International Journal of Engineering & Technology, 7 (3.28) (2018) 5-7

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

Intelligent Replication Strategy using Neural Network for High Availability in Cloud Environment

Noriha Hasni[1*], Zarina Mohamad[1], Fadhilah Ahmad[1], Wan Nor Shuhadah Wan Nik[1], Mustafa Mat Deris[2]

[1]Department of Computer Science, Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Terengganu, Malaysia
[2]Faculty of Information Technology and Multimedia University Tun Hussein Onn Malaysia, Batu Pahat, Johor, Malaysia
**Corresponding author E-mail: capricorndabest@yahoo.com*

Abstract

Cloud computing system has gained its popularity due to its ability to produce efficient data center storage management. Data center has to be available regardless of large amounts of data request. The performance of the system will be degraded if lack of data management in the data center. Data replication is one of the strategies to ensure the data is always available once it is needed. Thus, in this paper will propose an intelligent replication strategy in private cloud environment using Neural Network (NN) Feed Forward and Back Propagation methods. The performance of the propose strategy is analyzed by using a cloud simulator called Cloudsim. A data center, virtual machines and cloudlets are tested for their ability to produce replication by applying Feed Forward NN for pattern recognition, followed by Back Propagation NN to train and produce the Root Mean Square Error. Hence, the Root Mean Square is used to produce data availability and the data availability is used as the triggering factor in this replication strategy. The experimental result shows that the proposed strategy has reduced the response time of replication process and enhanced the availability of the data in cloud computing system.

Keywords: *Cloud Computing; Neural Network; Replication; Data Availability; Data Center*.

1. Introduction

Cloud computing is a system that has large scale parallelism and can be distributed widely across the globe [1]. It has gained popularity due to its capability to manage large storage of data.
Data center is a basic unit in cloud computing, which acts as a large scale storage that influences the performance of the system [2]. One of the characteristics that will be considered in data center is data availability. Data availability is to ensure that the data continue to be available for user access regardless of time and space boundaries. High availability is an important specification in a service level agreement in cloud services [3]. For example, the availability of the Amazon EC2 is 99.95%, the availability of Google Cloud Storage is 99.9% and the availability of Microsoft Windows Azure is 99.9% as per claim by [4]. One of the approaches for data availability is via data replication, which can be implemented in the data center.
Data replication is used to manage large data by increasing its availability [5]. The processes involve for data replication includes which data needs to be replicated, the number of replicas needs to be created, and where the location of these replicas are after they have been created [6].
3-replicas data replication strategy (3DRS) is the popular replication strategy in the cloud computing service. Data storage systems such as Amazon S3, Google File System and Hadoop Distributed File System have adopted a 3DRS by default for reliability purpose [7]. 3DRS in data center means once the master data are added in a data center, the data center will create another two replicas. The issue of the 3DRS is that, extra storage capacity is needed to replicate the data. Hence, additional cost is required for efficient storage management.
Thus, this paper proposed the combination of 3DRS with Neural Network (NN) for intelligent replication strategy called Storage Availability Incremental Replication Strategy (SAIRS). The main objective of this study is to increase the efficiency of data availability in the data center. SAIRS calculates Data Availability (DA) of the data by using the Feed Forward and Back Propagation methods in NN. The result indicates that SAIRS has the capability to intelligently perform the data replication.
This paper is structured as follows. Section 2 describes the system model. Section 3 contains the experimental result of the propose method. Conclusion and future work have been stated in Section 4.

2. Methodology

This section shows the details of system model, SAIRS algorithms and pseudo code.

2.1. System Model

Proposed cloud system consists of data centers, DC = $\{DC_1, DC_2, DC_3, \ldots, DC_i\}$ [8]. The data centers parameters are Ram, Unit of Storage and bandwidth. Those data centers consist datacenter broker and virtual machines, VM = $\{vm_1, vm_2, vm_3, \ldots, vm_i\}$. Datacenter broker is used to manage the VM. The parameters of the VMs are VM Id, User's Id, Mips, Processing Element, Ram, Bandwidth, Size and Name. Cloudlet Scheduler Time Shared also needs to be set in those VM. Cloudlet Scheduler Time shared able to balance the core of VM [9].
Cloudlet c = $\{c_1, c_2, c_3, \ldots, c_i\}$, represents the data in this system. The cloudlet parameters are Cloudlet Id, Cloudlet Length,

Number of Processing Element for Cloudlet, Cloudlet Output Size and Cloudlet Input Size. Those cloudlets are managed by Multi Criteria algorithm [2]. After the modeling of the cloud system, the next stages focus on SAIRS algorithm and pseudo code.

2.2. Storage Availability Incremental Replication Strategy (SAIRS)

In this paper, SAIRS has been enhanced with a NN prediction method to achieve data available in the cloud computing system.

