High Reliability Replication Technique for Web-Server Cluster Systems

M. Mat Deris[1], J.H. Abawajy[2], M. Zarina[3], and R. Mamat[4]

[1] Faculty of Information Technology and Multimedia,
College University Tun Hussein Onn
Batu Pahat, Johor, Malaysia
mustafa@kustem.edu.my
[2] Deakin University, School of Information Technology,
Geelong, VIC, Australia
jemal@deakin.edu.au
[3] KUSZA College,Kuala Terengganu, Malaysia
zarina@kusza.edu.my
[4] Department of Computer Science,
Faculty of Science and Technology, KUSTEM, K.Terengganu, Malaysia
rab@kustem.edu.my

Abstract. Providing reliable and efficient services are primary goals in designing a web server system. Data replication can be used to improve the reliability of the system. However, mapping mechanism is one of the primary concerns to data replication. In this paper, we propose a mapping mechanism model called enhanced domain name server (E-DNS) that dispatches the user requests through the URL-name to IP-address under Neighbor Replica Distribution Technique (NRDT) to improve the reliability of the system.

1 Introduction

With the ever increasing applications in the world wide web (WWW) such as distance learning education and e-commerce, the need for a reliable web server is likely to increase [6]. Thus, providing reliable and efficient services are primary goals in designing a web server cluster (WSC) system. This is due to the constraints of the eventual failure of hardware or/and software components. In order to provide reliable services, a WSC needs to maintain the availability of some data replicas while preserving one-copy consistency among all replicas [4]. Therefore, data replication plays an important role in a WSC as a highly reliable system.

The most common approaches in the replication techniques are the synchronous and asynchronous replications. The former provides a 'tight consistency' between data stores. This means that the latency between the data consistency achieved will be zero. Data in all nodes are always the same, no matter from which replica the updated originated. However, synchronous replication has drawbacks in practice [5]. The major argument is that, the response time to execute an operation is high because the time taken for all nodes that have the same copy to 'agree' to execute an operation is high. Whilst the asynchronous replication provides a 'loose consistency' between data

V.S. Sunderam et al. (Eds.): ICCS 2005, LNCS 3516, pp. 447 – 456, 2005.

stores. The replication process occurs asynchronous to originating transaction. In other words, there is always some degree of lag between the time the originating transaction is committed and the effects of the transaction are available at any replica(s) [1]. Nevertheless, the response time is lower than that of the synchronous technique.

Reliability refers to the probability that the system under consideration does not experience any failure in a given time interval. Thus, a reliable WSC is one that can continue to process user's requests even when the underlying system is unreliable [3]. When components fail, it should still be able to continue executing the requests without violating the database consistency.

In this paper we propose the enhanced domain name server (E-DNS) algorithm as a mapping mechanism under the Neighbor Replica Distribution Technique (NRDT) to improve the reliability of the system. In case of server failure, the server will be switched to neighboring IP address.

The paper is organized as follows. Section 2 reviews NRDT model. Section 3 describes typical DNS. Section 4 describes the concept of E-DNS. Section 5 concludes the proposed work.

2 Neighbor-Replica Distribution Technique (NRDT)

2.1 The Model

In a replicated database, copies of a data object may be stored at several sites in the network. Multiple copies of a data object must appear as a single logical data object to the transactions. This is termed as one-copy equivalence and is enforced by the replica control technique. The correctness criteria for replicated database is one-copy serializability [11], which ensures both one-copy equivalence and the serializable execution of transactions. In order to ensure the one-copy serializability, a replicated data object may be read by reading a quorum of copies, and it may be written by writing a quorum of copies. The selection of a quorum is restricted by the quorum intersection property to ensure one-copy equivalence: For any two operations o[x] and o'[x] on a data object x, where at least one of them is a write, the quorum must have a non-empty intersection. The quorum for an operation is defined as a set of copies whose number is sufficient to execute that operation.

Briefly, a site *i* initiates a NRDT transaction to update its data object. For all accessible data objects, a NRG transaction attempts to access a NRDT quorum. If a NRDT transaction gets a write quorum with non-empty intersection, it is accepted for execution and completion, otherwise it is rejected. We assume for the read quorum, if two transactions attempt to read a common data object, read operations do not change the values of the data object. Since read and write quorums must intersect and any two NRDT quorums must also intersect, then all transaction executions are one-copy serializable.

