaction's read is made incrementally as a lot of proofs of authorization square measure being evaluated by servers throughout the dealing execution. we have a tendency to currently gift two {increasingly progressively more and a lot of} more powerful definitions of consistencies within transactions [9].

Definition 2: (View Consistency) A read VT = fhqi; si; Psi (m(qi)); ti;Ci; : : : ; hqn; sn; Psn (m(qn)); tn;Cig is view consistent, or --consistent, if VT satisfies a predicate --consistent that places constraints on the versioning of the policies such --consistent(VT) $ 8i; j : ver(Psi) = ver(Ps j) for all policies happiness to an equivalent administrator A, where function ver is outlined as ver : P ! N.

Definition 3: (Global Consistency) V^T = { $\langle q_i,s_i,P_s(m(q_i)),t_i,C\rangle$,..., $\langle q_n, s_n, P_s(m(q_n)),t_n,C\rangle$ } is global consistent, or -consistent, if American state satisfies a predicate -consistent that places constraints on the versioning of the policies specified–consistent(V^T)↔$\forall_i$:ver(P_si)=ver(P) for all policies happiness to constant administrator A, and function ver follows constant same definition, while ver.(P) refers to the most recent policy version [12].

With a world consistency model, policies accustomed measure the proofs of authorization throughout a dealing execution among S servers ought to match the most recent policy version among the entire policy set P, for all policies implemented by a similar administrator A.
Given the on top of definitions, we tend to currently have a particular vocabulary for defining the conditions necessary for a dealing to be declared as "trusted".

Definition 4: (Trusted Transaction) Given a group action T ={q_1,q_2,...,q_n} and its corresponding read T is trust worthy $\forall f_s$:eval(f_si $^{,t),}$ at your time instance $^{t\,:\,\alpha(T)\,\le\,t\,\le}\omega(T)$ ∧(φ-consistent(V^T) ∨ψ-consistent(V^T))

Finally, we are saying that dealing is safe if it's a trustworthy transaction that additionally satisfies all information integrity constraints obligatory by the management system. a secure dealings is allowed to commit, whereas Associate in Nursing unsafe dealings is forced to rollback [11].

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 2, February 2015

C. Progressive timely Proofs of Authorization

Before we tend to outline the progressive timely proofs of authorization approach, we tend to outline a read instance that could be a read snapshot at a particular instance of your time.

