

Co-Existence Neighbourhood Model for Optimizing Cloud Infrastructure as a Service (IaaS) Within Interdependent Environment

Ahmad Shukri Mohd Noor, Mustafa Mat Deris, and Md Yazid Md Saman

Abstract—The tendency of current large distributed systems such Cloud computing is the delivery of computing as a service rather than a product. Availability is the most important properties in Cloud computing. One of central issues in Cloud environment is to provide reliable Infrastructure as a service (IaaS) with optimal availability. Since the resources or nodes become larger, increasingly dynamic and heterogeneous, the potential for failures in the systems is a significant disruptive factor.

This paper proposes the twin co-existence neighbourhood (TCeN) model. It focuses on improving high availability in which it predicts future availability expectation of interdependent environment in a distributed system over an extended period of time. With less complexity, this model not only theoretical sound but also feasible for practical implementation in real environment, this technique is very cost effective with less complexity compare to other techniques. The performance results show TCeN availability outperformed the Double-replica technique(DRT) with 50% less in-term of storage overhead. Moreover, results show TCeN availability outperforms the Double-replica technique (DRT) up to 14.2% with less overhead

Index Terms—High availability, cloud computing, distributed computing, replication technique, dependable system.

I. INTRODUCTION

Availability is the nontrivial issue in distributed system [1]-[3]. Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing. Since cloud computing have become larger, increasingly dynamic and heterogeneous, the potential for failures in current cloud computing systems is a significant disruptive factor.

Replication is creating multiple copies of a possibly mutating object (file, file system, database, and so on) with the objective to provide high availability, high integrity, high performance, or any combination thereof. For high availability the replicas need to be diverse, so failures are sufficiently independent [1]. Replication is a fundamental

technique to achieve high availability in distributed and dynamic environments [3], [4].

One of the most popular replication techniques that currently implemented successfully is Double-replica Technique (DRT). DRT has been proposed by Shen *et al.* [5]. There are numbers of fault tolerant and failure recovery techniques based on DRT namely Netarkivet's data grid and fast disaster recovery mechanism for volume replication systems. This paper focus on improving high availability in which it calculates future expectation of interdependent servers availability for Cloud Infrastructure as a service (IaaS) over an extended period of time. In this paper, we propose the IaaS to be established with the Twin co-existence neighbourhood (TCeN) model. With less complexity, it is not only theoretical sound but also ideal for practical implementation in real environment, this technique is very cost effective with less complexity compare to other techniques. Moreover, the performance results show TCeN availability performs better then DRT with less overhead.

The rest of the paper is organized as follows: Section II presents the background of the research with DRT. The failure recovery problems faced by this technique were also mentioned here. The TCeN architecture details with the proposed failure recovery mechanism and dealing with node failure illustrated in Section III. Section IV discusses the recovery performance of proposed mechanism. The research findings and conclusion are given in Sections V and 6, respectively.

II. DOUBLE-REPLICA TECHNIQUE (DRT)

Most of replication techniques for fault tolerant and high availability are using Double-replica technique DRT [5], [6]. DRM has been proposed by Shen *et al.* [6]. In this technique, each node has an equal capacity of storage and all data have two replicas on different nodes and all nodes have two data replicas [6].

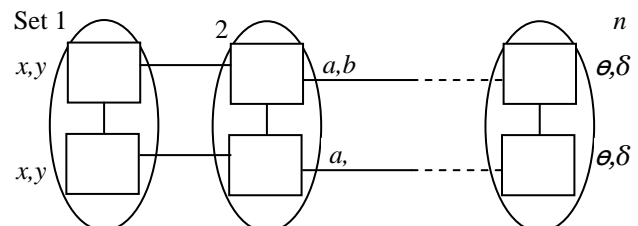


Fig. 1. Data replica distribution technique when $N = 2n$.

Manuscript received October 28, 2013; revised December 20, 2013.

Ahmad Shukri Mohd Noor is with the Department at Faculty of Science, Technology Universiti Malaysia Terengganu (UMT), Malaysia (e-mail: ashukri@umt.edu.my).

