CSIT-2013

Proceeding of The 2013 International Conference on Computer Science and Information Technology

16- 18 June 2013

Jogjakarta, Indonesia

Universitas Teknologi Yogyakarta (UTY)

K-Innovation

ICT BUSINESS CENTRE

Managing Knowledge,Generating Innovation through E-Management

DaEng

Data Engineering Research Center

FAULT TOLERANCE ON BINARY VOTE ASSIGNMENT CLOUD QUORUM (BVACQ) REPLICATION TECHNIQUE

A.Noraziah, Ainul Azila Che Fauzi, A.Fairuzullah, Nurzety Aqtar, Azlina Zinuddin
Faculty of Computer Systems & Software Engineering,
University Malaysia Pahang,
26300, Kuantan, Pahang, Malaysia
e-mail: noraziah@ump.edu.my, ainulazila@yahoo.com
Phone number: +6095492121

Tutut Herawan
Department of Information Science
Faculty of Computer Science & Information Technology
University of Malaya, 50603 Kuala Lumpur, Malaysia
tutut@um.edu.my

***Abstract*—Replication provides user with fast, local access to shared data, protects availability of applications and support fault tolerance because alternate data access options exist. In this paper, we will manage availability of data replication during failure cases in Cloud using a new proposed algorithm called Binary Vote Assignment on Cloud Quorum (BVACQ). This technique combines the replication and fault tolerant mechanism. The result shows that managing replication and fault tolerance through proposed BVACQ able to preserve data consistency. It also increases the degrees of data availability. This is because the missing data from during the server failure has been reconciled and replicated after that server recovered.**

***Index Terms*—Data replication, BVACQ, quorum, fault tolerance, Cloud Computing**

I. INTRODUCTION

Cloud Computing is described as enabling convenient, on-demand network access to shared pool of configurable computer resources that released with minimal management effort according to National Institute of Standards and Technology Information Technology Laboratory. Question has been raised, is it Cloud Computing is just a new name for Grid? There is a common need to be able to manage large facilities. Due to the scalability issue and relevant to these scenarios, researcher becomes concern on what will happen in the event of disaster. The ability to preserve the delivery of expected services despite the presence of fault caused errors within system itself is called fault tolerance [1]. One way to preserve data availability and reliability is through the replication mechanism. Research problems arise, how are mechanisms done to maintain availability of replicated data when faults occur in the event of disasters? Lazy Update Propagation is proposed in Cloud, called lazy master replication [2, 3]. The updates are propagated towards the other replicas which are called secondary copies [4]. The problem with this technique is it uses asynchronous replication. Thus, it cannot support critical data in real-time cloud computing applications. Another research problems arise, how are mechanisms done in order to maintain the availability and consistency of replicated data in the event of disaster? This is because, there is still several algorithm exists on how to manage real time replication and fault tolerant mechanism to preserve data availability in the event of disaster in Cloud through synchronous replication. This will jeopardize the data consistency and may produce incorrect results. BVAG Transaction Semantic combines replication and transaction [5], which considers data grid and not the Cloud. Therefore, we proposed a new replication techniques namely Binary Vote Assignment on Cloud Quorum (BVACQ) which is based from BVAG techniques to manage replication and fault tolerance in Cloud environment. In the event of disaster, some changes may have been made and others are not. This paper proposed a new fault tolerance on BVACQ replication control protocol, which able to manage replication data availability even in the failure cases in Cloud.

The rest of the paper is organized as follows: Section 2 is the Literature Review. Section 3 describes system model of the proposed BVACQ algorithm with example case. Finally, we conclude this paper in the Section 4.

