Journal of Computer Science 7 (9): 1338-1342, 2011
ISSN 1549-3636

Handling Fragmented Database Replication through Binary Vote Assignment Grid Quorum

Ainul Azila Che Fauzi, A. Noraziah, Noriyani Mohd Zain, A.H. Beg,
Nawsher Khan and Elrasheed Ismail Sultan
Faculty of Computer Systems and Software Engineering,
University Malaysia Pahang, 26300,
Kuantan, Pahang, Malaysia

Abstract: Problem statement: Organizations critically needed to supply recent data to users who may be geographically remote, while at the same time handle a volume request of distributed data around multiple sites. The storage, availability and consistency are important issues to be addressed in order to allow distributed users efficiently and safely access data from many different sites. **Approach:** Data replication is a way to deal with this problem since it provides user with fast, local access to shared data and protects availability of applications because alternate data access options exist. Handling fragmented database replication becomes challenging issue to administrator since the distributed database was scattered into split replica partitions or fragments. **Results:** This study presented a new mechanism on how to handle the fragmented database replication through the Binary Vote Assignment on Grid Quorum (BVAGQ). We address how to build reliable system by using the proposed BVAGQ for distributed database fragmentation. **Conclusion:** The result shows that managing fragmented database replication and transaction through proposed BVAGQ is able to preserve the data consistency.

Key words: Data replication, data grid, Hierarchical Replication Scheme (HRS), Distributed Database Systems (DDS), Binary Vote Assignment on Grid Quorum (BVAGQ), vertical fragmentation, horizontal fragmentation

INTRODUCTION

Nowadays, organizations critically need to supply recent data to users who may be geographically remote and to handle a volume of requests of data distributed around multiple sites. One way to provide access to such data is through replication. It is broadly installed in disaster tolerance systems to replicate data from the primary system to the remote backup system dynamically and online (Ren *et al*., 2003). Replication provides user with fast, local access to shared data and protects availability of applications because alternate data access options exist (Dastgheib, 2010). Distributed database replication involves the process of copying and maintaining database objects in multiple databases that make up a Distributed Database Systems (DDS) (Ren *et al*., 2003). Handling fragmented database replication becomes challenging issue to administrator since the distributed database is scattered into split replica partitions or fragments. Each partition or fragment of a distributed database may be replicated into several different sites in distributed environment. Changes applied at one site are captured and stored locally before being forwarded and applied at each of the remote locations. Fragmentation in distributed database is very useful in terms of usage, efficiency, parallelism and also for security. This strategy will partition the database into disjoint fragments. If data items are located at the site where they used most frequently, locality of reference is high. In fragmentations, similarly, reliability and availability are low Distributed Database, 2011. But by combining fragmentation with replication, performance should be good Distributed Database, 2011. Even if one site becomes unavailable, users can continue to query or even update the remaining fragments.

Data replication can be divided into three categories of fragmented replication scheme which are all-data-to-all-sites, some-data-to-all-sites and some-data-to-some-sites. The examples of all-data-to-all-sites protocols are Read-One-Write-All (ROWA) (Ahmad *et al*., 2010a; Deris *et al*., 2009) and Hierarchical Replication Scheme (HRS) (Perez *et al*., 2010). ROWA has been proposed preserving replicated data file in network environment (Ahmed *et al*., 2010a;

Corresponding Author: Ainul Azila Che Fauzi, Faculty of Computer Systems and Software Engineering, University Malaysia Pahang, 26300, Kuantan, Pahang, Malaysia

2010b). Meanwhile, replication in HRS starts when a transaction initiates at site 1. All the data will be replicate into other site. All sites will have all the same data. For some-data-to-all-sites category, The Majority Quorum protocol and Weighted Voting protocol employ voting to decide the quorums techniques (Choi and Youn, 2010). A tree structure has been assigned to the set of replicas in this technique. The replicas are positioned only in the leaves, whereas the non-leaf nodes of the tree are regarded as "logical replicas", which in a way summarize the state of their descendants (Storm and Theel, 2009). Besides Voting Protocol, Tree Quorum (TQ) (Choi and Youn, 2010) can also be categorized in some-data-to-all-sites. These replication protocols make use of a logical tree structure. The cost and availability vary according to the failure condition, whereas they are constant for other replication protocols (11). One more protocol in this category is Branch replication scheme (Perez *et al*., 2010). Its goals are to increase the scalability, performance and fault tolerance. Replicas are created as close as possible to the clients that request the data files. Using this technique, the growing of the replica tree is driven by client needs. Binary Vote Assignment on Data Grid (BVADG) (Ahmad *et al*., 2010b) is one of the protocols in some-data-to-some-sites protocol. A data will replicate to the neighboring sites from its primary site. Four sites on the corners of the grid have only two adjacent sites and other sites on the boundaries have only three neighbors. Thus, the number of neighbors of each sites is less than or equal to four.