Mustafa Mat Deris is with the Faculty of Information Technology and Multimedia, UTHM, Malaysia.

Md Yazid Md Saman is with the Computer Science Department, Faculty of Science and Technology, Universiti Malaysia Terengganu (UMT), Malaysia.

With N nodes, it divides to n set of nodes ($N = 2n$) where each set consists of two nodes as illustrated in Fig. 1. The rectangle-shape shows the node, and each oval is a set that consists of two nodes. Data x and y have two replicas, that is in nodes from set 1, and data a and b also have two replicas which are located in nodes from set 2 and so on.

III. TWIN CO-EXISTANCE NEIGHBOURHOOD (TCeN) MODEL

The $N = 2n$ TCeN structure is a set of all nodes that are logically organized in a two-dimensional structure form. The TCeN model utilizes the structure of DRT which is all nodes are logically organized in the form of $N = 2n$ with certain exceptional technique.

This technique does not require any new additional node or storage for increasing high availability. Instead, the data will be replicated to logical adjacent neighbor.

IV. AVAILABILITY PREDICTION

Availability is the measure of how often or how long a service or a system component is available for use [7], [8].

$$\text{Availability} = \frac{\text{Operational}}{\text{Operational} + \text{Non-Operational}} \quad (1)$$

The *availability*, $A(t)$, of a component or system is defined as the probability that the component or system is operating at time t , given that it was operating at time zero.

The *unavailability*, $Q(t)$, of a component or system is defined as the probability that the component or system is not operating at time t , given that it was operating at time zero. Thus, $A(t) = 1 - Q(t)$.

Let be the number of nodes that are operating correctly at time t , be the number of nodes that have failed at time t , and N be the number of nodes that are in operation at time t a in [8]

$$A(t) = \frac{N_o(t)}{N} = \frac{N_o(t)}{N_o(t) + N_f(t)} \quad (2)$$

Similarly, the *unavailability* Q can be defined as

$$Q(t) = \frac{N_f(t)}{N} = \frac{N_f(t)}{N_o(t) + N_f(t)} \quad (3)$$

Of course, at any time t , $A(t) = 1 - Q(t)$.

The availability in series can be expressed as in 13

$$A = A_V \times A_Z \quad (4)$$

And the availability in parallel can be expressed as

$$A = 1 - (1 - A_V)(1 - A_Z) \quad (5)$$

If however there is mixed environment between parallel and serial the availability A can be defined as

$$A = (1 - (1 - A_W)(1 - A_X)) \times (1 - (1 - A_V)(1 - A_Z)) \quad (6)$$

This paper will compare TCeN model availability

prediction against DRT within interdependent environment. In real commercial environment, this prediction is very essential to develop and commit to service level agreements (SLAs). The terms stated in Service level agreements (SLAs) determine the degree of a system's high availability. A system that is designed to be highly available withstands failures that are caused by planned or unplanned outages. Service level agreements (SLAs) determine the degree of responsibility to maintain services that are available to users, costs, resources, and the complexity of the services. For example, a banking application that handles stock trading must maintain the highest degree of availability during active stock trading hours.

If the points of failure of the system are not analyzed, and then the system availability cannot be predicted correctly, thus SLA is flawed from the beginning.

A. Availability Prediction for Interdependent Infrastructure Environment

A single point of failure (SPOF) exists when a hardware or software component of a system can potentially bring down the entire system without any means of quick recovery. Highly available systems tend to avoid a single point of failure by using redundancy in every operation. The components for SPOF are arranged in serial architecture. In the basic form, the expected availability of a SPOF system equals the expected availability of each of the components of the system multiplied together. So if the system was composed of two servers, and each server had an expected availability of 99%, then the expected availability of the system would be $99\% \times 99\% = 98.01\%$. When a system is made up of n number of components that are each single points of failure then the system availability can be calculated as serial architecture as expressed in formula (4).