2.2.1. Phase One: Pattern Creation Algorithm

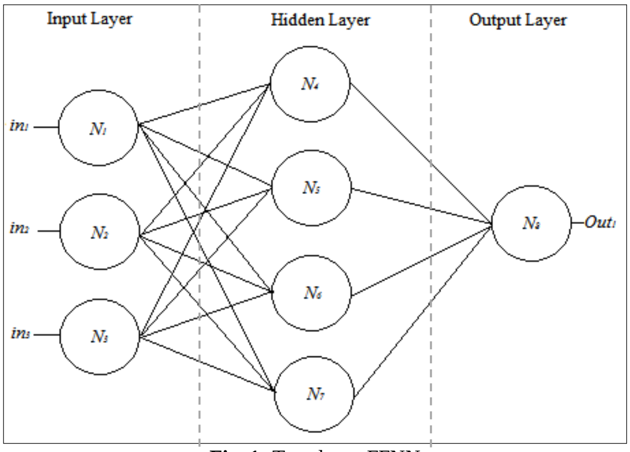

Fig. 1: Two-layer FFNN

Figure 1 shows two-layer FFNN applied to create pattern in SAIRS. Cloudlet Output Size, n_1, Cloudlet Length, $n_2$and Cloudlet Processing Element Number, n_3 are used as inputs in FFNN. Then, calculate the Internal Value, U_j [10] of the layer by using:

$$\mathrm{U}_j = \sum_j \mathrm{w}_{ij}\mathrm{n}_i \tag{1}$$

which w_{ij} is established weight for input, $i^{th}$and neuron $j\ ^{th}j$. The sum of w_{ij} and n_i produce U_j. U_j and established threshold, t is sent through activation function to produce output of the layer, o_j [11] :

$$o_j = f\ (t + U_j) \tag{2}$$

2.2.2. Phase Two: Back Propagation Training

BPNN is used to train the cloudlet. From the output layer, BPNN starts with this equation:

$$w_{ij} = w'_{ij} + LR * e_j * n_i \tag{3}$$

Weight w_{ij} is adjusted by previous weight w'_{ij}, learning rate, LR, error, $e_j$and input, n_i [12].

$$e_j = o_j * \left(1 - o_j\right) * (o_p - o_j) \tag{4}$$

Error term is adjusted to output, o_j multiply with one minus o_j and multiply with output prediction, o_p minus o_j. Then, the error term for the next layer, e_j is modified:

$$e_j = o_j * \left(1 - o_j\right) * \sum(e_k * w'_{jk}) \tag{5}$$

which o_j multiply with one minus o_j and multiply with sum of previous error term, e_k multiply with previous weight adjustment, w'_{jk}.

2.2.3. Phase Three: Network Error

After the BPNN training, Network Error is indicated. First, Total-Sum-Squared-Error (TSSE) is calculated using:

$$TSSE = \frac{1}{2}\sum_p \sum_o (o_p - o_j)^2 \tag{6}$$

which p is pattern and o_p is output prediction. Then, Root Mean Square Error, RMSE calculates using:

$$RMSE = \sqrt{\frac{2*TSSE}{p*o_j}} \tag{7}$$

RMSE is used in DA calculation:

$$DA = \left(1 - RMSE^{\sqrt{n_1}}\right)^{\sqrt{n_1}} \tag{8}$$

DA is used as the triggering factor in SAIRS.

2.2.4. Phase Four: SAIRS Pseudo Code

Based on SAIRS, once a cloudlet has entered a data center, the cloudlet will be trained by NN. The cloudlet is replicated when DA is more or equals to 0.99999 as stated by [13]. The first replica cloudlet will go to another data center. Then, the Figure 2 shows the SAIRS in cloudsim.

```
Input : c_i
    DEFINE i = 0
    INITIALIZE c_i
    ADD c_i into DC_i
    Train NN c_i
    IF DA of c_i ≥ 0.99999 THEN
      Replicate c_i → c_{i+1}
     DC_i DETERMINE DC_{i+1}
      ADD c_i into DC_{i+1}
    FINISH
```

Fig. 2: Pseudo code of SAIRS in Cloud Computing

3. Results and Discussion

The effectiveness of SAIRS is evaluated in this section. The SAIRS is implemented in Cloudsim. Two DC are using in this cloudsim, while other parameters list in table below:

Table 1: Parameters setting for CloudSim simulator as using by [14]

Component Type	Parameter	Value
DC	Number of Host Scheduling Policy	c1 Space-Shared
DC Broker	Number of DC Broker	1
VMs	Number of VMs Number of Processing Element (PE) MIPS of PE MIPS of VMS VM 1 VM 2 VM 3 VM 4 VM RAM size VM Scheduling Policy Bandwidth	4 1 2000 200 300 400 500 2048MB Space-Shared 10000
Cloudlet	Number of Cloudlets CL 1 CL 2 CL 3 CL 4 CL 5 CL 6 CL 7 CL 8	10 5200 3000 2000 7000 6800 6600 5800 5400

	CL 9	5000
	CL 10	4800
	Cloudlet Scheduler Policy	Space-Shared
	CPU Utilization Model	Full Utilization of CPU
	RAM Utilization Model	Full Utilization of RAM
	Bandwidth Utilization Model	Full Utilization of Bandwidth

RMSE of cloudlet is produced after 3 times of NN training. RMSE is used in the DA algorithm as to produce triggering factor for SAIRS. Refer table below for SAIRS result.