Research background: Data replication: Replication is the process of sharing information to ensure consistency between redundant resources such as software or hardware components. This process helps to improve reliability, fault-tolerance, or accessibility of data (Gudiu *et al*., 2010; Connolly and Begg, 1998). Data replication may occur if the same data is stored in multiple storage devices. Meanwhile, computation replication occurs when the same computing task is executed many times. A computational task is typically replicated in space, i.e., executed on separate devices, or it could be replicated in time, if it is executed repeatedly on a single device. Whether one replicates data or computation, the objective is to have some group of processes that handle incoming events. If we replicate data, these processes are passive and operate only to maintain the stored data, reply to read requests and apply updates. When we replicate computation, the usual goal is to provide fault-tolerance. For example, a replicated service might be used to control a telephone switch, with the objective of ensuring that even if the primary controller fails, the backup can take over its functions (Storm and Theel, 2009).

Distributed database fragmentation: Fragmentation in distributed database is very useful in terms of usage because usually, applications study with only some of relations rather than entire of it (Connolly and Begg, 1998). In data distribution, it is better to study with subsets of relations as the unit of distribution. The other benefit from fragmentation is the efficiency. Data is stored close to where it is most frequently used and for data that is not needed, it is not stored. By using fragmentation, a transaction can be divided into several subqueries that operate on fragments. So, it will increase the degrees of parallelism. Besides, it also good for security as data not required for local applications is not stored. So, it will not available to unauthorized users. There are two main types of fragmentation which are horizontal and vertical. Horizontal fragments are subsets of tuples, whereas vertical fragments are subsets of attributes. Figure 1a and b show the horizontal and vertical fragmentations.

Horizontal fragmentation: Horizontal fragmentation groups together the tuples in a relation that are used by the important transactions (Atlas at the University of Chicago, 2011). A horizontal fragment is produced by specifying a predicate that performs a restriction on the tuples in the relation. It is defined using the Selection operation of the relational algebra. Given a relation R, a horizontal fragment is defined as:

$\sigma p (R)$

where, p is a predicate based on one or more attributes of the relation.

Vertical fragmentation: Vertical fragmentation groups together the attributes in a relation that are used jointly by the important transactions (Atlas at the University of Chicago, 2011). A vertical fragment is defined using the Projection operation of the relational algebra. Given a relation R, a vertical fragmentation is defined as:

$\Pi a1,\ldots,an (R)$

where, a1,…,an are attributes of the relation R.

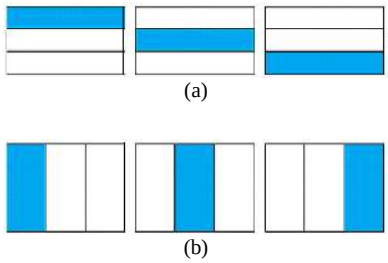

Fig. 1a and b: Horizontal and vertical fragmentations

MATERIALS AND METHODS

Binary Vote Assignment Grid Quorum (BVAGQ) technique will be used to approach the research. In BVAGQ, all sites are logically organized in form of two-dimensional grid structure. Each site has a premier data file. A site is either operational or failed and the state (operational or failed) of each site is statistically independent to the others. A data will replicate to the neighboring sites from its primary site. Consider a case of 9 sites logically organized in 3×3 two-dimensional grid structures. Four sites on the corners of the grid have only two adjacent sites and other sites on the boundaries have only three neighbors. Thus, the number of neighbors of each sites is less than or equal to 4. In Fig. 2, data from site 1 will replicate to site 2 and 4 which are its neighbors. Site 5 has four neighbors, which are sites 2, 4, 6 and 8. So, site 5 has five replicas. Meanwhile, site 6 replicates to site 3, 5 and 9.

Definition:

- V is a transaction
- S is relation in database
- S_i is vertical fragmented relation derived from S, where i = 1,2,...,n
- PK is a primary key
- x is an instant in T which will be modified by element of V
- T is a tuple in fragmented S
- S^i_{PKxx} is a horizontal fragmentation relation derived from S_i
- P_i is an attribute in S where i = 1,2,...,n
- $M_{i,j}$ is an instant in relation S where i and j = 1,2,...,n
- i represent a row in S
- j represent a column in S
- η and ψ are groups for the transaction V
- γ=a or b where it represents different group for the transaction V (before and until get quorum)
- Vη is a set of transactions that comes before Vψ
- While Vψ is a set of transactions that comes after Vη
- D is the union of all data objects managed by all transactions V of BVAG
- 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
- BVAG transaction elements $V\eta = \{V_{\eta x,qr} | r=1,2,...,k\}$ where $V_{\eta x,qr}$ is a queued element of Vη transaction
- BVAG transaction elements $V\psi = \{V_{\psi x,qr} | r=1,2,...,k\}$ where $V_{\psi x,qr}$ is a queued element of Vψ transaction