Consider the sample of distributed servers with single points of failure having the following eight components and the sample availability for each node as shown in Table I. A single point of failure of the distributed system can also be design in a parallel design or joint serial and parallel. The serial architectural design for the online distributed servers as illustrated in the Table I and Fig. 2.

TABLE I: SINGLE POINTS OF FAILURE WITH EIGHT COMPONENTS / NODES AND AVAILABILITY FOR EACH NODE

| Component | Availability |
|-------------------|--------------|
| Web | 0.95 |
| Application | 0.955 |
| Database | 0.95 |
| DNS | 0.97 |
| Firewall | 0.96 |
| Switch | 0.97 |
| Data Center | 0.95 |
| Applications2 | 0.95 |
| SPOF Availability | 0.703 |

A single point of failure of the distributed system can also be design in a parallel design or joint serial and parallel. The serial architectural design for the online distributed servers as illustrated in the Table I can be illustrated in Fig. 3. In SPOF, if any of these components

fails, the whole system will be unavailable. The expected availability of the site would be $0.95 \times 0.955 \times 0.95 \times 0.97 \times 0.96 \times 0.97 \times 0.95 \times 0.95 = 0.703$ or the availability the distributed system is 70.3%. With such a slightly low availability for highly accessible system, therefore the needs for high availability models are very significant that can discover alternative ways to increase the availability.

| Web | Application | Firewall | Switch | Database | Data Center | DNS | ISP | Total |
|------|-------------|----------|--------|----------|-------------|------|------|-------|
| N1 | N2 | N3 | N4 | N5 | N6 | N7 | N8 | 0.703 |
| 0.95 | 0.955 | 0.95 | 0.97 | 0.96 | 0.97 | 0.95 | 0.95 | |

Fig. 2. The serial architectural design for the distributed servers.

B. DRT Availability Prediction Model

The DRT availability prediction model adds to each of the servers with second replica. When a system is comprised of two redundant components, then the availability of the system can be calculated by using parallel formula as expressed (5). Thus the DRT can be define as

$$A_s = 1 - (1 - A_n)^2 \quad (7)$$

| Web | Application | Firewall | Switch | Database | Data Center | DNS | ISP | Total availability |
|------------|-------------|------------|------------|------------|-------------|------------|------------|--------------------|
| N1 0.95 | N2 0.955 | N3 0.95 | N4 0.97 | N5 0.96 | N6 0.97 | N7 0.95 | N8 0.95 | 0.9975 |
| N1 0.95 | N2 0.955 | N3 0.95 | N4 0.97 | N5 0.96 | N6 0.97 | N7 0.95 | N8 0.95 | 0.9975 |
| 0.9975 | 0.9979 | 0.9975 | 0.9991 | 0.9984 | 0.9991 | 0.9975 | 0.9975 | 0.9695 |

Fig. 3. DRT architectural design for interdependent servers.

Using the example of a web server in Table I, with an availability of 95%, and with an additional another web server as a backup server would increase the availability to: $1 - (1 - 0.95) \times (1 - 0.95) = 0.9975$. Thus the availability of the web server increases from 95% to 99.75%. Imagine that we add to each of the servers with second replica or backup server as illustrated in DRT architectural design in Fig. 4. Table II demonstrates how to calculate the DRT high availability for each server and total availability for the whole system. Now, the availability of the whole system would be: $0.9975 \times 0.99798 \times 0.9975 \times 0.9991 \times 0.9984 \times 0.9991 \times 0.9975 \times 0.9975 = 0.9847$.