In the design of the WSC, a client on the Internet will notice only one IP address coming from the cluster, not those individual servers in the cluster. The cluster (with only one IP address visible to the public) is composed of a node called Request Distributor Agent (RDA) and a group of servers. The servers are logically connected

to each other in the form of a grid structure, each of which is connected to RDA. Figure 1 shows architecture of the cluster-server system with nine servers. The RDA will forward legitimate Internet requests to the appropriate servers in the cluster. It returns any replies from the servers back to the clients.

Each node has the premier data (eg. *file a* will be located to server A, *file b* will be located to server B, and so on). The RDA will forward any request to the appropriate server with the premier data file. This is done by maintaining a server-service table that contains all the services provided by the cluster together with the corresponding addresses and their neighbors.

One advantage of a server cluster over a single server is its high security. If a single server is used, it is reachable from the Internet and therefore vulnerable for vicious [3]. Only the RDA has the IP address visible from the Internet, and all other stations of the cluster bear only private IP address. Therefore, all cluster-server stations are not reachable directly from the outside. A firewall system may be installed on the RDA to protect the whole cluster. To attack one of the cluster server stations, one has to first land on RDA and launch an attack from there. Network Address Translation is used on RDA to translate the destination IP address of incoming packets to an internal IP address, and that of the outgoing packets to the IP address on Internet where the requests.

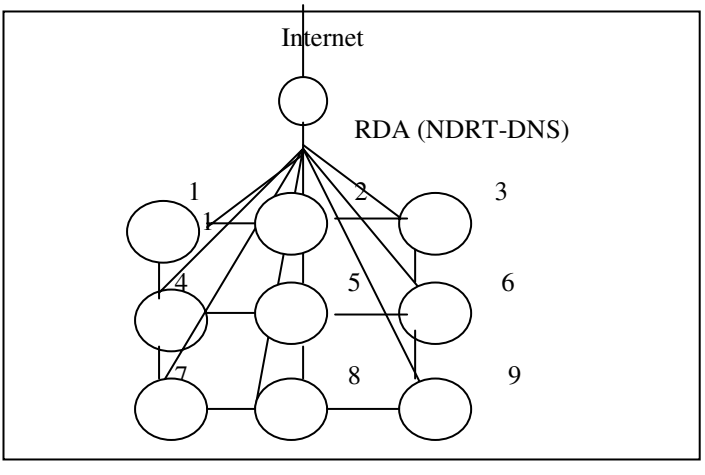

Fig. 1. A Cluster with 9 servers

2.2 The NRDT Technique

In NRDT, all sites are logically organized in the form of a two-dimensional grid structure. For example, if a NRDT consists of nine nodes, it will be logically organized in the form of 3 x 3 grid as shown in Figure. 1. Each node has a *master* data file. In the remainder of this paper, we assume that replica copies are data files. A node is either operational or failed and the state (operational or failed) of each node is

statistically independent to the others. When a node is operational, the copy at the node is available; otherwise it is unavailable.

***Definition 2.2.1*:** A node X is a neighbor to node Y, if X is logically-located adjacent to Y.

A data will replicate to the neighboring nodes from its primary node. The number of data replication, *d*, can be calculated using Property 2.2, as described below.

***Property 2.2*:** The number of data replication from each node, $d \leq 5$.

Proof: Let n be a set of all nodes that are logically organized in a two-dimensional grid structure form. Then n nodes are labeled m(i,j), $1 \leq i < \sqrt{n}$, $1 \leq j < \sqrt{n}$. Two way links will connect nodes n(i,j) with its four neighbors, nodes m(i±1,j) and m(i,j±1), as long as there are nodes in the grid. Note that, four nodes on the corners of the grid, have only two adjacent nodes, and other nodes on the boundaries have only three neighbors. Thus the number of neighbors of each node is less than or equal to 4. Since the data will be replicated to neighbors, then the number of data replication from each node, *d*, is:

$d \leq$ the number of neighbors + a data from node itself = 4+1 =5.

For example, from Figure. 1, data from node 1 will replicate to node 2 and node 4 which are its neighbors. Node 5 has four neighbors, which are nodes 2, 4, 6, and 8. As such, node 5 has five replicas.

For simplicity, the primary node of any data file and its neighbors are assigned with vote one and vote zero otherwise. This vote assignment is called neighbor binary vote assignment on grid. A neighbor binary vote assignment on grid, B, is a function such that

$B(i) \in \{0,1\}$, $1 \leq i \leq n$

where $B(i)$ is the vote assigned to node *i*. This assignment is treated as an allocation of replicated copies and a vote assigned to the node results in a copy allocated at the neighbor. That is,

1 vote $\equiv$ 1 copy.

