ISSN: 0975-766X
CODEN: IJPTFI
Available Online through *Research Article*
www.ijptonline.com

SIMULATION OF CLOUD COMPUTING FAULT TOLERANCE USING CLOUD ANALYST AND SIMUBRAIN TOOLS

Lokesh VK[1] , J. Jabanjalin Hilda[2]
[1,2]School of Computer Science and Engineering, VIT University, Vellore, Tamil Nadu, India.
Email: jabanjalin.hilda@vit.ac.in

Received on: 10-02-2017 *Accepted on: 22-03-2017*

Abstract

Background and Objective: Cloud computing has become inseparable from modern day businesses. It provides dynamically scalable integration of software and resources. This dynamic nature cause unprecedented faults in the cloud. Fault tolerance is the capacity of the system to detect and rectify the occurred fault smoothly. **Methods:** This paper discusses and compares various papers and methodologies proposed to detect and rectify fault in cloud. This paper also proposes an algorithm based on artificial neural networks to detect fault tolerance. **Result and Conclusion**: The Cloud Analyst and Simubraintool calculates the Response time and determine the error factor. The Artificial Neural Network (ANN) approach gives a better response time and the error is abridged by using this method.

Keywords- ANN, Cloud Analyst, Fault tolerance, Fault detection, Simubrain.

1. Introduction

Cloud computing provides computer services over the internet. Instead of keeping the resource and application in our own computer we can store and access the application from some other computer in another location. Cloud computing provides reusability [1,2]. Cloud computing provides networking, virtualization, utility computing, web services, distributed computing and software services. Cloud computing consists of 3 elements: Clients, Data center and Distributed servers. Clients are basically end users; data center comprises of collection of servers which provide the service and distributed servers are servers located in various locations. There are different clouds classified based on the access criteria: public, private, hybrid and community cloud [3]. Cloud service providers offer three kinds of services vizinfrastructure as a service, platform as a service andsoftware as a service, The characteristics of the cloud can be summed up as quality of service, pricing, virtualization, security, fault tolerance, device independence, location

independence, on-demand services, multi-tenancy and scalability [4]. Cloud computing is preferred mainly because of its ability to minimize response time, reduce running time of a task, cost, risk and increase intelligence of the machine.

2. Fault tolerance in cloud computing

A device is said to be fault tolerant if it is able to function in a permissible manner even if one or more parts of it are not functioning or malfunctioning. Fault tolerance is highly important for a cloud system as a cloud system is heterogeneous and even if one or more components of it fail the system should be able to function properly. It considers various parameters such as availability, security, usability, over-head, response-time, throughput.

The different fault tolerance techniques that are used in cloud computing are: Reactive and proactive fault tolerance [5].

2.1. Reactive fault tolerance

When the failure of the system has previously occurred, Reactive fault tolerance technique comes into picture. Various techniques based on this policy are task resubmission, retry, user defined exception handling, job migration, rescue overflow, replication, check pointing.

