International Journal of Modern Electronics and Communication Engineering (IJMECE) ISSN: 2321-2152
Volume No.-7, Issue No.-2, March, 2019

Enhancement of Fault Tolerance during Load Balancing in Cloud Computing

Bikash Chandra Pattanaik1st
CSE Department, GIET College
Name of organization
Khorda, India
E-mail: bikashpatnaik73@gmail.com

Kabrambam Rupabanta Singh 2nd
CSE Department, DRIEMS College
Cuttack, India
E-mail: rupabanta2008@gmail.com

Chandralekha 3rd
CSE Department, DRIEMS College
Cuttack, India
E-mail: chandralekha.sonu@gmail.com

Abstract: Cloud computing is built upon the advancement of virtualization and distributed computing to support cost-efficient usage of computing resources and to provide on demand services. The primary aim of cloud computing is to provide efficient access to remote and geographically distributed resources by maintaining the property of reliability. However the processing on remote computers creates more chances of occurrence of errors .So there is a need of creating fault tolerant system to maintain the reliability of the remote commuters. This paper focuses on maintenance load balancing and enhancing the fault tolerance of the system by integrating the reactive and proactive approaches of fault management. Our proposed system has been simulated by using Cloudsim plus and the simulation result shows that the reliability and fault tolerance has enhanced.

Keywords: Cloud computing, fault tolerance, load balancing, reliability, fault rate

I. INTRODUCTION

Cloud computing paradigm has been widely adopted due to the increasing demand for flexibility and scalability in dynamically obtaining and releasing computing resources in a cost-effective and device-dependent manner. It helps in hosting applications without the burden of installation and maintenance. The increasing popularity of this paradigm has increased the importance of its correct and continuous operation even in the presence of faulty components which is an attractive alternative to classic information processing systems. Among the many addressable issues of cloud computing such as fault tolerance, workflow scheduling, security, load balancing is one of the important issue which the researchers are supposed to look forward.

Load balancing is the process of distributing workloads and computing resources to improve the performance of the system. It allows the users to manage the demands of application or workload by allocating resources among multiple computers, networks or servers. Load balancing concerns the distribution of resources among the users or requests in uniform manner so that no node is overloaded or sitting idle. Like in, all other internet based distributed computing tasks, load balancing is an important aspect in cloud computing. The main goal of load balancing in cloud computing is to achieve maximum resource utilization, maximize throughput, maximize fault tolerance, minimize response time and avoid overload.

In traditional distributed computing, parallel computing and grid computing environments load balancing algorithms are categorized as static, dynamic or mixed scheduling algorithms based on their nature [1] where(a)Static Load Balancing Algorithm is suitable for small distributed environments with high Internet speed and ignorable communication delays(b)Dynamic Load Balancing Algorithm focuses on reducing communication delays and execution time and thus are suitable for large distributed environments(c)Mixed Load Balancing Algorithm focuses on symmetrical distribution of assigned computing task and reducing communication cost of distributed computing nodes. Based on above categorization, cloud computing clearly falls under the second category which means balancing load in cloud computing environment requires focusing on dynamic load balancing algorithms. Thus cloud computing environment requires a load balancing algorithm which could cater to dynamic service demands of users while providing optimized load balancing. The advantage of using dynamic load balancing is that if any node fails, it will not halt the system; it will only affect the system performance. The efficiency of load balancing [2]algorithm depends on performance of the parameters like reliability, resource utilization fault tolerance, adaptatibility, throughput ,migration time, response time, overhead, scalability etc.

The remainder of the paper is organized as follows. The section II represents the related work done in the proposed research area. Section III describes the proposed model. Implementation details and results have been defined in section IV .Finally concluding remarks and future work has been given in section V.