Table 2: SAIRS result

Cloudlet ID	Replicate
0	√
1	√
2	√
3	√
4	X
5	√
6	√
7	√
8	√
9	X

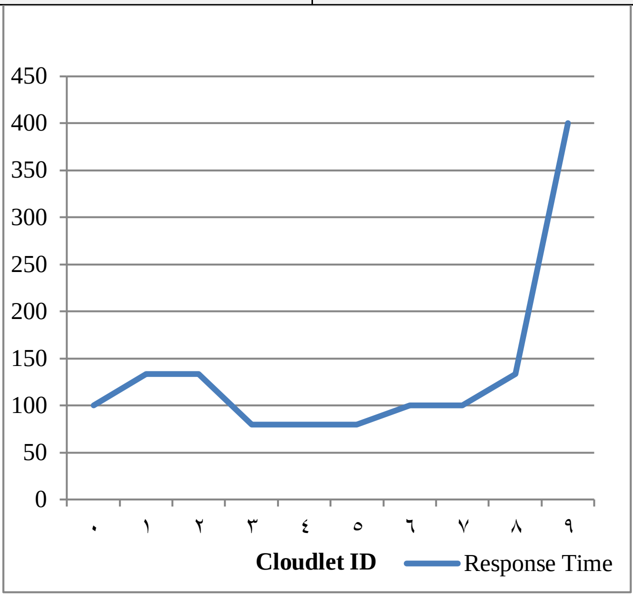

Fig. 3: SAIRS response time

Based on the SAIRS replication and response time result, there is 80% of cloudlets are success to be replicated with average response time is 133.99.

4. Conclusion

In this purpose, it proposes SAIRS approach. This method applies as another alternative replication strategy to ensure data availability in cloud computing system. In the experiment, the result shows the improvement of the data availability in the Cloudsim. However, there are a lot of research needs to be done in this method. In the future, we will do comparison SAIRS with another method.

Acknowledgement

Appreciation goes to the Ministry of Science, Technology and Innovation, MOSTI, Universiti Sultan Zainal Abidin, supervisor and team for the support of this project.

References

[1] Mohamed-K Hussein & Mohamed-H Mousa (2012), A Light-weight Data Replication for Cloud Data Centers Environment, International Journal of Engineering and Innovative Technology, Vol.1, No.6, 169-175.

[2] Wenhao Li, Yun Yang & Dong Yuan (2011), A Novel Cost-effective Dynamic Data Replication Strategy for Reliability in Cloud Data Centres, *IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing*, 496-502.

[3] Xu Wang, Hailong Sun, Ting Deng & Jinpeng Huai (2015), On the tradeoff of availability and consistency for quorum systems in data center networks, Computer Networks, Vol.76, 191–206.

[4] Rui Li, Wei Feng, Huayi Wua & Qunying Huang (2014), A replication strategy for a distributed high-speed caching system based on spatiotemporal access patterns of geospatial data, Computers, Environment and Urban Systems, Vol.61, Part B, 163-171.

[5] Sai-Qin Long, Yue-Long Zhao & Wei Chen (2014), MORM: A Multi-objective Optimized Replication Management strategy for cloud storage cluster, Journal of Systems Architecture, Vol.60, No.2, 234-244.

[6] Saiqin Long, Yuelong Zhao & Wei Chen (2013), A three-phase energy-saving strategy for cloud storage systems, The Journal of Systems and Software, Vol.87, 38-47.

[7] Wenhao Li, Yun Yang & Dong Yuan (2011), A Novel Cost-effective Dynamic Data Replication Strategy for Reliability in Cloud Data Centres, *IEEE Ninth International Conference on Dependable, Autonomic and Secure Computing*, 496-502.

[8] Navneet Kaur Gill & Sarbjeet Singh (2015), Dynamic Cost-Aware Re-replication and Rebalancing Strategy in Cloud System, 3rd International Conference on Frontiers of Intelligent Computing: Theory and Applications, 39-47.

[9] Ranjan Kumar & G.Sahoo (2014), Cloud Computing Simulation Using CloudSim, International Journal of Engineering Trends and Technology, Vol.8, No.2, 82-86.

[10] Ross Berteig (2013), Basic Concepts for NNs. NN Technology. Neuralyst, https://www.chesh-ireeng.com/Neuralyst/nnbg.htm.

[11] Frauke Günther & Stefan Fritsch (2010), neuralnet: Training of NNs, *The R Journal*, Vol. 2/1, 30-38.

[12] Ross Berteig (1996), Neuralyst™ Implementation Details. NN Technology, Neuralyst, https://www.cheshireeng.com/Neuralyst/doc/formulae.htm .

[13] Jiawei Yuan & Shucheng Yu (2014), Privacy Preserving Back-Propagation NN Learning Made Practical with Cloud Computing, IEEE transactions on parallel and distributed systems, Vol.25, No.1, 212-221.

[14] M. Aminu, M. Zarina, W. Nor & A. Fadhilah (2015), Multi-Criteria Strategy for Job Scheduling and Resource Load Balancing in Cloud Computing Environment, Indian Journal of Science and Technology, Vol.8, No.30.