

LBVFT: A Load Balancing Technique for Virtualization and Fault Tolerance in Cloud Computing

Pranesh Das

Department of Computer Science and Engineering
National Institute of Technology, Rourkela
Orissa, India, 769008

Pabitra Mohan Khilar

Department of Computer Science and Engineering
National Institute of Technology, Rourkela
Orissa, India, 769008

ABSTRACT

Load balancing in cloud computing is a grand challenge problem now a days. The main load balancing issues in cloud computing is load calculation and load distribution. To solve these issues, many load balancing techniques have been designed to distribute tasks properly. In this paper, we have proposed a Load Balancing Technique for Virtualization and Fault Tolerance in Cloud Computing (LBVFT) to assign the tasks to the virtual nodes. A Cloud Manager (CM) module and a Decision Maker (DM) are used in the proposed scheme to manage the virtualization, load balancing and to handle the faults. LBVFT is mainly designed to assign tasks to the virtual nodes depending on the success rates (SR) and the previous load history. In the load assigning technique assignment is done by the load balancer (LB) of cloud manager (CM) module in the basis of higher success rate and lower load of the available nodes.

General Terms

Cloud computing, Load balancing, Virtualization, Fault tolerance.

Keywords

LBVFT, VFT, Cloud Manager, Decision Maker, Fault handler, Success Rate, Load.

1. INTRODUCTION

An issue in distributed scheduling is load balancing [2] , [3] ,[4] which tries to distribute the tasks to be executed among the resources of the system. Load balancing can be achieved either locally or in a distributed fashion. Distributing tasks across a communication medium is sometimes referred to as the resource allocation problem. Resource allocation actually refers to scheduling multiple resources. In this paper gives a load assigning technique for the VFT model [1] which is given in figure 1. The proposed load assigning scheme for the VFT model [1] assigns a task to the available virtual nodes depending on their success rates and the load history. Because of the very large infrastructure of cloud and the increasing demand of services an effective fault tolerant technique for cloud computing is required and for which an effective load balancing approach is required. In the VFT model the load balancer takes high responsibility by distributing loads only to those virtual nodes whose corresponding physical servers have a good performance history.

2. LITERATURE REVIEW

A lot of work has been done in the area of load balancing and fault tolerance for cloud computing. But due to its virtualization and internet based service providing behavior load balancing and fault tolerance in cloud computing are still a big challenge. Many researchers have given various load balancing techniques and strategies in [2], [3], [4], [5], [6] and [8]. The proposed technique distributes loads in a smart way by considering the success rates and the loads history of the available virtual nodes. Thus the LBVFT helps the VFT model [1] to tolerate not only faults but also reduce the chance of future faults by not assigning tasks to virtual nodes of physical servers whose success rates are very low and loads are very high.

3. PRELIMINARIES

This research work focus on assigning tasks to those virtual nodes which are having higher success rate (SR) and lower load history to provide high availability in cloud environment by using a virtualization and fault tolerance approach for cloud data centers. This paper presents an approach to provide high availability by the VFT model [1] to the requests of cloud's clients. To achieve this, a virtualization scheme with reactive fault tolerance for cloud computing is proposed in [1]. According to the scheme load balancer (LB) searches for the available virtual nodes having good SR value and lower load history by various searching algorithms like Binary Search, Linear Search and Randomized Searching Algorithm (RSA) [7] etc. LB then distributes loads to the desired virtual nodes.

4. PROPOSED ALGORITHM

4.1 Description

Here in the proposed load balancing approach when cloud manager (CM) of VFT model gets the request from the cloud service provider (CSP) it creates set of virtual nodes with the help of system hardware, host operating system and hypervisor module. System hardware means set of physical servers which are connected by distributed network .CM then gives the responsibility to the load balancer to assign the tasks. As in the VFT model to achieve fault tolerance redundant virtual nodes are used so, the LB will assign the same task to a number of available virtual nodes which are having good SR value and lower load in the performance record table. Thus the same task will be executed in all the selected nodes and the result will be sent to the Decision Maker (DM) module of the VFT model (figure 1).