TABLE II: DRT HIGH AVAILABILITY CALCULATION

| Site | Server | Available | Unavailable | Availability | Total |
|--------------------|------------------------|-----------|-----------------|---------------------|---------|
| N1 | Web 0.95 | 95.0% | $(1 - 0.95)^2$ | $1 - (1 - 0.95)^2$ | 0.99750 |
| N2 | Application 0.955 | 95.5% | $(1 - 0.955)^2$ | $1 - (1 - 0.955)^2$ | 0.99798 |
| N3 | Firewall 0.95 | 95.0% | $(1 - 0.95)^2$ | $1 - (1 - 0.95)^2$ | 0.9975 |
| N4 | Switch 0.97 | 97.0% | $(1 - 0.97)^2$ | $1 - (1 - 0.97)^2$ | 0.9991 |
| N5 | Database 0.96 | 96.0% | $(1 - 0.96)^2$ | $1 - (1 - 0.96)^2$ | 0.9984 |
| N6 | Data Center 0.97 | 97.0% | $(1 - 0.97)^2$ | $1 - (1 - 0.97)^2$ | 0.9991 |
| N7 | DNS 0.95 | 95.0% | $(1 - 0.95)^2$ | $1 - (1 - 0.95)^2$ | 0.9975 |
| N8 | ISP 0.95 | 95.0% | $(1 - 0.95)^2$ | $1 - (1 - 0.95)^2$ | 0.9975 |
| Total Availability | | | | | 0.9695 |

Removing single points of failure improved the availability of the system from 69.56% to 98.46%. The result shows that the availability increase is not trivial. However it still has a lack, this is due to DRT has to double up the resource.

C. TCeN Availability Prediction Model

Typically, the cloud computing infrastructure (IaaS) will be comprised of multiple components, some with parallel or serial or both in a system, and each with different levels of component. Modeling these requires slightly more complicated as each component weighted with different availability, thus in real environment the use of simple k -out-of- n model is quite uncommon. Thus the TCeN can be define as

$$A_s = 1 - ((1 - A_1)(1 - A_2) \dots (1 - A_n)) \quad (8)$$

Assume that we apply the TCeN model to each of the servers as illustrated in TCeN design combines parallel with serial architecture in Fig. 4. Table III demonstrates how to calculate the TCeN high availability for each server and total availability for the whole system.

| Web | Application | Firewall | Switch | Database | Data Center | DNS | ISP | Total Availability |
|-------------|-------------|------------|------------|------------|-------------|------------|------------|--------------------|
| N1 0.95 | N2 0.955 | N3 0.95 | N4 0.97 | N5 0.96 | N6 0.97 | N7 0.95 | N8 0.95 | 0.9837 |
| N2 0.955 | N1 0.95 | N4 0.97 | N3 0.95 | N6 0.97 | N5 0.96 | N8 0.95 | N7 0.95 | 0.9837 |
| 0.99775 | 0.99775 | 0.9985 | 0.9985 | 0.9988 | 0.9988 | 0.9975 | 0.9975 | 0.9837 |

Fig. 4. TCeN merges parallel with serial architecture.

The Table III demonstrates that TCeN model increase availability for the system better then DRT model. Moreover TCeN maintain the same number of resources.

TABLE III: TCeN HIGH AVAILABILITY CALCULATION

| Site | Server | Availability | Total |
|-------|------------------------|----------------------|---------|
| N1 | Web 0.95 | $1 - ((1-N1)(1-N2))$ | 0.99775 |
| N2 | Application 0.955 | $1 - ((1-N1)(1-N2))$ | 0.99775 |
| N3 | Firewall 0.95 | $1 - ((1-N3)(1-N4))$ | 0.9985 |
| N4 | Switch 0.97 | $1 - ((1-N3)(1-N4))$ | 0.9985 |
| N5 | Data base 0.96 | $1 - ((1-N5)(1-N6))$ | 0.9988 |
| N6 | Data Center 0.97 | $1 - ((1-N5)(1-N6))$ | 0.9988 |
| N7 | DNS 0.95 | $1 - ((1-N7)(1-N8))$ | 0.9975 |
| N8 | ISP 0.95 | $1 - ((1-N7)(1-N8))$ | 0.9975 |
| Total | | | 0.9837 |

V. PERFORMANCE RESULTS

This section focus on the comparison between TCeN and DRT in improving high availability of the interdependent servers in which it predicts future expectation of the interdependent servers availability over an extended period of 10 years time. Let consider a Cloud Infrastructure with the server's availability as illustrated in Table IV. It shows comparison of the availabilities predictions between TCeN and DRT for a year.