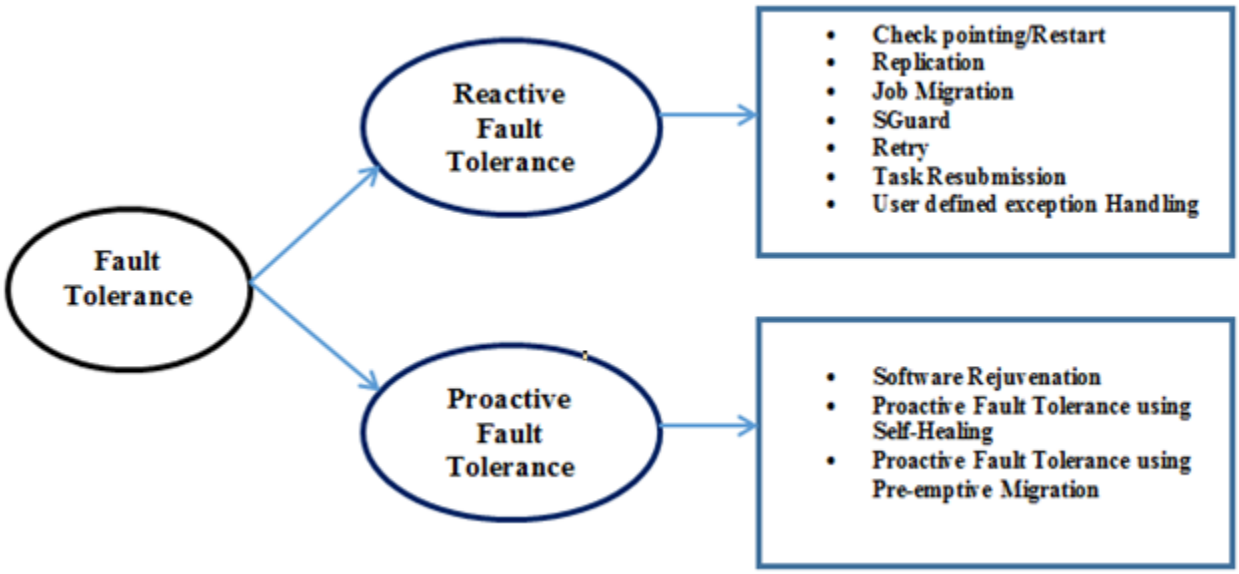


Figure 1: Classification of fault tolerance.

2.2. Proactive Fault Tolerance:

This technique finds out the fault beforehand and replaces the malfunctioning constituents and recovers the process. Self-healing, preemptive migration and software rejuvenation follow this technique[6].

Failure detector: It is an application that detects failures in the cloud system. It can be of two types reliable and unreliable. If output reliable then reliable detector otherwise unreliable detector[7]. The correctness properties of detectors are: completeness, accuracy, speed, scale, detection time, mistake recurrence time and duration of the mistake.

3. Related Work

Heartbeat Strategy for failure detection is the most commonly used technique for fault detection. In this technique, there is a monitoring process q to which all the other processes p send "I am alive" message after a certain interval of time continuously[8]. On the off chance that the message is not got by the checking procedure q from p after the settled time, then the q adds p to a rundown of suspected procedures. In the event that after some time q gets the message from p then it will expel the procedure from the suspected list.

Chandra and Toug [9] put forwarddevelopment to the existing strategy in their paper. They used fixed time points to determine whether a process p should be placed in the suspecting list by the monitoring process q. The fixed time point or freshness point is an approximation of the onset time of the message from p. Detection time is independent of the previous message in this algorithm.

Chen FD [10] proposed in his paper that in order to estimate the onset time of the next heart beat we sample the onset time of the message in recent past. Thus, the sampled value is set as the arrival time along with a safety margin.

Bertier FD [11,12] in his paper suggested that the freshness points can be dynamically calculated using Jacobson's estimation. Safety margin adaptive is based on the mutable error in the last approximation.

4. Methodology

An extended survey was done on various fault tolerance techniques available at present for cloud computing. A new technique using artificial neural networks was implemented, analyzed and results obtained.A broad review was done for different models of artificial neural networks which can be utilized for fault discovery. Our proposed detector depends on Heartbeat methodology which utilizes Artificial Neural Network for the estimation of expected onset time from a virtual machine[13].

The observing procedure q utilizes an expected value which passes on q how much time it needs to weight for the following heart beat message from a procedure p. On the off chance that after the assessed time q does not get the heart beat message from p, it will begin speculating p. The estimated time is permitted to change after some time to make it

versatile with genuine correspondence loads. The estimated time permitted is registered by a simulated neural system which will help the finder to stay away from false discoveries.

The tools used to simulate the cloud are cloud analyst and Simubrain.A cloud network of 1 data center and 4 user base is created and simulated in cloud analyst.

The response time is calculated.A back propagation network is created in Simubrain. The calculated response time is given as the input data to the BPN and the target data is fixed as the desired response time and the network is trained with activation as 1. The number of iterations and the error rate determines whether the user base will send the message before deadline or not.