II. RELATED WORK

In [3], proposed a load balancing algorithm which balances the load on the host as well as preserves the fault tolerance level of the system based on virtual migration.[4]Based on VM migration, Wilcox proposed a load balancing scheme named modified central scheduler load balancing (MCSLB), which migrates VMs from the heaviest load host to the lightest host. In [5]Wang et al., proposed a scheduling algorithm to utilize better executing efficiency and maintain the load balancing of the system by combining opportunistic load balancing and load balance min–min scheduling algorithms. Zhang et al., in [6] presented a load balancing mechanism based on ant colony and complex network theory in the open cloud computing federation to realize load balancing in the distributed system. In [7] , to achieve the best load balancing and reduce or avoid dynamic migration, Hu et al., proposed a scheduling strategy on load balancing of VM resources based on a genetic algorithm with the consideration of system variation and historical data.[8] proposed an Ant Colony Optimization algorithm for load balancing in grid computing is proposed which will determine the best resource to be allocated to the jobs, based on resource capacity and at the same time balance the load of entire resources on grid. The main objective is to achieve high throughput and thus increase the performance in grid environment. A review on the load balancing studies for the cloud environment is presented in [9].

In [10], the author introduces an innovative perspective on adopting a fault tolerant mechanism that shades the implementation of cloud server with cloud selection to avoid network congestion and health monitoring for fault detection with migration technique to adaptively handle the occurrence of faults. The authors in [11] presented a proactive Fault Tolerance (FT) approach to HPC systems in the cloud to reduce the wall clock execution time in the presence of faults .In [12], the authors proposed a novel approach that optimally solves the problem of host overload detection by maximizing the mean intermigration time under the specified QoS goal based on a Markov chain model. [13] Implements an adaptive technique to achieve the required levels of usability, fault tolerance, and parallelism. This algorithm overcomes heterogeneity, provides high level of scalability, highly adaptable to dynamic environments and is highly fault tolerant. H. Liu et al. [14] proposed a load balancing strategy for virtual storage, here storage virtualization is achieved using three-layered architecture and load balancing is achieved using two load balancing modules. It helps in improving the efficiency of concurrent access by using replica balancing. This reduces the response time and enhancing the fault tolerance. [15]Hamid Arabnejad et al. proposed a fuzzy job distributor (load balancer) for fault tolerance management to reduce levels of management complexity for the user. It aims to reduce the possibility of fault occurrences in the system by a fair distribution of user job requests among available resources. In [16] an algorithm using Artificial Neural Network for fault detection has been proposed to overcome the gaps of previously implemented algorithms and to provide a fault tolerant model. A method of VM placement based on adaptive selection of fault-tolerant strategy for cloud applications is proposed in [17].The authors [18] stated the policies of the methods of creating the capacity of fault tolerance, detection and recovery in cloud computing . Soma Prathiba, and S. Sowvernica [19], discussed about failures, fault tolerant clustering methods and fault tolerant models that are specific for scientific workflow application. In [20] K. Devi and D. Paulraj proposed a multilevel fault tolerance system to tolerate the faults in real time cloud environment by providing higher reliability and availability.

III. PROPOSED SYSTEM

A. *Proposed modified Algorithm*

The work in this paper focuses on maintaining the fault tolerance level of the overall system while load balancing is considered. Our proposed work enables applications to respond promptly to traffic increase while also achieving greater fault tolerance. To balance the incoming load equally across the scaled up nodes dynamic load balancing algorithms are used which in turn reduces the fault. The load balancing comprise of two phases, scheduling at the VM initial placement stage and scheduling at the VM live migration stage. In initial placement stage of VMs, a modified round robin algorithm is proposed which allocates VM to the host in a cyclic manner but before allocating the VM it checks whether the same service type is already running in the host. The Modified Round Robin Algorithm maintains the fault tolerance level of the services at the initial allocation stage itself. After the initial allocation, the system is monitored at regular interval of time. The reactive and proactive fault tolerance [21] approaches have been integrated to enhance fault tolerance. The job migration method has been used to migrate virtual machine among hosts in the data centre, so that every host can run at the optimal load states .If the load of a host is too heavy, it will violate the SLA (service level agreement), which affects the Quality of Service (QoS) of the executing services. On the contrary, if the load of a host is too low, resources utilization of the host will be very low. Then the overhead due to these hosts is high. Therefore, the resource utilization thresholds of a host need to be monitored and recorded in the load balancing algorithm. The load of the host is measured in terms of its CPU usage. Based on CPU utilization the load of CPU can be diagnosed as overload, under load and average load.