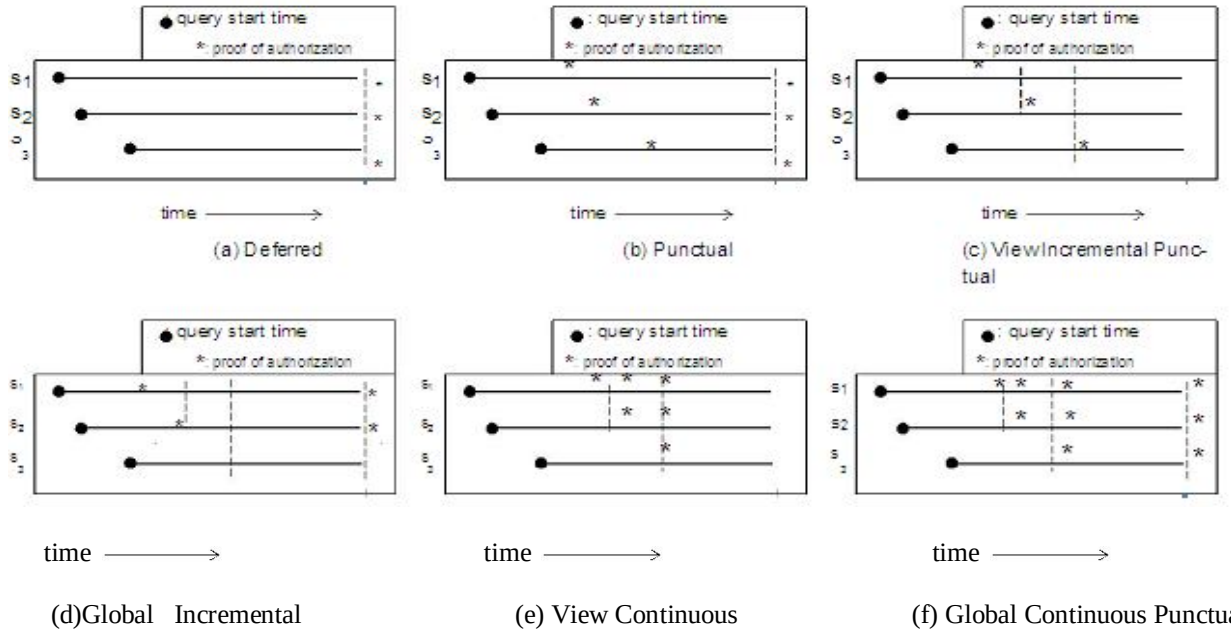

Fig. 2. Different proofs of authorization

Definition 7: : (View Instance) A read instance $V^T \subseteq V^T$ is defined as $V^T = \{f_s \mid f_s = \langle q_i, s_i, PAsi\ (m(q_i)), t, C\rangle \in V^T \wedge \leq t_i\}, \forall t, t_i : \alpha(T) \leq t \leq t_i \leq \omega(T)$.
Informally, a read instance Vermont ti is that the set of all proofs of authorization evaluated by servers concerned in group action T up till the time instance ti.

Definition 8: (Incremental timely Proofs of Authorization) Given a dealing T and its corresponding read VT , T is trusted underneath the progressive timely proofs of authorization approach, I at any time instance $t_i : \alpha(T) \leq t_i \leq \omega(T)$, $\forall fsi$: eval(f_s ,t_i) (φ-consistent(V^T) ∨ ψ-consistent(V^T))
Incremental on time proofs develop a stronger notion of trusted transactions, as dealing isn't allowed to proceed unless every server achieves the specified level of the policy consistency with all previous servers. this means that every one participating servers are going to be forced to possess a regular read with the primary death penalty server unless a more modern policy version shows up at a later server, during which case the dealing aborts.

D. Implementing Safe Transactions

A safe dealing may be a dealing that's each sure (i.e., satisfies the correctness properties of proofs of authorization) and info correct (i.e., satisfies the information integrity constraints). We 1st describe Associate in nursing rule that enforces sure transactions, and so expand this rule to enforce safe transactions. Finally, we tend to show however these algorithms will be used to implement the approaches mentioned [14].

Two-Phase Validation rule
A common characteristic of most of our papered approaches to achieve sure transactions is that they would like for policy consistency validation at the tip of a group action. That is, so as for a sure group action to commit, its Tm has

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 2, February 2015

got to enforce either view or international consistency among the servers taking part in the group action. Toward this, we have a tendency to propose a replacement rule called Two-Phase Validation (2PV).

As the name implies, 2PV operates in 2 phases: assortment and validation. Throughout assortment, the metal initial sends a Prepare to- Validate message to every participant server. In response to this message, every participant (1) evaluates the proofs for every query of the group action victimisation the most recent policies it's accessible and (2) sends a reply back to the metal containing the reality value (TRUE/FALSE) of these proofs alongside the version number and policy symbol for every policy used. Further, each participant keeps track of its reply (i.e., the state of every query) which incorporates the id of the metal (TMid), the id of the transaction (Tid) to that the question belongs, and a group of policy versions utilized in the query's authorization (vi; pi).

Algorithm 1: Two-Phase Validation Commit - 2PVC (TM)

1 Send "Prepare-to-Commit" to all or any participants
2 expect all replies (Yes/No, True/False, and a collection of policy versions for every distinctive policy)
3 If any participant replied No for integrity check
4 ABORT
5 determine the most important version for all distinctive policies
6 If all participants utilize the most important version for every unique policy
7 If any responded False
8 ABORT
9 Otherwise
10 COMMIT
11 Otherwise, for participants with previous policies
12 Send "Update" with the most important version number of every policy
13 expect all replies
14 Goto 5

Punctual can come back proof evaluations upon capital punishment every query. Nonetheless this can be done on one server, and so will not would like 2PVC or 2PV to distribute the choice. To provide for trusty transactions, each need a commit-time analysis at all participants victimization 2PVC [12].

Incremental punctual proofs square measures lightly different. As queries square measure dead, the metallic element should conjointly check for consistency within the taking part servers. Hence, a variant of the basic 2PV protocol is employed throughout the dealings execution. For read consistency, the metallic element must check the version number it receives from every server there with of the terribly first taking part server. If they're different, the dealing aborts attributable to a consistency violation. At commit time, the entire proofs can are generated with consistent policies and only 2PC is invoked. Within the international consistency case, the TM needs to validate the policy versions used against the newest policy version identified by the master policies server to choose whether to abort or not. At commit time, 2PVC is invoked by the metallic element to envision the info integrity constraints and verify that the master policies server has not received any newer policy versions [15].