II. LITERATURE REVIEW

2.1 Fault Tolerance in Cloud Computing

Cloud computing has become a hot topic of research among academic and industry community since nowadays everyone is preparing to migrate to cloud. This is because Cloud Computing brings advantages in many fields such as education, business and engineering. There are many more exciting areas of development in Cloud Computing with new ideas as well as hybrid of old ones being deployed for production as well as research systems. There is question has been asked, is Cloud Computing just a new name for Grid? But there is no straightforward answer to such questions. The vision is the same which is to reduce the cost

of computing, increase reliability, and increase flexibility by transforming computers from something that we buy and operate ourselves to something that is operated by a third party. But, things are different now than they were 10 years ago [7]. The prospect of needing only a credit card to get on-demand access to computers in tens of data centres distributed throughout the world where resources that be applied to problems with massive, potentially distributed data, is exciting. So grid and cloud are operating at a different scale, and operating at these new, more massive scales can demand fundamentally different approaches to tackling problems. Nevertheless, the problems are mostly the same in Clouds and Grids [22, 23].

There is a common need to be able to manage large facilities. Due to the scalability issue and relevant to this scenarios, researcher becomes concern on what will happen in the event of disaster, as such when the system failure. There are several types of failure such as a server omits to respond to a request, a server responds incorrectly to a request, returns the wrong value or makes incorrect state transition. Besides, there are also failures where a server does not respond in the specified real-time interval which means the server responds too late or responds too early. The other failures that might happen are a server repeatedly fails to respond to requests until being restarted, the server restarts in initial state, some part of the state is as before the crash while the remainder is reset to initial state, the server restarts in the state before the crash and the server never restarts [8]. The only way to solve these failures is by enables fault tolerance.

There are several techniques for fault tolerance on replication grid such as check-pointing, scheduling, optimistic temporal replication fault tolerant protocol, and Decentralized Grading Autonomous Selection (DGAS) middleware. Check-pointing is the process of saving the state of a running application to stable storage [8]. The save state can be used to resume the execution of the application from the point in the computation where the check point was last taken without restart the application from its very beginning when the failure occur. In scheduling, there is a fault tolerance mechanism called eager scheduling that use for load balancing and failure masking [10]. As long as the task's result has not been returned, it will reschedule a task to idle processors. The crashed will be handled without require of detecting them. In [9] have proposed an optimistic temporal replication fault tolerant protocol for transactional mobile agent based on check pointing, chain control and message passing to detect and recover failed agent. In [10] presented a novel middleware, Decentralized Grading Autonomous Selection (DGAS) that enables decentralized decision making for fault tolerant service execution in pervasive systems. By using the DGAS, each device can computes its success probability and decides whether to execute the service replica. The ability to preserve the delivery of expected services despite the presence of fault caused errors within the system itself is called fault tolerance [1, 21]. It aims at the avoidance of failures in the presence of faults. A fault tolerant service detects errors and recovers from them without participation of any external agents, such as humans. Errors are detected and corrected and permanent faults are located and removed while the system continues to deliver acceptable services. Strategies to recover from errors include roll-back, which implies bringing the system to a correct state saved before the error occurred, roll forward, i.e. bringing the system to a fresh state without errors, or compensation, i.e. masking an error, in situations when the system contains enough redundancy to do that. Hence fault tolerance could be considered as the survival attribute of computer systems. There is one important method to achieve fault tolerance in grids which is replication. Data replication is one of the technique or key components in data grid to increase availability and reliability of the data [22, 23]. Moreover, it also can reduce access delay, bandwidth consumption [11], fault tolerance [1, 12, 13, 14, and 15] and load balancing [14]. To speed up data access for data grid systems, data can be replicated in multiple locations, so that a user can access the data from nearby locations [15].

2.2 Data Replication

Organizations need to provide current data to users who may be geographically remote and request distributed data around multiple sites in data grid [12]. The challenges will be faced to those who design, maintain and manage the data is in ensuring the efficient access of replicated data to such a huge network and widely distributed. In managing replication some of the issues must be considers such as strategies of replication, replica selection strategies to find the best-fit replica, replica consistency and replica location mechanism. Replication strategies determine when and where to create a replica, taking into account of the factors such as request number of the data, network conditions, storage availability of nodes, etc [13, 14, 18, 19, 20].