Let $$L_B = \sum_{i=1}^{n} B(i)$$

where, L_B is the total number of votes assigned to the primary node and its neighbors and it also equals to the number of copies of a file allocated in the system. Thus, $L_B = d$.

Let r and w denote the read quorum and write quorum, respectively. To ensure that the read operation always gets up-to-date values, r + w must be greater than the total number of copies (votes) assigned to all nodes. The following conditions are used to ensure consistency:

1) $1 \leq r \leq L_B$, $1 \leq w \leq L_B$,
2) $r + w = L_B + 1$.

Conditions (1) and (2) ensure that there is a nonempty intersection of copies between every pair of read and write operations. Thus, the conditions ensure that a read operation can access the most recently updated copy of the replicated data. Timestamps can be used to determine which copies are most recently updated.

Let S(B) be the set of nodes at which replicated copies are stored corresponding to the assignment B. Then

$$S(B) = \{i \mid B(i) = 1,\ 1 \le i \le n\}.$$

***Definition 2.2.2*:** For a quorum q, a *quorum group* is any subset of S(B) whose size is greater than or equal to q. The collection of quorum group is defined as the *quorum set*.

Let Q(B,q) be the quorum set with respect to the assignment B and quorum q, then

$$Q(B,q) = \{ G \mid G \subseteq S(B) \text{ and } |G| \ge q\}$$

For example, from Figure. 1, let node 5 be the primary node of the master data file x. Its neighbors are nodes 2, 4, 6 and 8. Consider an assignment B for the data file x, such that

$$B_x(5)=B_x(2)=B_x(4)=B_x(6)=B_x(8) = 1$$

and $L_{B_x} = B_x(5)+ B_x(2)+ B_x(4)+ B_x(6)+ B_x(8) = 5.$

Therefore, $S(B_x) = \{5,2,4,6,8\}$.

If a read quorum for data file x, r =2 and a write quorum $w = L_{Bx}-r+1 = 4$, then the quorum sets for read and write operations are $Q(B_x,2)$ and $Q(B_x,4)$, respectively, where

$Q(B_x,2)=\{\{5,2\},\{5,4\},\{5,6\},\{5,8\},\{5,2,4\},\{5,2,6\},\{5,2,8\},\{5,4,6\},\{5,4,8\},\{5,6,8\},$
$\{2,4,6\},\{2,4,8\},\{2,6,8\},\{4,6,8\},\{5,2,4,6\}, \{5,2,4,8\},\{5,2,6,8\},\{5,4,6,8\},$
$\{2,4,6,8\}, \{5,2,4,6,8\}\}$

and

$Q(B_x,4) = \{\{5,2,4,6\},\{5,2,4,8\},\{5,2,6,8\},\{5,4,6,8\}, \{2,4,6,8\}, \{5,2,4,6,8\}\}$

2.3 The Correctness of NRDT

In this section, we will show that the NRDT technique is one-copy serializable. We start by defining sets of groups called *coterie* [12] and to avoid confusion we refer the sets of copies as *groups*. Thus, sets of *groups* are sets of sets of copies.

***Definition* 2.3.1.** *Coterie*. Let **U** be a set of *groups* that compose the system. A set of *groups* **T** is a coterie under **U** iff

i) $G \in \mathbf{T}$ implies that $G \neq \emptyset$ and $G \subseteq \mathbf{U}$.
ii) If $G, H \in \mathbf{T}$ then $G \cap H \neq \emptyset$ (intersection property)
iii) There are no $G, H \in \mathbf{T}$ such that $G \subset H$ (minimality).

By definition of coterie and from Definition 2.2.2, then Q(B,w) is a coterie, because it satisfies all coterie's properties.

Since read operations do not change the value of the accessed data object, a read quorum does not need to satisfy the intersection property. To ensure that a read operation can access the most recently updated copy of the replicated data, two conditions in sub-section 2.2 must be conformed. While a write quorum needs to satisfy read-write and write-write intersection properties. The correct criterion for

replicated database is one-copy serializable. The next theorem provides us with a mechanism to check whether NRDT is correct.

Theorem 2.3.1. The NRDT technique is one-copy serializable.