Proposed Algorithm

1. Allocate VM to the host by round robin method
2. Monitor the system at regular interval and mark the host as overload, under load and Average based on the CPU usage
3. If overload host is found then go to step 4 else go to step 2
4. Sort the VM in descending order and select the VM to be migrated
5. Sort the underload marked host in ascending order
6. Find the suitable host among the sorted under load host. If any VM is already allocated or host has not enough resources for further allocation then no migration
7. If destination host is found then go to step 8 else go to step 9
8. Perform the VM migration
9. End

B. Fault Tolerance Model

The fault tolerance model can be defined as follows; Let the cloud environment have the redundant N+Y=K configuration having N operating hosts and Y processing hosts as spare hosts. Here if all of the N hosts fail, the system fails otherwise whenever one host is working, the system is still working. The following assumptions are made to formulate the model.

- The system consists of N operating hosts, Y spare hosts. When any operating host fails then it can be replaced by the available spare host.
- All the operating (spare) hosts have the same hardware failure rate λ_h (**α**) arising from an exponential distribution.
- Each host have only two states, working state and failed state.
- All the failures involved are mutually independent
- The system may also fail due to common cause with failure rate **λ_P**.
- When all standbys are exhausted, the remaining operating hosts fail with degraded failure rate **λ_d**

If ith component has exponential distributed life time with constant failure rate of λ_i, then reliability is

$$R_s(t) = e^{-\sum_{i=1}^{N} \lambda_i t}$$

1. The system will work till k of giving N is good. Then the total reliability is calculated as follows

$$r_{k\text{-out-of-}N}(t) = \sum_{i=k}^{N} \binom{N}{i} (R(t))(1 - R(t))^{N-1}$$

--------- (1)

System operates only if at least K-out - of – N components operates

2. When there are n failed operating hosts in the system, the state dependent fault rate is given by

$$\text{Fault Rate} = F\,n = N\lambda + (Y-n)\alpha \quad 0 \le n \le Y \quad \text{or}$$
$$F\,n = (N+Y-n)\,\lambda d \quad Y < n \le k-1$$

--------- (2)

The above two equations represent the performance metrics for our proposed model.

C. Measuring Fault Tolerance

The fault tolerance has to be done on the basis of the availability of spare hosts. The adjudicator node contains the time checker (TC), reliability calculator (RC) and decision mechanism algorithms to find the reliable VM to process the client request. The node with maximum reliability is selected as the system event output. To provide the fault tolerance the data can be stored on multiple cloud using virtualization techniques. Acceptance Test (AT) module checks whether the fault will occur or not. The AT can respond both success and failure case of the VM. If result is success then it passes result to TC module. If the result is failure, then AT sends a verify exception signal to TC. TC is defined to evaluate response time in milliseconds within specified time bound. If the VM can respond within the specified time limit, and that host is taken as reliable then passed to RC module. The RC module assesses the reliability for each VM. To measure the fault rate a fault injector module has been defined. As the proposed system tolerates the faults and makes the decision on the basis of the reliability of the processing nodes, the reliability of the VM is improved, which changes after every computing cycle. In the beginning the reliability of each VM is 100%. If a processing node manages to produce a correct result within the time limit, its reliability increases using a reliability Factor r, and if the processing node fails to produce the correct result within the time limit, its reliability decreases using adaptability factor n. The reliability assessment algorithm is more convergent towards failure conditions. The VM which responds above the time limit is considered as failure