5. Analysis and Result:

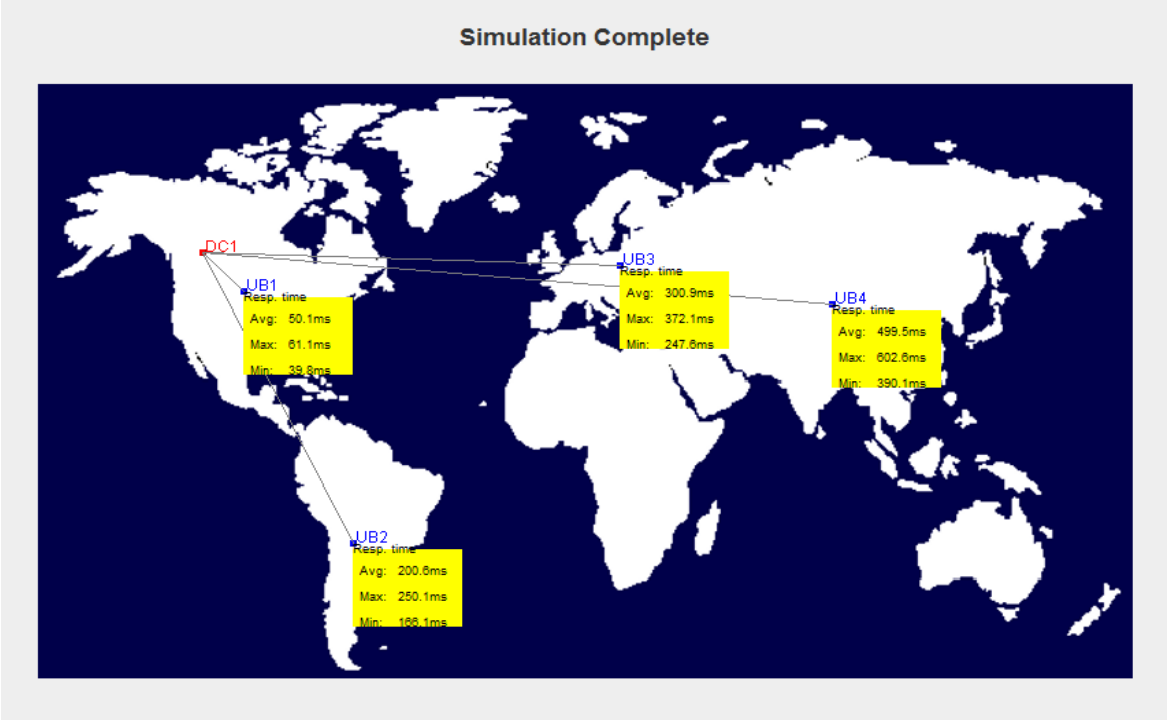
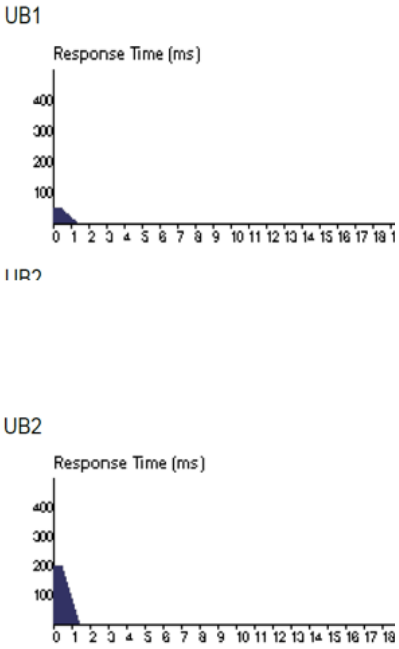


Figure 2: Simulated cloud network.

Metrics	Avg(ms)	Min(ms)	Max(ms)
Overall Response Time:	261.19	39.81	602.61
Data Center Proecessing Time:	0.32	0.01	0.86

Table 1: Overall Response Time SummaryTable 2: Response Time by Region.