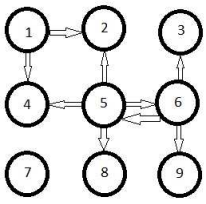

Fig. 2: Primary and neighbors replica coordination

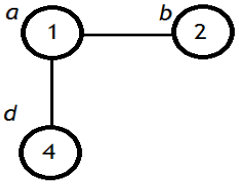

Fig. 3: Three replication servers connected to each

- BVAG transaction elements $V\lambda = \{ V_{\lambda x,qr} | r=1,2,...,k\}$ where $V_{\lambda x,qr}$ is a queued element either in different set of transactions Vηor Vψ
- $V_{\lambda x,q1}$ is a transaction that is transformed from $V_{\lambda x,qr}$. $V_{ux,q1}$ represents the transaction feedback from A neighbor site. $V_{ux,q1}$ exists if either $V_{\lambda x,qr}$ or $V_{\lambda x,q1}$ exists
- Successful transaction at primary site $V_{\lambda x,qr} = 0$
- Where $V_{\lambda x,qr} \epsilon D$ (i.e., the transaction locked an instant x at primary). Meanwhile, successful transaction at neighbor site $V(u_{x,q1}) = 0$, where $_{ux,q1\varepsilon D}$ (i.e.,, the transaction locked a data x at neighbor)

RESULTS

To make it clearer on how we manage To make it clearer on how we manage the transaction using BVAGQ, here we present the example case. Each node is connected to one another through an Ethernet switch hub. A cluster with 3 replication servers connected to each as shown in Fig. 3.

Using BVAG rules, each primary replica will copy database x to its neighbor replicas. Client can access database x at any server that has its replica. We assume that primary database a located in Server 1, primary database b will be at Server 2 and so on. Based on BVAGQ model, a for $V_{\lambda a,q1}$ will be any instant a, b, c, d, e, f, g, h and i.

DISCUSSION

For the first experiment, consider $V\lambda_{a,\ q1}$,λ=η request to update data a at server 1. The first request

that get lock which is $V\lambda_{a,\ q1}$ will proceed with the transaction and $V\lambda_a$, qr+1,… $V\lambda_a$, qk aborted as shown in Table 1. $V\lambda_{a,\ q1}$ is the write counter for $V\lambda_{a,\ q1}$ that increases when it gets a lock. Next, the $V\lambda_{a,\ q1}$ fragmented into S_2 and S_2 is fragmented into S^2_{PKxx} Based on the primary key of the fragmented tuple; instant a will be updated. After finish update, the transaction will commit.

For second transaction, if two sets of transactions, $V\lambda_{a,\ q1,}$ $\lambda=\eta,\psi$ and $V\lambda_{a,\ q1,}\lambda=\psi$ initiates to update database a at replica 1, transaction $V\lambda_{a,\ q1,}$ $\lambda=\psi$ will abort. Transaction $V\lambda_{a,\ q1,}$ $\lambda=\psi$ is aborted because we already fix the system will choose the first transaction that make request based on timestamps. After identify which transaction will be executed, we will fragmented the database using horizontal and vertical fragmentation to get the instant that we want to update. From Table 2, we can see that $V_{\lambda a,q1}$ precede the transaction execution. $V_{\lambda a,q1}$ fragmented into S_2 and again S_2 is fragmented into S^2_{PKxx} Instant a will be update. After that, all replica will commit and unlock.

Table1: Experiment result

Time	Replica 1	Replica 2	Replica 3
0.1	unlock (a)	unlock (a)	unlock (a)
0.2	Begin transaction	Begin Transaction	Begin transaction
0.3	$V_{\eta a,\ q1}$ get lock		
0.4	$V_{\eta a,qr+1....,}$ $V_{\eta a,\ qk}$ aborted		
0.5	$V_{\eta a,\ q1}$ fragmented into S_2		
0.6	S_2 is fragmented into S^2_{PKxx}		
0.7	S^2_{PKxx} divided into T_1 and T_2 Where, $T_1 = P_1$ = Primary key $T_2 = P_6$ = instant a		
0.8	update a		
0.9	Commit λ_a,q_1	Commit λ_a,q_1	Commit λ_a,q_1
0.10	unlock (a)	unlock (a)	unlock (a)