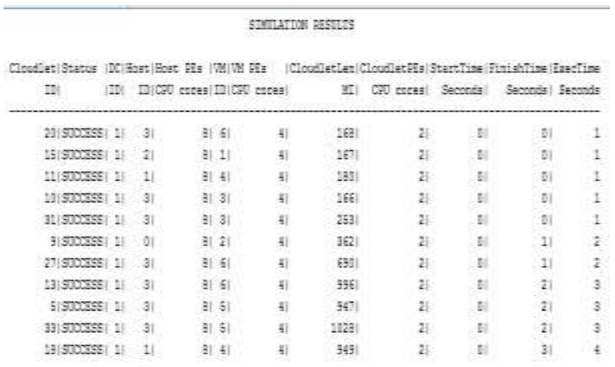
IV. IMPLEMENTATION

The Java based Cloudsim plus 3.0 toolkit has been used as simulation environment which is a full-featured, highly extensible simulation framework enabling modelling, simulation, and experimentation of Cloud computing infrastructures and application services. The main scenario considered one data enters situated in a given location that replies to requests from users from different locations. The data centre was built homogeneous in order to gain measurable information when machines fail due to the fault injector. Each machine in the data enter has the following configuration: x86 architecture, Linux operating system, 5GB of RAM, 100 GB of storage and 1GB available bandwidth. In our experiment we considered 7 virtual machines deployed. Each virtual machine comes with 1GB RAM and four processing element (PEs). The round robin policy of load balancing is applied in this work.

This scenario was run several times with different amount of failures and the results were represented in figure 1.The reliability graph is plotted in figure 2 which shows that initially the total reliability is increasing up to a maximum level**.** However if we increase the number of hosts then the reliability is decreasing as the cost of implementing the host is more and the overhead is more due to less resource utilization of CPU. The Graph represents the total reliability for taking different number of spare hosts .Further the graph indicates that the reliability is optimized when we provide some number of spare hosts to handle the faults while maintaining the load balancing. The faults that are created by the fault injector are physical host faults that cause a decrease in reliability but it can be improved by providing extra number of spare host and maintaining load balancing. This in turn minimizes the fault rate which is shown in figure 3.Figure 4 and figure 5 represents the number of overloaded hosts and number of migration as per increasing number of hosts depending upon fixed and variable number of cloudlets.

The simulation has been ran several times, each time increasing the number of faults by one, until reaching eight which is the maximum number of faults that can be generated in this scenarios without completely faulting the entire data centre. To measure the reliability and fault rate the following values are taken.

N = 5, Y = 2, n = 3, $\lambda = 0.09$, $\lambda_h = 0.06$, $\lambda_d = 0.4$, $\lambda_P = (0.1,.4,.6,.8)$, $\alpha = (0.3,.6, .9)$



```
                                SIMULATION RESULTS

Cloudlet|Status  |DC|Host|Host PEs |VM|VM PEs   |CloudletLen|CloudletPEs|StartTime|FinishTime|ExecTime
     ID|         |ID|  ID|CPU cores|ID|CPU cores|         MI|  CPU cores|  Seconds|   Seconds| Seconds
-------------------------------------------------------------------------------------------------------
      23|SUCCESS| 1|   3|        8| 6|        4|        168|          2|        0|         0|       1
      15|SUCCESS| 1|   2|        8| 1|        4|        167|          2|        0|         0|       1
      11|SUCCESS| 1|   1|        8| 4|        4|        183|          2|        0|         0|       1
      10|SUCCESS| 1|   3|        8| 3|        4|        166|          2|        0|         0|       1
      31|SUCCESS| 1|   3|        8| 3|        4|        253|          2|        0|         0|       1
       9|SUCCESS| 1|   0|        8| 2|        4|        362|          2|        0|         1|       2
      27|SUCCESS| 1|   3|        8| 6|        4|        693|          2|        0|         1|       2
      13|SUCCESS| 1|   3|        8| 6|        4|        396|          2|        0|         2|       3
       5|SUCCESS| 1|   3|        8| 5|        4|        347|          2|        0|         2|       3
      33|SUCCESS| 1|   3|        8| 5|        4|       1028|          2|        0|         2|       3
      19|SUCCESS| 1|   1|        8| 4|        4|        549|          2|        0|         3|       4
```