Read-One-Write-All (ROWA), Branch Replication Scheme (BRS), Hierarchical Replication Scheme (HRS) are the example of existing replication techniques. ROWA technique has been proposed for managing data in mobile and peer-to-peer environment [16]. This technique restricts the availability of write operations since they cannot be executed at the failure of any copy and provides read operations with high degree of availability at low cost. In BRS technique [13], the clients that who request for the data file, the replicas are created as close as possible to them. The root replica grows toward the clients in a branching way, slip replicas into several sub replicas [1]. In this technique, the replica tree will be growing based on the client needs. The expansion of the replication tree might not be symmetric and different branches could have different depths. In HRS technique, a hierarchical replication consists of a root database server and one or more database servers organized into a hierarchy topology [17]. Using this

technique, the data will be replicated or copy at all sites and has the highest storage of use.

III. System Model

Binary Vote Assignment on Cloud Quorum (BVACQ) technique will be used to approach the research. Each site has a premier data file. In the remainder of this paper, we assume that replica copies are data files. A data is logically replicated to the neighbouring sites from its primary site. In this section, we proposed the new BVACQ algorithm by considering the distributed database fragmentation. The following notations are defined:

a) V is a transaction.
b) S is relation in database.
c) S_i is vertical fragmented relation derived from S, where i = 1,2,...,n.
d) P_k is a primary key
e) x is an instant in T which will be modified by element of V.
f) T is a tuple in fragmented S.
g) $^{Si}_{Pkxx}$ is a horizontal fragmentation relation derived from S_i.
h) P_i is an attribute in S where i = 1,2,...,n.
i) $M_{i,j}$ is an instant in relation S where i and j = 1,2,...,n.
j) i represent a row in S.
k) j represent a column in S.
l) η and ψ are groups for the transaction V.
m) $\gamma = \alpha$ or β where it represents different group for the transaction V (before and until get quorum).
n) V_η is a set of transactions that comes before V_ψ, while V_ψ is a set of transactions that comes after V_η.
o) D is the union of all data objects managed by all transactions V of BVACQ.
p) Target set = {-1, 0, 1} is the result of transaction V; where -1 represents unknown status, 0 represents no failure and 1 represents accessing failure.
q) BVACQ transaction elements V_η = $\{V_{\eta x,qr} | r=1,2,...,k\}$ where $V_{\eta x,qr}$ is a queued element of V_η transaction.
r) BVACQ transaction elements V_ψ = $\{V_{\psi x,qr} | r=1,2,...,k\}$ where $V_{\psi x,qr}$ is a queued element of V_ψ transaction.
s) BVACQ transaction elements V_λ = $\{ V_{\lambda x,qr} | r=1,2,...,k\}$ where $V_{\lambda x,qr}$ is a queued element either in different set of transactions V_η or V_ψ.
t) $\hat{V}_{\lambda x,q1}$ is a transaction that is transformed from $V_{\lambda x,qr}$.
u) $V_{\mu x,q_1}$ represents the transaction feedback from a neighbour site. $V_{\mu x,q_1}$ exists if either $V_{\lambda x,qr}$ or $\hat{V}_{\lambda x,q1}$ exists.
v) Successful transaction at primary site $V_{\lambda x,qr} = 0$ where $V_{\lambda x,qr} \in D$ (i.e., the transaction locked an instant x at primary). Meanwhile, successful transaction at neighbour site $V(\mu_{x,q_1}) = 0$, where $\mu_{x,q_1} \in D$ (i.e., the transaction locked a data x at neighbour).

Experimental Result

To make it clearer on how we manage the transaction using BVACQ, here we present the example case. Each node is connected to one another through a cluster with 3 replication servers connected to each as shown in Figure 1.