Finally, Continuous proofs square measure the foremost concerned. Unlike the case of progressive prompt in an exceedingly read consistency, Continuous proofs invoke 2PV at the execution of every question which will update the older policies with the new policy and re-evaluate. Once a question is requested, its thulium can (1) execute 2PV to validate authorizations of all queries up to the present point, and (2) upon CONTINUE being the choice of 2PV, submit consecutive question to be dead at the suitable server, otherwise the dealings aborts. Identical actions occur underneath global consistency with the exception that the newest policy version is employed as known by the master policy server.

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 2, February 2015

IV. EVALUATIONS

A. *Experiment and Setup*

We used Java to implement every proof approach delineated in Sec. three with support for each read and world consistency. Although the approaches were enforced in their totality, the underlying info and policy social control systems were simulated with parameters. To understand the performance implications of the different approaches, we varied the (i) protocol used, (ii) level of consistency desired, (iii) frequency of master policy updates, (iv) transaction length, and (v) variety of servers accessible [17].

Our experimentation framework consists of 3 main components: a irregular dealings generator, a master policy server that controls the propagation of policy updates, and an array of dealings process servers.Our experiments were run among an enquiry research lab consisting of thirty eight Apple macintosh mini computers. These machines were running OS X ten.6.8 and had one.83 giga cycle per second Intel Core couple processors including 2GB of RAM. All machines were connected to a gigabit local area network LAN with average trip times of 0.35 ms. All WAN experiments were conjointly conducted within this tested by artificial means delaying packet transmission by an extra seventy five ms [18].

For each simulation and every potential combination of parameters, a thousand transactions were run to collect average statistics on dealing process delays evoked by the particular protocol and system parameter decisions. The irregular transactions were haphazardly composed of information reads and writes with equal chance [19]. To simulate policy updates at different servers, the master policy server picks a random taking part server to receive the updates.

Given that our interest during this article lies in exploring the average performance of every of the different approaches, we made few assumptions to modify the experimentations and help limit the influence of different factors on dealings execution time. Specifically, we tend to assume the existence of one master policy server that must be consulted for the most recent policy version happiness to a particular policy administrator. This simplifies the 2PV protocol and reduces the quantity of changed messages to comprehend the most recent version [20].

V. SIMULATION

Using 1, and 2, we tend to plot three and four to indicate our simulation results for each the local area network arrangement and therefore the simulated WAN, severally. Every figure shows the execution time of the committed dealings (y-axis) because the chance of the policy update changes (x-axis).The figures distinction between the four different approaches for proofs of authorization each with the 2 validation modes, namely, read and international consistency. The figures show different transactions length: (a) short transactions involve 8–15 operations running on up to 5 servers, (b) medium transactions involve 16–30 operations running on up to fifteen servers, and (c) long transactions involve 31–50 operations running on up to twenty five servers [29]. For every case, and as a baseline, we have a tendency to measured the dealings execution time when transactions execute with none proof of authorization and square measure terminated exploitation the essential 2PC (shown in figures as a solid line concerning postponed 2PC only). All told cases, the average dealings execution time of postponed proofs with 2PVC was effectively an equivalent because the baseline indicating that 2PVC has negligible overhead over the essential 2PC [27].

The relative performance of the different proofs of authorization is consistent throughout the different experiments. From the figures, we are able to conclude that the delayed proofs have the most effective performance of all, because the group action operations are allowed to proceed while not interruption till commit time. Of course, proofs of authorization failing at commit time will force the group action to travel into a doubtless costly rollback. Which will not be the case with the opposite schemes, as the proofs area unit evaluated earlier throughout the execution of the transactions and also the rollback method of aborted transactions involves fewer operations [22].
Punctual proofs return next in terms of performance [26]. The Minor difference between punctual and delayed proofs is because punctual proofs incur the price for the native authorization checks every of that is within the vary of 3-5 ms. Both Deferred and punctual proofs square measure on the average insensitive to the chance of policy updates (as accomplished from the graph slope). This is often thanks to the very fact that each scheme solely enforce consistency at commit time [25].