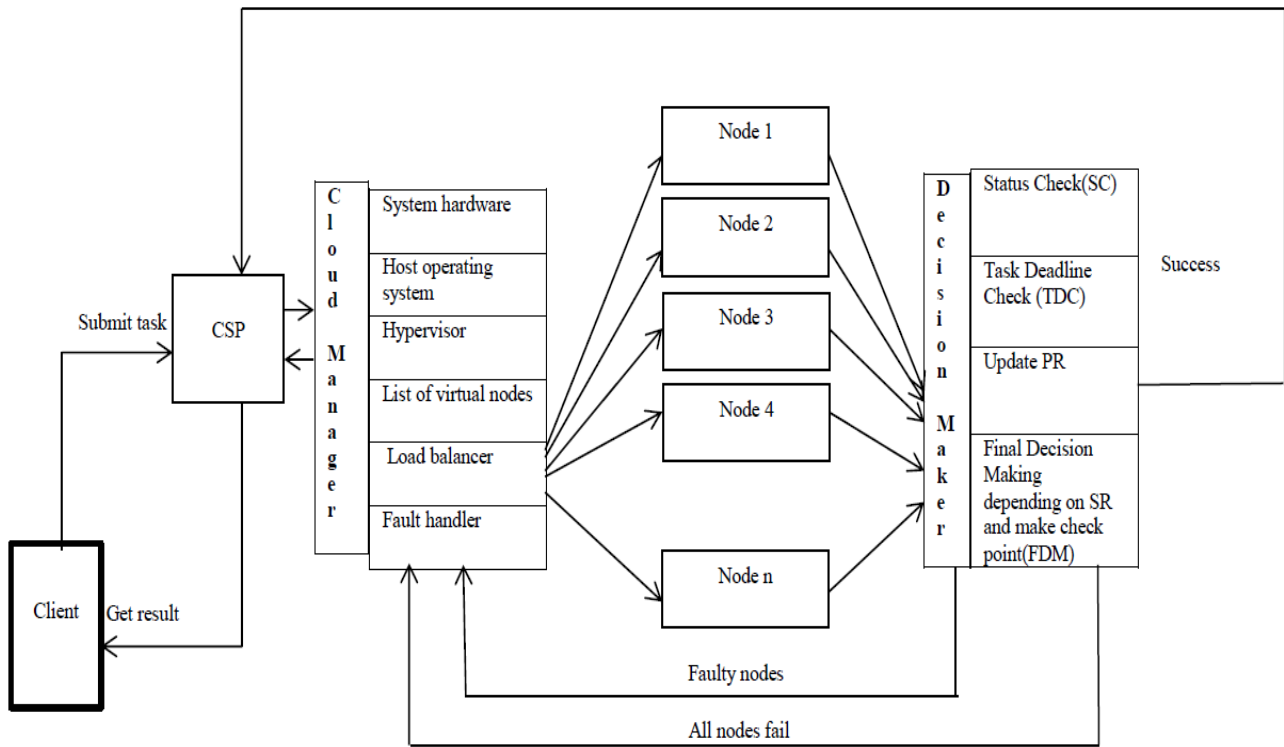


Figure 1: The VFT Model

4.2 Working of the LBVFT algorithm

1. Initially success rate (SR)=0.5,maximum SR=1, $0 < SR \leq 1$.
2. Input high SR, higher SR, low SR, lower SR, low load, lower load, high load and higher load values.
3. $SR = n1/n2$
4. n1 is the number of times the virtual node of a particular physical server gives successful results.
5. n2 is the number of times the Load Balancer of the cloud manager(CM) assigns tasks to a particular server's virtual node.
6. Loads of the virtual nodes are given. Loads are generally calculated by the load balancer and is updated and kept in the performance record table of the virtual nodes time to time, which is beyond the scope of this paper.
7. Search for all the available virtual nodes having lower load history and higher SR values in the performance record table.
8. if((SR==higher || SR==high) &&(load==lower))
{
 Select the node

9. else
{
 if((SR==higher|| SR==high)&&(load==low))
 {
 Select the node
 }
}
10. if((SR==Higher|| SR==High)&&(load==higher))
{
 Don't select the node if enough nodes are available
}
11. if((SR==Higher|| SR==High)&&(load==high))
{
 Don't select the node if enough nodes are available
}
12. if ((SR==low||SR==lower)&&(load==low))

```

{
    Select the node if numbers of nodes are less
}
13. if ((SR==low||SR==lower)&&(load==lower))
{
    Select the node if numbers of nodes are less
}
14. if((SR==low||SR==lower)&&(load==high))
{
    Don't select the node
}
15. if((SR==low||SR==lower)&&(load==higher))
{
    Don't select the node
}
16. Select m nodes from the list of available virtual
nodes (m is the user input; m-1 is the number of
redundant nodes).
17. Submit the task to all the selected m nodes.
18. The tasks are then executed by the allocated virtual
nodes and the results are sent to the decision maker
(DM) module of the VFT model.
    
```

4.3 Different Scenarios

4.3.1 High or higher success rate and low or lower load

In this case a node is better one for selection and the node is selected. As the load is low and the SR value is high, there is a less chance of future fault.