Figure1. Simulation Result

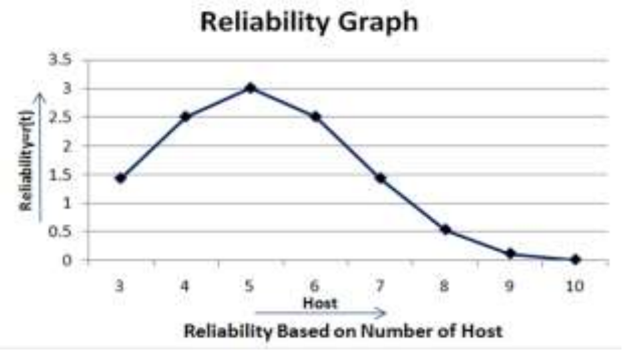


Figure2. Reliability Graph

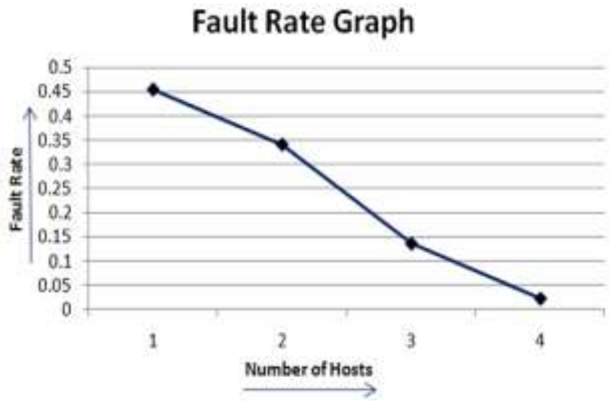


Figure3. Fault Rate Graph

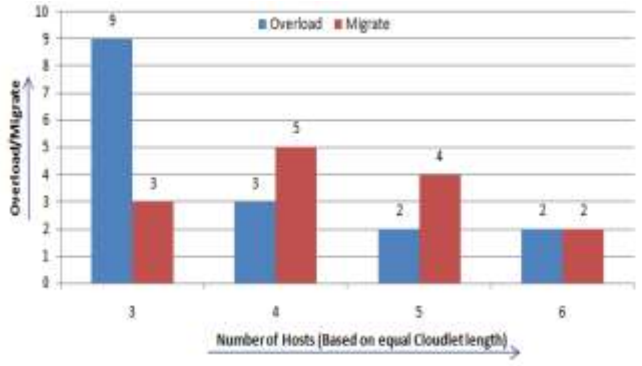


Figure4. Number of Overload/Migrate Hosts (Equal Cloudlet Length)

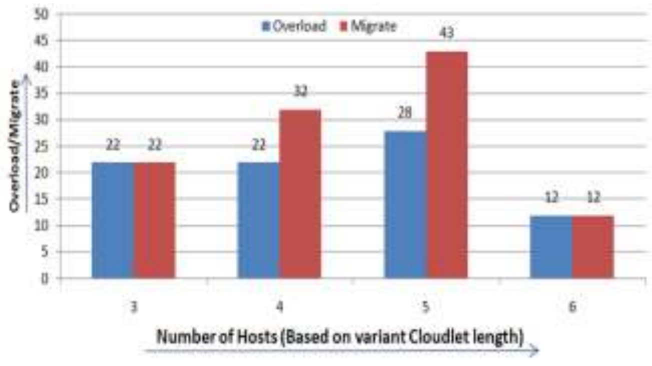


Figure5. Number of Overload/Migrate Hosts (Variant Cloudlet Length)

V. Conclusion

This paper gives an overview about fault management during load balancing in cloud environment. Our proposed model for fault management is dynamic in nature and considers the balancing of load to enhance the fault tolerance level by providing some spare hosts. The simulation result shows that the reliability can be more which in turn enhances the quality of service requirements of cloud users. As the fault tolerance is increased the fault rate is also decreasing .Our future work will be to consider the other Qos parameters such as availability, recoverability and warranty period factors for enhancing the fault management in cloud computing.

References

[1] M. Amar, K. Anurag, K. Rakesh, K. Rupesh and Y. Prashant "SLA Driven Load Balancing For Web Applications in Cloud Computing Environment", Information and Knowledge Management, vol.1, No.1, pp. 5-13,2011.