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 2, February 2015

VI. CONCLUSION AND FUTURE WORK

Despite the recognition of cloud services and their wide adoption by enterprises and governments, cloud suppliers still lack services that guarantee each knowledge and access management policy consistency across multiple knowledge centres [31]. During this article, we identified many consistency issues which will arise throughout cloud-hosted dealings process victimization weak consistency models, notably if policy-based authorization systems square measure used to enforce access controls. to the current finish, we tend to developed a variety of light-weight proof social control and consistency models i.e., Deferred, Punctual, progressive, and Continuous proofs, with read or world consistency that will enforce more and more strong protections with lowest runtime overheads [32]. We used simulated workloads to by experimentation value implementations of our projected consistency models relative to three core metrics: dealings process performance, accuracy (i.e., global vs. read consistency and recency of policies used), and exactness (level of agreement among dealings participants). we tend to found that top performance comes at a cost: delayed and prompt proofs had lowest overheads, but did not discover bound sorts of consistency issues [30]. On the other hand, high accuracy models (i.e., progressive and Continuous) needed higher code quality to implement correctly, and had solely moderate performance in comparison to the lower accuracy schemes. To higher explore the differences between these approaches, we tend to conjointly administered a trade-o analysis of our schemes parenthetically however application-centric requirements influence the pertinence of the eight protocol variants explored during this article.

VII. ACKNOWLEDGEMENT

The author would like to thank the Vice Chancellor, Dean-Engineering, Director, Secretary, Correspondent, HOD of Computer Science & Engineering, Dr. K.P. Kaliyamurthie, Bharath University, Chennai for their motivation and constant encouragement. The author would like to specially thank **Dr. A. Kumaravel** for his guidance and for critical review of this manuscript and for his valuable input and fruitful discussions in completing the work and the Faculty Members of Department of Computer Science & Engineering. Also, he takes privilege in extending gratitude to his parents and family members who rendered their support throughout this Research work.

REFERENCES

1. M. Myers, R. Ankney, A. Malpani, S. Galperin, and C. Adams, "X.509 internet public key infrastructure online certificate status protocol", RFC 2560, June - 1999, http: //tools.ietf.org/html/rfc5280.
2. S. Das, D. Agrawal, and A. El Abbadi, "Elastras: an elastic transactional data store in the cloud", in USENIX Hot Cloud, 2009.
3. D. J. Abadi, "Data management in the cloud: Limitations and opportunities", IEEE Data Engineering Bulletin, Mar. 2009.
4. A. J. Lee and M. Winslett, "Safety and consistency in policy-based authorization systems", in ACM CCS, 2006.
5. K.G.S. Venkatesan. Dr. V.Khanna, Dr. A.Chandrasekar, "Autonomous system for mesh network by using packet transmission & failure detection", Inter. Journal of Innovative Research in computer & comm. Engineering, Vol. 2, Issue 12, December -2014.
6. E. Rissanen, "extensible access control markup language (xacml) version 3.0," Jan. 2013, http: //docs.oasis-open.org/xacml/3.0/xacml-3.0-core-spec-os-en.html.
7. K.G.S. Venkatesan and M. Elamurugaselvam, "Design based object oriented Metrics to measure coupling & cohesion", International journal
   a. of Advanced & Innovative Research, Vol. 2, Issue 5, pp. 778 – 785, 2013.
8. D. Cooper et al., "Internet x.509 public key infrastructure certificate and certificate revocation list (crl) profile," RFC 5280, May 2008, http://tools.ietf.org/html/rfc5280.
9. Sathish Raja and K.G.S. Venkatesan, "Email spam zombies scrutinizer in email sending network infrastructures", International journal of Scientific & Engineering Research, Vol. 4, Issue 4, PP. 366 – 373, April 2013.
10. K.G.S. Venkatesan. Dr. V.Khanna, "Inclusion of flow management for Automatic & Dynamic route discovery system by ARS ", Inter. Journal of Advanced Research in computer science & software Engineering, Vol. 2, Issue 12, pp. 1-9, December -2012.
11. J. Li, N. Li, and W. H. Winsborough, "Automated trust negotiation using cryptographic credentials", in ACM CCS, Nov. 2005.
12. K.G.S. Venkatesan. K.P. Shyamraj, S. Anand, "SAT : A Security Architecture in wireless mesh networks ", International Journal of Advanced Research in computer science & software Engineering, Vol. 3, Issue 4, pp. 325-331, April -2013.
13. M. Armbrust et al., "Above the clouds: A berkeley view of cloud computing", University of California, Berkeley, Tech. Rep., Feb- 2009.
14. Selvakumari.P, K.G.S. Venkatesan, "Vehicular communication using Fvmr Technique" , International Journal of Applied Engineering Research, Vol. 9, Issue 22, PP. 6133 – 6139, 2014.
15. L. Bauer et al., "Distributed proving in access-control systems," in Proc. of the IEEE Symposium on Security and Privacy, May 2005.
16. K.G.S. Venkatesan, "Automatic detection & control of Malware spread in decentralized peer to peer network", International Journal of Innovative Research in computer & comm. Engineering, Vol. 1, Issue 7, September -2013.
17. P. Indira Priya, K.G.S. Venkatesan, "Finding the K-Edge connectivity in MANET using DLTRT", International Journal of Applied
18. Engineering Research, Vol. 9, Issue 22, PP. 5898 – 5904, 2014.
19. K.G.S. Venkatesan, "Planning in FARS by dynamic multipath reconfiguration system failure recovery in wireless mesh network", International Journal of Innovative Research in computer & comm. Engineering, Vol. 2, Issue 8, August -2014.