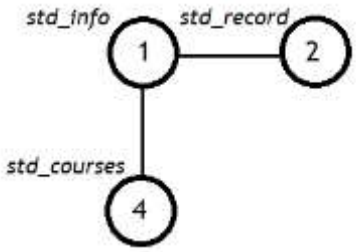

Figure 1. Three replication servers connected to each.

Using BVACQ rules, each primary replica will copy other database to its neighbour replicas. Client can access other database at any server that has its replica. We assume that primary database *std_info* located in Server 1, primary database *std_record* will be at Server 2 and primary database *std_courses* will be at Server 3. Based on BVACQ model, *status* for V_{λ_{status},q_1} will be any instant *a*, *b*, *c*, *d*, *e*, *f*, *g*, *h* and *i*. If two sets of transactions, V_η and V_ψ initiates to update database *std_info* at replica 1 and 2, first it needs to request to update database *std_info* from primary replica 1 and 2. If $V_{\lambda_{status},q_1} = 0$, the other transactions in queue will abort. Then, neighbour binary voting assignment is initiated. Transactions in both nodes, V_η and V_ψ will propagate lock. The first transaction that initiated will get the lock and other transactions will be aborted. So now Replica 1 and 2 have a transaction waiting but transactions cannot read or update database *a* at same time.

Primary nodes 1 propagate lock to its neighbor replicas 2 while primary nodes 2 propagate lock to its neighbour replica 4. Primary replica for V_{η_{status},q_1} propagates lock to its neighbor replicas 2 and 4. Primary replica for

V_{ψ_{status}, q_1} propagates lock to its neighbour replicas 1 and 4. The first transaction get majority quorum will be transform to $\hat{V}_{\lambda_{status}, q_1}$. The details of experiment result are shown in Table 1.

TABLE 1: EXPERIMENTAL RESULT

REPLICA / TIME TAKEN (ms)	1	2	4
t1	unlock(*x*)	unlock(*x*)	unlock(*x*)
t2	begin_transaction	begin_transaction	begin_transaction
t3	$V_{\eta_{x,q_1}}$ write lock(*x*), counter_w(*x*)=1		
t4	$V_{\eta_{x,q_1}}$ pro-pagate lock:B		
t5		$V_{\eta_{x,q_1}}$ lock(*x*) from E	
t6	$V_{\eta_{x,q_1}}$ get lock:B, counter_w(*x*)=2		
t7	$V_{\eta_{x,q_1}}$ propagate lock:D		
t8			D fail to response to $V_{\eta_{x,q_1}}$
t9	$V_{\eta_{x,q_1}}$ get unknown status from: D, counter_w(*x*)=2		
t10	$V_{\eta_{x,q_1}}$ obtain quorum		
t11	$V_{\eta_{x,q_1}}$ update *x*		
t12	commit $\hat{V}_{\lambda_{x,q_1}} \in V_\eta$	commit $\hat{V}_{\lambda_{x,q_1}} \in V_\eta$	unknown status
t13	unlock(*x*)	unlock(*x*)	unlock(*x*)

From the result from Table 1, at time equal to 1 (*t1*), instant *x* at all servers are unlocked. At *t2*, the transaction begins. At *t3*, there is a transaction, $V_{\eta_{x,q_1}}$ request to update instant *x* at server *E*. The transaction initiates lock. Hence, write counter for server *E* now is equal to 1. At *t4*, $V_{\eta_{x,q_1}}$ propagate lock at its neighbour replica *B* At server B, $V_{\eta_{x,q_1}}$ lock(*x*) from *E*. Thus at *t6*, the transaction achieved in getting locked from the *B* then write quorum is equal to 2. Next, $V_{\eta_{x,q_1}}$ propagate lock at server *D* at *t7* and at *t8*, *D* fail to response to $V_{\eta_{x,q_1}}$. Thus at *t9*, $V_{\eta_{x,q_1}}$ get unknown status from *D*, then write quorum is equal to 2. At *t10*, $V_{\eta_{x,q_1}}$ obtain quorums and then instant *x* is updated at *t11*. Finally, at *t12*, $\hat{V}_{\lambda_{x,q_1}} \in V_\eta$ is commit at *E* and *B* and at *t13*, instant *x* at all replica servers will unlock and ready for next transaction. Since *E* receives unknown status, this replication process will not stop here. Server *E* will continuously propagate server *D* until it success. Then it will replicate $V_{\eta_{x,q_1}}$ to server *D*. Hence, all replicated server will have the same data. When server *D* recovers, all the missing transaction update during failure will be reconcile from the first missing transaction until the latest sequences of transaction.