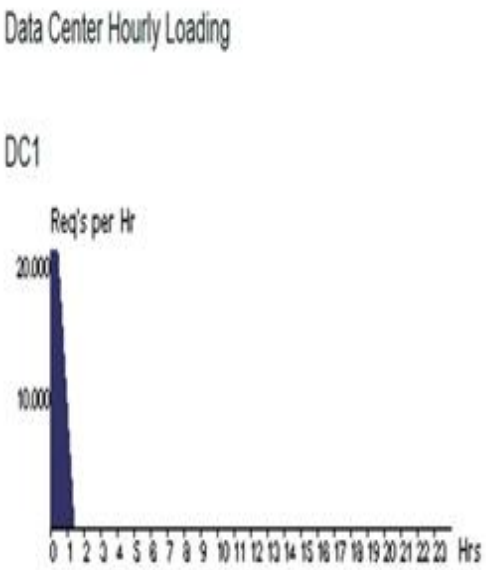


Userbase	Avg(ms)	Min(ms)	Max(ms)
UB1	50.11	39.81	60.58
UB2	200.75	166.11	250.11
UB3	300.86	247.61	372.12
UB4	499.57	390.12	602.61

Figure 3: Average response time for each node.

Table 3: Data Center Request Servicing TImes

Data Center	Avg(ms)	Min(ms)	Max(ms)
DC1	0.32	0.01	0.86



Cost

Total Virtual Machine Cost ($):	0.51
Total Data Transfer Cost ($):	0.22
Grand Total: ($)	0.73

Data Center	VM Cost $	Data Transfer Cost $	Total $
DC1	0.51	0.22	0.73

Figure 4: Data Center VM Cost.

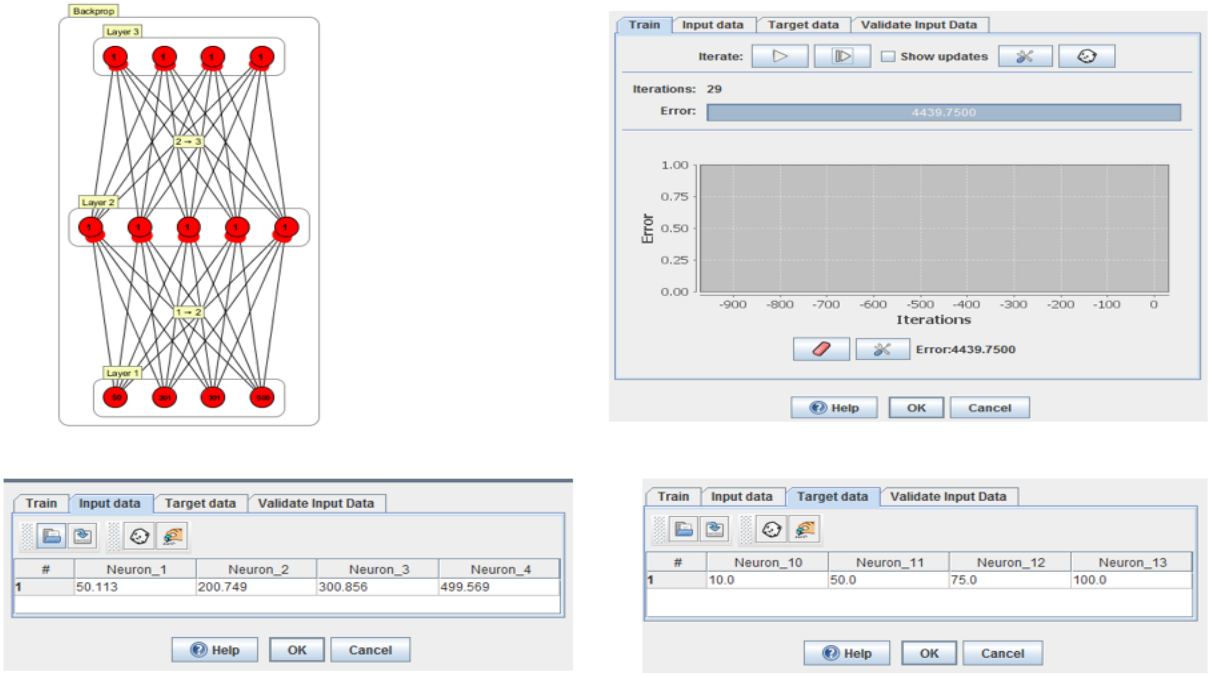


Figure 5: VPN network trained in Simubrain.