Proof: The theorem holds on condition that the NRDT technique satisfies the quorum intersection properties, i.e., write-write and read-write intersections. Since Q(B,w) is a coterie then it satisfies the write-write intersection. However, for the case of a read-write intersection, it can be easily shown that $\forall G \in Q(B,r)$ and $\forall H \in Q(B,w)$, then $G \cap H \neq \varnothing$.

3 Overview of DNS

At its most basic level, the DNS provides distributed database of name-to-address mappings spread across a hierarchy of nameservers. The namespace is partitioned into a hierarchy of domains and subdomains with each domain administered independently by an authoritative nameserver. Nameservers store the mapping of names to address in resource records, each having an associated time to live (TTL) field that determines how long the entry can be cached by other nameserver in the system. A large TTL value reduces the load on the nameservers but limits the frequency of update propagation through the system. The different types of resource records and additional details about the DNS are described in [9]. The most widely used nameserver implementation in the DNS is the Berkeley Internet Name Domain (BIND) [2].

Nameserver can be implemented in the form of iterative or recursive queries. In an iterative query, the nameserver returns either an answer to the query from its local database, or a referral to another nameserver that may be able to answer the query. In handling a recursive query, the nameserver return a final answer, querying any other nameserver necessary to resolve the name. Most nameserver within the hierarchy are configured to send and accept only iterative queries. Local nameserver, however, typically accept recursive queries from clients.

4 E-DNS

E-DNS is proposed to enhance DNS according to the NRDT requirements. The function is similar to RDA in NRDT model. In case of server(s) fails, user cannot access data due to no other server take over the service. All clients or user requests from Internet are cross through to DNS to decide the right server to replay the request. When a server fails, the client who maps the name to IP address will find out that the server is down. The problem still unsolved although the client may press ‘reload’ or ‘refresh’ button in their browsers. This problem is unacceptable and unreliable. E-DNS is a viable alternative or solution to this problem because it is developed to take over the service in case of server(s) failure. To do this we used the complete DNS BIND software and only nameserver (it is the text file) in DNS will be updated according to match NRDT requirement. NRDT program will check the server fail/up every *t* second (time can be configured). E-DNS centralizes dispatching host name into single IP address and update it according to NRDT pattern. In case of server

failures, the server will be switched to its neighboring IP address. One of the main attractions of this approach is its ease of deployment.

4.1 E-DNS Algorithm

We design an algorithm for E-DNS. There are three important modules for E-DNS algorithms: communication module, recovering module and neighbor module. The communication module will check the server either up or down/fail, while the neighbor module will seek the appropriate neighbors when the primary server fails and the recovering module will update the failed of nameserver in DNS.

```
Main
 Read time.conf  #user define the time
 Do while time true
  Call communication-module
  If  died
    Call neighbor-module until true
 If true
    Call recovering-module
 End do
End Main

Procedure communication-module
 Create sockets
 Assign  port to sockets
 Get service
 Get IP
 Ping IP
End procedure

Procedure Neighbor-Module
 Read from NRDT able  (have logic structure by metric)
 Insert into array (#row,#cow)
 For i = 1 to 2
  For j  = 1 to 2
   If  i = 1
    If   j  = 1
     If  row  !=1
     New-metric =  cow –  1,row
   If  j = 2
    New-metric = cow + 1,row
  End if  #i = 1

  If   i =2
   If   j = 1
       New-metric =  cow, row – 1
   If   j = 2
```

```
            New-metric = cow, row +1
        End if # i= 2
      End for #i
    End for #j
End procedure

Procedure recovering-module
  Read  DNS text fail
  Seek IP dead  (#IP take from communication module)
  Replace new IP (#IP take from neighbor module)
  Re-run DNS software
End Procedure
```

4.2 Simulation of E-DNS

Assume that in a web sever cluster, there are six servers with the IP-address shown below:

1) p1.project.com is primary mail server with IP 999.999.9.111
2) p2.project.com is primary ftp server with IP 999.999.9.222
3) p3.project.com is primary web server with IP 999.999.9.333
4) p4.project.com is primary distance learning server with IP 999.999.9.444
5) p5.project.com is primary telnet server with IP 999.999.9.555
6) p6.project.com is primary other server with IP 999.999.9.666

1. Under normal *conFiguration* on BIND nameserver file, mapping address is simply done through the name to the IP address of single server as show below;

p1.project.com	IN	A	999.999.9.111
p2.project.com	IN	A	999.999.9.222
p3.project.com	IN	A	999.999.9.333
p4.project.com	IN	A	999.999.9.444
p5.project.com	IN	A	999.999.9.555
p6.project.com	IN	A	999.999.9.666

2. *ConFiguration* of the NRDT table consists of port, server name, IP and logical structure for every node (metric). The logical table *conFiguration* based NRDT is shown as.