IV. CONCLUSIONS

In order to preserve data consistency and reliability of the systems, managing transactions is very important. With the aim of managing replication data availability even in the failure cases in Cloud, we design a new model called Binary Vote Assignment on Cloud Quorum (BVACQ). From the experiment result, we can say that managing replication and transaction through proposed BVAGQ able to preserve data consistency. It also increases the degrees of data consistency by replicate the missing data into the server with unknown status (server that fail to response to transaction's request) after the server recover from error.

ACKNOWLEDGMENT

Appreciation conveyed to Ministry of Higher Education Malaysia for project financing under Exploratory Research Grant Scheme RDU120608.

REFERENCES

[1] Noraziah Ahmad, Noriyani Mat Zin, Roslina Mohd. Sidek, Mohammad Fadel Jamil Klaib, and Mohd. Helmy Abd Wahab, "Neighbour Replica Transaction Failure Framework in Data Grid", F. Zavoral et al. (Eds.): NDT 2010, Part II, CCIS 88, pp. 488–495, 2010. © Springer-Verlag Berlin Heidelberg 2011.

[2] J. Gray, P. Helland, P. O'Neil, D. Shasha," The Danger of Replication And A Solution", Proceedings of ACM SIGMOD Int. Conf. on Management of Data, Montreal, 1996.

[3] Aiqiang Gao, Luhong Diao "Lazy Update Propagation for Data Replication in Cloud Computing", Pervasive Computing and Applications (ICPCA), pp. 250 – 254, 2010.

[4] Esther Pacitti , Pascale Minet, Eric Simon "Replica Consistency in Lazy Master Replicated Databases Distributed and Parallel Databases", 9, 237267, 2001.

[5] A.Noraziah, M.Mat Deris, R.Norhayati, M.Y.M.Saman, "Distributed Transaction Semantic for Binary Vote Assignment Grid", The Fifth International Conference On Electrical Engineering/Electronics, Computer, Telecommunications And Information Technology, IEEE Xplore, Vol. 1, pp. 41-44, Mei 2008.

[6] M.Mat Deris, D.J.Evans, M.Y.Saman, A.Noraziah, "Binary Vote Assignment on Grid For Efficient Access of Replicated Data", International Journal of Computer Mathematics, Taylor and Francis, UK, Vol.80, No.12, pp. 1489 – 1498, Dec. 2003.

[7] I. Foster; Y. Zhao, I.Raicu, S. Lu, "Cloud Computing and Grid Computing 360-Degree Compared", Grid Computing Environments Workshop, pp. 1-10, 2008.

[8] S. Siva Sathya, K. Syam Babu, "Survey of fault tolerant techniques for grid", Computer Science Review, Volume 4, Issue 2, May 2010, Pages 101-120.

[9] Zeghache Linda, Badache Nadjib, "Optimistic Replication Approach for Transactional Mobile Agent Fault Tolerance", 2010 11th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, IEEE, 2010, DOI 10.1109/SNPD.2010.37.

[10] Tamhane, S.A.; Kumar, M., "Middleware for decentralised fault tolerant service execution using replication in pervasive systems", Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010, 8th IEEE International Conference, pp.474-479, March 29 2010-April 2 2010 doi: 10.1109/PERCOMW.2010.5470622.