TABLE IV: THE AVAILABILITIES COMPARISON BETWEEN TCeN AND DRT

| Node | Servers | Availability (A) | Unavailable (Q) | DRT Availability | TCeN Availability |
|---------------------------------|-------------|------------------|-----------------|------------------|-------------------|
| 1 | Web | 0.94 | 0.060 | 0.99750 | 0.99775 |
| 2 | Application | 0.95 | 0.050 | 0.99798 | 0.99775 |
| 3 | Firewall | 0.95 | 0.050 | 0.9975 | 0.9985 |
| 4 | Switch | 0.91 | 0.090 | 0.9991 | 0.9985 |
| 5 | Database | 0.9 | 0.100 | 0.9984 | 0.9988 |
| 6 | Data Center | 0.93 | 0.070 | 0.9991 | 0.9988 |
| 7 | DNS | 0.92 | 0.080 | 0.9975 | 0.9975 |
| 8 | ISP | 0.94 | 0.060 | 0.9975 | 0.9975 |
| Total Un-Availability (Q_s) | | | | 0.0305 | 0.0163 |
| Total Availability (A_s) | | | | 0.9695 | 0.9837 |

The DRT and TCeN availability for the second year onwards can be estimated based on the total availability results specified Table IV. For instance, the Table IV shows the TCeN availability for the first year (A_s) is as 0.9837. Therefore the availability prediction for second year (A_{y2}) can be calculated as $A_{y2} = 1 - 2Q_s$, for the third year $A_{y3} = 1 - 3Q$ and so forth[15]. The same calculation also been applied to DRT availability. The availability

prediction over an extended period of 10 year for TCeN and DRT is shown in Table V and Fig. 5, respectively. While the Table IV shows the availability comparison for TCeN and DRT over the same period of times.

TABLE V: AVAILABILITY PREDICTION FOR 10 YEAR

| Year | DRT | TCeN |
|------|--------|--------|
| 1 | 0.9695 | 0.9837 |
| 2 | 0.939 | 0.9674 |
| 3 | 0.9085 | 0.9511 |
| 4 | 0.878 | 0.9348 |
| 5 | 0.8475 | 0.9185 |
| 6 | 0.817 | 0.9022 |
| 7 | 0.7865 | 0.8859 |
| 8 | 0.756 | 0.8696 |
| 9 | 0.7255 | 0.8533 |
| 10 | 0.695 | 0.837 |

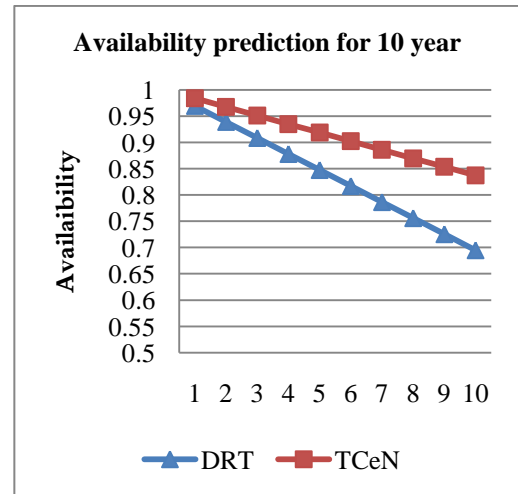


Fig. 5. Availability prediction for 10 year.

From the Fig. 5 observation, it demonstrates that, as the years goes by the availability gap is apparently larger and larger. This is especially for DRT, the DRT availability reduce about 1.42% per year or 14.2% for ten years. However the TCeN availability reduces about 0.12% per year or 1.2% for period of ten years. The graph plotted in Fig. 6 demonstrated the availability gap between NDRT and DRT for 10 years. Moreover, the performance results show TCeN availability outperformed the Double-replica technique(DRT) with 50% less in-term of storage overhead.