80	p1.project.com	999.999.9.111	11
21	p2.project.com	999.999.9.222	12
23	p3.project.com	999.999.9.333	13
80	p4.project.com	999.999.9.444	21
80	p5.project.com	999.999.9.444	23
80	p6.project.com	999.999.9.444	24

3. The NRDT program will check the server either it is up or down/fail. For example, if the server with 999.999.9.222 (metric 12) fails, then its neighbors (could be with the

metrics 11,13,22) will be replaced (for example with IP address 999.999.9.333). The nameserver update becomes;

p1.project.com	IN	A	999.999.9.111
p2.project.com	IN	A	999.999.9.333
p3.project.com	IN	A	999.999.9.333
p4.project.com	IN	A	999.999.9.444
p5.project.com	IN	A	999.999.9.555
p6.project.com	IN	A	999.999.9.666

When requests come from Internet to browser ftp server, with the primary IP address 999.999.9.222, it will automatically mapped to the IP address 999.999.333. Thus, services on Internet are accessible at any point of time. Consequently, the reliability of the server is increased.

5 Conclusion

Web server cluster is a popular architecture used to improve the reliability and availability of the systems. However, with the current DNS algorithm, it is unreliable due to the fact that in a case of server(s) fails, users cannot access the data since no other server take over the service. In this paper, an enhanced domain name server (E-DNS) algorithm on NRDT technique, has been proposed to improve the reliability of the WSC. The algorithm is used as a mapping mechanism to dispatch the user requests through the URL-name to IP-address. It centralizes dispatching host name into single IP address and update it according to NRDT pattern. In case of server failures, the server will be switched to its neighboring IP address. The simulation showed that the proposed algorithm works well and provides a convenient approach to increase the reliability of WSC.

References

[1] M. Buretta, "*Data replication: Tools and Techniques for Managing Distributed Information*", John Wiley, New York, (1997).
[2] P. Albitz, and C.Liu, "DNS and BIND." *O'Reilly and Associates*, inc.,(2001).
[3] J. Liu, L Xu, B. Gu, J. Zhang, "Scalable, High Performance Internet Cluster Server", *IEEE 4th Int'l Conf.* HPC-ASIA, Beijing, pp. 941-944, (2000)
[4] M. Mat Deris, A. Mamat, P. C. Seng, H. Ibrahim, "Three Dimensional Grid Structure for Efficient Access of Replication Data", *Int'l Journal of interconnection Network, World Scientific*, Vol. 2, No. 3, pp 317-329, (2001).
[5] A. Moissis, "*Sybase Replication Server: A practical Architecture for Distributing and Sharinf Information*", Technical Document, Sybase Inc, (1996).
[6] E. Pacitti and E. Simon, " Update Propagation Strategies to Improve Fresh LazyMaster Replicated Databases", *Journal* VLDB, Vol. 8, no. 4, (2000).
[7] H. H. Shen, S. M. Chen, and W. M. Zheng,"Reserch on Data Replication Distribution Technique for Web Server Cluster", *IEEE Proc. 4th Int'l. Conference on Performance Computing,* Beijing, pp. 966-968, (2000).

[8] W. Zhou and A. Goscinski, “Managing Replicated Remote Procedure Call Transactions”, *The Computer Journal*, Vol. 42, no. 7, pp592-608, (1999).

[9] P. Mockapetris , “ Domain names-implementation and specication ,” Internet Request for Commnets (RFC 1035), (November 1987).

[10] Internet Software Consortium, ” Berkeley Internet domain (BIND)”,http://www.isc.org /product /BIND, (June 2000).

[11] P.A. Bernstein and N.Goodman, " An Algorithm for Concurrency Control and Recovery in Replicated Distributed Databases,"*ACM Trans. Database Systems*,vol 9,no. 4(1994),pp.596-615.

[12] M. Maekawa," A √n Algorithm for Mutual Exclusion in Decentralized Systems,"*ACM Trans. Computer Systems*,vol. 3,no. 2(1992), pp. 145-159.