[11] Mohammad Bsoul, Ahmad Al-Khasawneh, Emad Eddien Abdallah, Yousef Kilani, "Enhanced FastSpreadReplicationstrategyforDataGrid", Journal of Network and Computer Applications 34 (2011) 575–580, DOI:10.1016/j.jnca.2010.12.006.

[12] Noraziah Ahmad, Ainul Azila Che Fauzi, Roslina Mohd. Sidek, Noriyani Mat Zin, and Abul Hashem Beg, "Lowest Data Replication Storage of Binary Vote Assignment Data Grid", F. Zavoral et al. (Eds.): NDT 2010, Part II, CCIS 88, pp. 466–473, 2010. © Springer-Verlag Berlin Heidelberg 2010.

[13] José M. Pérez, Félix García-Carballeira, Jesús Carretero, Alejandro Calderón, Javier Fernández, "Branch replication scheme: A new model for data replication in large scale data grids", Future Generation Computer Systems, Vol. 26, No. 1, pg 12-20, 2010.

[14] Xin Sun, Jun Zheng, Qiongxin Liu, Yushu Liu, "Dynamic Data Replication Based on Access Cost in Distributed Systems", 2009 Fourth International Conference on Computer Sciences and Convergence Information Technology, IEEE, 2009, DOI:10.1109/ICCIT.2009.198.

[15] K. Sashi, Antony Selvadoss Thanamani, "Dynamic replication in a data grid using a Modified BHR Region Based Algorithm", Future Generation Computer Systems 27 (2011) 202–210, 2010 DOI:10.1016/j.future.2010.08.011.

[16] Budiarto, S.Noshio, M.Tsukamoto,"Data Management Issue in Mobile and Peer to Peer Environment", Data Knowledge Engineering, Elsevier, 41, pp.183-204, 2002.

[17] Pérez, J.M., Carballeira, F.G., Carretero, J., Calderón, A., Fernandez, J.: Branch Replication Scheme, "A New model for Data Replication in Large Scale Data Grids", Computer Architecture Group, Computer Science Department, Universidad Carlos III de Madrid, Leganes, Madrid, Spain (2009).

[18] Ainul Azila Che Fauzi, Noraziah Ahmad, Abul Hashem Beg, "A New Binary Vote Assignment Algorithm to Manage Replication in Distributed Database Environment", The 2011 International Conference on Computer Communication and Management, 2011.

[19] A.Noraziah, Ainul Azila Che Fauzi, Mustafa Mat Deris, Md Yazid Mohd Saman, Noriyani Mohd Zain, Nawsher Khan, "Managing Educational Resource - Student Information Systems Using BVAGQ Fragmented Database Replication Model", Procedia Social and Behavioral Sciences, Elsevier, 2011.

[20] Noraziah Ahmad, Nawsher Khan, Ahmed N. Abdalla, Abul Hashem Beg, "Novel Database Design for Student Information System", Journal of Computer Science, Vol. 6, No. 1, pp. 43-46, 2010.

[21] M.Mat Deris, M.Rabiei, A.Noraziah, H.M. Suzuri, "High Service Reliability for Cluster Server Systems", International Conference on Cluster Computing, Vol. 1, pp 280-287, 2003.

[22] Nawsher Khan, A.Noraziah, Mustafa Mat Deris, Elrasheed I.Ismail "Cloud Computing: Comparison of Various Features", Ezendu et al. (Eds): Communications in Computer and Information Science (CCIS) vol. 194, Springer-Verlag Berlin Heidelberg, pp. 243–254, 2011.

[23] Nawsher Khan, A. Noraziah, Tutut Herawan, Zakira Inayat, "Cloud Computing: Locally Sub-Clouds Instead of Globally one Cloud", International Journal of Cloud Application and Computing (IJCAC), IGI Global, Vol 2, No. 3, pg 68-84, 2012.