VI. CONCLUSION

Replication of data or processes is an effective way to provide high availability cloud computing. Throughout this paper, availability of TCeN and DRT for interdependent server environment is modeled and analyzed. It has shown that DRT can eliminate single points of failure as well as improve the availability. However DRT improvement comes with higher overhead cost, since DRT need to double up the number of servers in the system. On the other the analysis demonstrates that TCeN model able to increase

availability for the system better than DRT model without introducing new overhead cost. The analysis of the models shows that the TCeN theoretically sound as well as practically feasible for providing high availability within interdependent environment of the Cloud Infrastructure. Thus the implementation of interdependent environment of the Cloud Infrastructure

REFERENCES

- [1] R. V. Renesse and R. Guerraoui, "Replication techniques for availability," in *Lecture Notes in Computer Science*, B. Charron-Bost, R. Pedone, and A. Schiper, Eds. Replication, vol. 5959, 2010, pp. 19–40.
- [2] Y. Wang, Z. Li, and W. Lin, "A fast disaster recovery mechanism for volume replication systems," in *Lecture Notes in Computer Science*, R. Perrot *et al.* Eds. HPCC 2007, vol. 4782, 2007, pp. 732–743.
- [3] A. S. M. Noor and M. M. Deris, "Failure recovery mechanism in neighbor replica distribution architecture," in *Lecture Notes in Computer Science*, R. Zhu *et al.*, Eds. ICICA 2010, vol. 6377, 2010, pp. 41–48.
- [4] S. Bora, "A fault tolerant system using collaborative agents," in *Lecture Notes in Artificial Intelligence*, F. A. Savaci *et al.*, Eds. TAINN 2005, vol. 3949, 2006, pp. 211–218.
- [5] H. H. Shen, S. M. Chen, W. M. Zheng, and S. M. Shi, "A communication model for data availability on server clusters," in *Proc. International Symposium on Distributed Computing and Application*, 2001, pp. 169–171.
- [6] R. Mamat, M. M. Deris, and M. Jalil, "Neighbor replica distribution technique for cluster server systems," *Malaysian Journal of Computer Science*, vol. 17, no. 2, pp. 11–20, 2004.
- [7] L. Parziale, A. Dias, L. T. Filho, D. Smith, J. V. Stee, and M. Ver, "Achieving high availability on linux for system z with linux-ha," Release 2, IBM Corp, 2009.
- [8] *Guidelines to Understanding Reliability Prediction*, European Power Supply Manufacturers Association, Wellingborough, Northants, UK, 2005.

Ahmad Shukri Mohd Noor is a senior lecture of Computer science Department at Faculty of Science and Technology Universiti Malaysia Terengganu (UMT). Currently, he is a postdoctoral researcher with Distributed, Reliable and Intelligent control and cognitive Systems group (DRIS) at Department of Computer Science, Faculty of Science and Engineering, University of Hull, United Kingdom. He received a B.Sc. in Computer Science from Coventry University 1997, M.Sc. MSc in High Performance System from University College Science and Technology Malaysia 1989 and Ph.D from University Tun Hussien Onn Malaysia 2012. His research interests include distributed computing, data grid and Application system development

Mustafa Mat Deris is a professor of computer science in the Faculty of Information Technology and Multimedia, UTHM, Malaysia. He received a B.Sc. from University Putra Malaysia 1982, M.Sc. from University of Bradford 1989, England and Ph.D from University Putra Malaysia 2001. His research interests include distributed databases, data grid, database performance issues and data mining. He has appointed as one of the editorial board members for International Journal of Information Technology, World Enformatika Society, International Journal of Applied Science, Engineering and Technology (IJSET), International Journal of Electrical, Computer, and Systems Engineering (IJCSE), International Journal of Applied Mathematics and Computer Sciences (IJAMCS) and Encyclopedia on Mobile Computing and Commerce, Idea Group, USA; Apart from that he is also a reviewer for several International Journals such as Journal of Parallel and Distributed Databases,

Md Yazid Mohd Saman is a professor of Computer Science Department at Faculty of Science and Technology Universiti Malaysia Terengganu (UMT). He received B.Sc from Essex university, M.Sc from UTM and Ph.D from Loughborough University, United Kingdom, his areas of expertise distributed & parallel computing; computer network; Simulation & performance modeling.