[2] L. M. Vaquero, L. Rodero-Merino, J. Caceres and M. Lindner, "A break in the clouds: towards a cloud definition," SIGCOMM ACM Computer Communication Review, vol. 39, pp. 50–55, December 2008.

[3] S. G. Gayatri, S. Latha, "Implementing a Fault Tolerance Enabled Load Balancing Algorithm in the Cloud Computing Environment", IJEDER, vol. 5, no. 1, 2017.

[4] S. Wang, K. Yan, W. Liao, and S. Wang, "Towards a Load Balancing in a Three-level Cloud Computing network", 3rd IEEE Int. Conf. Computer Science and Information Technology(ICCSIT), Chengdu, China, pp. 108–113, July 9–11, 2010.

[5] T.C. Wilcox Jr , "Dynamic load balancing of Virtual machines hosted on Xen.", Master's Thesis, Brigham Young University, USA, 2009.

[6] Z. Zhang and X. Zhang, "A Load Balancing Mechanism Based on Ant Colony and complex Network Theory in Open Cloud Computing Federation", 2nd Int. Conf. Industrial Mechatronics and Automation (ICIMA), Wuhan, China, May 30–31, pp. 240–243, 2010.

[7] J.Hu, J. Gu, G. Sun, and T. Zhao , "A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment" 3rd Int. Symp. Parallel Architectures, Algorithms and Programming (PAAP), Dalian, China, December 18–20, pp. 89–96, 2010.

[8] S. Suryadevera, J. Chourasia, S. Rathore, and A. Jhummarwala, "Load balancing in computational grids using ant colony optimization algorithm," International Journal of Computer & Communication Technology, vol. 3, no. 3, 2012.

[9] N. J. Kansal, "Cloud load balancing techniques: a step towards green computing", IJCSI International Journal of Computer Science, vol. 9, no. 1, 2012.

[10] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, "A novel ultrathin elevated channel low-temperature poly-Si TFT," IEEE Electron Device Lett., vol. 20, pp. 569–571, Nov. 1999.

[11] I.P. Egwutuoha, S. Chen, D. Levy, B. Selic and R.. Calvo, "A proactive fault tolerance approach to HPC in the cloud.", International Conference on Cloud Computing, pp. 268–273, 2012.

[12] A. Beloglazov, R. Buyya, "Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints", IEEE Trans. Parallel Distributed. System, vol. 24, no. 7, pp. 1366–1379, 2013.

[13] K. Maheshwari, M.. Lim, L. Wang, K. Birman, R. van Renesse, "Toward a reliable, secure and fault tolerant smart grid state estimation in the cloud.", In Innovative Smart Grid Technologies (ISGT), pp. 1–6, 2013.

[14] H. Liu, S. Liu, X. Meng, C. Yang, Y. Zhang, LBVS: A Load Balancing Strategy for Virtual Storage" International Conference on Service Sciences (ICSS), pp 257-262, 2010.

[15] Hamid Arabnejad, Claus Pahl , Giovani Estrada, Areeg Samir, Frank Fowley," A Fuzzy Load Balancer for Adaptive Fault Tolerance Management in Cloud Platforms",ESOCC 2017, pp 109-124, 2017.

[16] Amin, Z., Singh, H., Sethi, N.: Review on fault tolerance techniques in cloud computing. Int. J. Comput. Appl. 116(18), 11–17, 2015

[17] Chen, X., Jiang, J.H.: A method of virtual machine placement for fault-tolerant cloud applications. Intel. Autom. Soft Comput. 22(4), 587–597, 2016

[18] Cheraghlou, M.N., Khadem-Zadeh, A., Haghparast, M.: A survey of fault tolerance architecture in cloud computing. J. Netw. Comput. Appl. 61, 81–92, 2016.

[19] S. Prathiba, S. Sowvernica, "Survey of Failures and Fault Tolerance in Cloud", ICCCT,IEEE 2017

[20] K. Devi, D. Paulraj,"Multilevel fault tolerance in Cloud Environment",ICICCS ,IEEE,2017.

[21] P. Kumari, P. Kaur,"A Survey of Fault Tolerance in Cloud Computing",JKSU-Computer and Information Science, 2017.