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 2, February 2015

20. J. Camenisch and A. Lysyanskaya, "Anecient system for non-transferable anonymous credentials with optional anonymity revocation," in EUROCRYPT, 2001.
21. K.G.S. Venkatesan, D. Priya, "Secret-Key generation of Mosaic Image", Indian Jourl. of Applied Researh., Vol. 3, Issue 6, PP. 164-166, 2013
22. Needhu.C, K.G.S. Venkatesan, "A System for Retrieving Information directly from online social network user Link", International Journal of Applied Engineering Research, Vol. 9, Issue 22, PP. 6023 – 6028, 2014.
23. P. K. Chrysanthis, G. Samaras, and Y. J. Al-Houmaily, "Recovery and performance of atomic commit processing in distributed database systems," in Recovery Mechanisms in Database Systems. PHPTR, 1998.
24. K.G.S. Venkatesan. Dr. V.Khanna, "Inclusion of flow management for Automatic & dynamic route discovery system by ARS", International Journal of Advanced Research in computer science & software Engg., Vol.2, Issue 12, PP. 1 – 9, December – 2012.
25. Annapurna Vemparala, K.G.S. Venkatesan, "A Reputation based scheme for routing misbehavior detection in MANET"S ", International Journal of computer science & Management Research, Vol. 2, Issue 6, June - 2013.
26. K.G.S. Venkatesan, R. Resmi, R. Remya, "Anonymizimg Geographic routing for preserving location privacy using unlinkability and unobservability", International Journal of Advanced Research in computer science & software Engg., Vol. 4, Issue 3, PP. 523 – 528, March – 2014.
27. Selvakumari. P, K.G.S.Venkatesan, "Vehicular communication using Fvmr Technique", International Journal of Applied Engineering Research, Vol. 9, Issue 22, PP. 6133 – 6139, 2014.
28. K.G.S. Venkatesan, G. Julin Leeya, G. Dayalin Leena, "Efficient colour image watermarking using factor Entrenching method", International Journal of Advanced Research in computer science & software Engg., Vol. 4, Issue 3, PP. 529 – 538, March – 2014.
29. L. A. Laranjeira and G. N. Rodrigues, "Extending the reliability of wireless sensor networks through informed periodic redeployment." in SERE. IEEE, 2012, pp. 167–176.
30. K.G.S. Venkatesan. Kausik Mondal, Abhishek Kumar, "Enhancement of social network security by Third party application", International Journal of Advanced Research in computer science & software Engg., Vol. 3, Issue 3, PP. 230 – 237, March – 2013.
31. Annapurna Vemparala, K.G.S. Venkatesan, "Routing Misbehavior detection in MANET'S using an ACK based scheme", International Journal of Advanced & Innovative Research, Vol. 2, Issue 5, PP. 261 – 268, 2013.
32. K.G.S. Venkatesan. Kishore, Mukthar Hussain, "SAT : A Security Architecture in wireless mesh networks", International Journal of Advanced Research in computer science & software Engg., Vol. 3, Issue 3, PP. 325 – 331, April – 2013.
33. F. Fabbri, C. Buratti, and R. Verdone, "A multi-sink multi-hop wireless sensor network over a square region: Connectivity and energy consumption issues", in GLOBECOM Workshops, 2008 IEEE, PP. 1–6, November – 2008.