Table2: Experiment result

Time	Replica 1	Replica 2	Replica 3
0.1	unlock (a)	unlock (a)	unlock (a)
0.2	Begin transaction		
0.3	write lock (a) counter (a)=1		
0.4	wait		
0.5	$V_{\eta a,\ q1}$ propagate lock: 2 $V_{\psi a,q1}$ aborted		
0.6	$V_{\eta a,qr+1....,}$ $V_{\eta a,\ qk}$ aborted		
0.7	$V_{\eta a,\ q1}$ fragmented into S_2		
0.8	S_2 is fragmented into S^2_{PKxx}		
0.9	S^2_{PKxx} divided into T_1 and T_2 Where, $T_1 = P_1$ = Primary key $T_2 = P_6$ = instant a		
0.10	update a		
0.11	Commit λ_a,q_1		
0.12	unlock (a)	unlock (a)	unlock (a)

CONCLUSION

Handling fragmented database replication is very important in order to preserve the data availability, consistency and reliability of the systems. Therefore, a new Binary Vote Assignment on Grid Quorum technique has been proposed to maintain and manage the fragmented database replication. From the experiment result, it shows that the system preserves the data consistency through the synchronization approach for all replicated sites. Furthermore, it guarantees the consistency since the transaction execution is obeyed the one-copy-serializability.

ACKNOWLEDGEMENT

Appreciation conveyed to Ministry of Higher Education Malaysia for supporting this project under Fundamental Research Grant Scheme, RDU100109.

REFERENCES

Ahmad, N. A.A.C. Fauzi, N.M. Zin and A.H. Beg, 2010a. Lowest data replication storage of binary vote assignment data grid. Proceeding of the 2nd International Conference on 'Networked Digital Technologies' (NDT 2010). Springer-Verlag Berlin Heidelberg, pp: 466-473.

Ahmad, N., R.M. Sidek, M.F.J. Klaib and T.L. Jayan, 2010b. A novel algorithm of managing replication and transaction through Read-One-Write-All Monitoring Synchronization Transaction System (ROWA-MSTS). Proceedings of the 2nd International Conference on Network Applications Protocols and Services, Sept. 22-23, IEEE Xplore Press, Kedah, pp: 20-25. DOI: 10.1109/NETAPPS.2010.11

Ahmed, A., A.N. Abdalla and R.M. Sidek, 2010a. Data replication using read-one-write-all monitoring synchronization transaction system in distributed environment. J. Comput. Sci., 6: 1095-1098. DOI: 10.3844/jcssp.2010.1095.1098

Ahmed, N., R.M. Sidek and M.F.J. Klaib, 2010b. Development of ROWA-MSTS in distributed systems environment. Proceedings of the 2nd International Conference on Computer Research and Development, May 7-10, IEEE Xplore Press, Kuala Lumpur, pp: 868-871. DOI: 10.1109/ICCRD.2010.70

Choi, S.C. and H.Y. Youn, 2010. Dynamic hybrid replication effectively combining tree and grid topology. J. Supercomput., 1-23. DOI: 10.1007/s11227-010-0536-6

Connolly, T. and C. Begg, 1998. Database System A Practical Approach to Design, Implementation and Management (International Computer Science Series). 2nd Edn, Addison-Wesley Publishing Company, ISBN: 10: 0201342871, pp: 848.

Dastgheib, A.M., 2010. Introducing a suggestive dynamic data replication algorithm. Proceedings of the International Conference Electronic Computer Technology, May 7-10, IEEE Xplore Press, Kuala Lumpur, pp: 97-101. DOI: 10.1109/ICECTECH.2010.5479980

Deris, M.M., J.H. Abawajy, D. Taniar and A. Mamat, 2009. Managing data using neighbour replication on a triangular-grid structure. Int. J. High Perform. Comput. Network., 6: 56-65. DOI: 10.1504/IJHPCN.2009.026292

Gudiu, A., E. Voişan and F. Dragan, 2010. Database replication driven communication model for distributed dedicated web hosting systems. Proceeding of the International Joint Conference on Computational Cybernetics and Technical Informatics, May 27-29, IEEE Xplore Press, Timisoara, pp: 311-314. DOI: 10.1109/ICCCYB.2010.5491260

Perez, J.M., F. Garcia-Carballeira, J. Carretero, A. Calderon and J. Fernandez, 2010. Branch replication scheme: A new model for data replication in large scale data grids. Future Generation Comput. Syst., 26: 12-20. DOI: 10.1016/J.FUTURE.2009.05.015

Ren, K., Z. Li and C. Wang, 2010. LBDRP: A low-bandwidth data replication protocol on journal-based application. Proceedings of the 2nd International Conference on Computer Engineering and Technology, Apr. 16-18, IEEE Xplore Press, Chengdu, pp: 89-92. DOI: 10.1109/ICCET.2010.5485707

Storm, C. and O. Theel, 2009. A general approach to analyzing quorum-based heterogeneous dynamic data replication schemes. Proceedings of the 10th International Conference on Distributed Computing and Networking (ICDCN'09), Springer-Verlag Berlin, Heidelberg, pp: 349-361. DOI: 10.1007/978-3-540-92295-7_42