The outcome of this approach is:

- This technique is specifically constructed for dynamic clouds and is a pro-active fault tolerance technique.
- The approach uses ANN to estimate arrival time of the interrupt.
- The detection time obtained in this algorithm is independent of the previous interrupt.
- The failure detector is adaptive and efficient.

6. Conclusion

Cloud computing is a recent addition to the computing world. It offers various services at a lower price and flexible structure. The biggest challenge in cloud computing is fault tolerance. This is caused by its dynamic nature. To achieve complete reliability and robustness, artificial neural networks are used to calculate the earliest response and minimize the error in it. First the fault is detected and response time analyzed using cloudAnalyst and using these values BPN ANN is trained and error factor determined. By this method, the error factor is less and response time of the node can be determined.

7. References

1. Foster, I., Zhao, Y., Raicu, I., & Lu, S. (2008, November). Cloud computing and grid computing 360-degree compared. In *Grid Computing Environments Workshop, 2008. GCE'08* (pp. 1-10). Ieee.

2. L. M. Vaquero, L. R. Merino, J. Caceres, M. Lindner, "A break in the clouds: towards a cloud definition", *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 1, 2009.
3. Powell D, editor. Delta-4: a generic architecture for dependable distributed computing. Springer Science & Business Media; 2012 Dec 6.
4. Furht B. Cloud computing fundamentals. InHandbook of cloud computing 2010 (pp. 3-19).Springer US.
5. P. Watson, P. Lord, F. Gibson, P. Periorellis, G. Pitsilis, "Cloud computing for e-science with carmen" in , pp. 1-5, 2008, IBERGRID.
6. L. E. Moser, P. M. Melliar-Smith, D. A. Agarwal, R. K. Budhia, C. A. Lingley-Papadopoulos, "Totem: A fault-tolerant multicast group communication system", Communications of the ACM, vol. 39, no. 4, pp. 54-63, April 1996.
7. Lau KK, Tran CM. Server-side exception handling by composite Web Services. InEmerging Web Services Technology Volume III 2010 (pp. 37-54). Birkhäuser Basel.
8. Bala A, Chana I. Fault tolerance-challenges, techniques and implementation in cloud computing. IJCSI International Journal of Computer Science Issues. 2012 Jan;9(1):1694-0814.
9. Hayashibara, N., Defago, X., Yared, R., & Katayama, T. (2004, October). The φ accrual failure detector.In Reliable Distributed Systems, 2004. Proceedings of the 23rd IEEE International Symposium on (pp. 66-78). IEEE.
10. Gupta, I., Chandra, T. D., &Goldszmidt, G. S. (2001, August). On scalable and efficient distributed failure detectors. In *Proceedings of the twentieth annual ACM symposium on Principles of distributed computing* (pp. 170-179). ACM
11. Maier, G., Sommer, R., Dreger, H., Feldmann, A., Paxson, V., & Schneider, F. (2008, August). Enriching network security analysis with time travel.In *ACM SIGCOMM Computer Communication Review* (Vol. 38, No. 4, pp. 183-194).ACM.
12. Bahl, P., Chandra, R., Greenberg, A., Kandula, S., Maltz, D. A., & Zhang, M. (2007, August). Towards highly reliable enterprise network services via inference of multi-level dependencies.In *ACM SIGCOMM Computer Communication Review* (Vol. 37, No. 4, pp. 13-24).ACM.
14. G. Santos, L. Lung, C. Montez, "FTWeb: A fault-tolerant infrastructure for Web Services", Proceedings of the IEEE International Enterprise Computing Conference, pp. 95-105, September 2005.