4.3.2 High or higher success rate and high or higher load

If a node having this property is selected then there is a chance of future fault as the load is high. So, this kind of nodes is avoided for selection if sufficient numbers of virtual nodes are available.

4.3.3 Low or lower success rate and high or higher load

A node having this feature is not selected, because the SR value is low as well as load is high.

4.3.4 Low or lower success rate and low or lower load

In this case a node may be selected if sufficient numbers of nodes are not available.

Table 1: Simulation Result Obtained from the VFT model [1]

Cycle	Task Dead line	Virtual Node 1					Virtual Node 2					Virtual Node 3					Sel ect ed No de
		S C	T D C	Finish Time	SR	Load	S C	T D C	Finish Time	SR	Load	S C	T D C	Finish Time	SR	Load	
Start	-	-	-	-	0.5	L1	-	-	-	0.5	L2	-	-	-	0.5	L3	-
1	1700	1	1	1600.0	0.667	L4	1	1	1601.6	0.667	L5	1	1	1610	0.667	L6	1
2	1602	1	1	1600.4	0.75	L7	1	1	1602.0	0.75	L8	1	0	1610.4	0.5	L9	2
3	1601	1	1	1600.8	0.8	L10	1	0	1602.4	0.6	L11	1	0	1610.8	0.4	L12	1
4	1605	1	1	1601.2	0.833	L13	1	1	1602.8	0.667	L14	1	0	1611.2	0.333	L15	1
5	1600	1	0	1602.4	0.714	L16	1	0	1603.2	0.571	L17	1	0	1611.6	0.286	L18	-
6	1900	1	1	1602.0	0.75	L19	1	1	1603.6	0.625	L20	1	1	1612.0	0.375	L21	1
7	1700	1	1	1602.4	0.778	L22	1	1	1604.0	0.667	L23	1	1	1612.4	0.444	L24	1
8	2100	1	1	1603.6	0.8	L25	1	1	1604.4	0.7	L26	1	1	1612.8	0.5	L27	1
9	1700	1	1	1603.2	0.818	L28	0	0	-	0.636	L29	1	1	1613.2	0.545	L30	1
10	2000	1	1	1603.6	0.833	L31	1	1	1605.2	0.666	L32	1	1	1613.6	0.583	L33	1

Here on the above table “1” denotes success and “0” denotes fail in SC and TDC column.

Table 2: Constructed from the above simulated results

Tasks	Vm1			Vm2			Vm3			Select ed Nodes
		SR	Load		SR	Load		SR	Load	
Start		0.5	L1		0.5	L2		0.5	L3	-
t1		0.667	L4		0.667	L5		0.667	L6	1,2
t2		0.75	L7		0.75	L8		0.5	L9	1,2
t3		0.8	L10		0.6	L11		0.4	L12	1,2
t4		0.833	L13		0.667	L14		0.333	L15	1,3
t5		0.714	L16		0.571	L17		0.286	L18	1,2
t6		0.75	L19		0.625	L20		0.375	L21	1,2
t7		0.778	L22		0.667	L23		0.444	L24	1,2
t8		0.8	L25		0.7	L26		0.5	L27	2,3
t9		0.818	L28		0.636	L29		0.545	L30	1,2
t10		0.833	L31		0.666	L32		0.583	L33	1,2

Here on the above table L1, L2, L3.....L33 are the loads of the virtual nodes at that instant.

5. SIMULATION RESULTS AND DISCUSSION

Simulation is done in Cloudsim2.0 [9], [10] with NetBeans IDE 6.7.1. The simulation result that is obtained from VFT model [1] is given in Table 1. In the resultant table L1, L2....L33 are the loads of the virtual nodes at that instant of time. From Table 1 another table Table 2 is constructed. It is assumed that there are 3 virtual nodes available and their SR values and loads are taken. Here it is considered that there are 10 tasks and for every task 3

virtual nodes are there. Proposed load assigning technique will assign tasks to those nodes which have good SR and load values. It is assumed that number of redundant node is 1. Hence for every task 2 virtual nodes will be selected by the proposed technique which is shown in Table 2.

6. CONCLUSION AND FUTURE WORK

This paper gives a smart load distribution strategy for our virtualization and fault tolerance in cloud computing (VFT) model using success rate of the computing nodes and previous load history. In Table 2 it is seen that maximum time node 1 and 2 is selected. For task t8 and task t4 although node 1 and node 2 has high SR values still are not selected because it is assumed that loads for those nodes are very high. This task assigning technique helps the VFT model to give a good performance. As only higher SR values and lower loads are considered during virtual node selection hence, there is a very less chance of system failure. Our future work is to provide an efficient load balancing, load migration, load calculation and fault handling technique to make the VFT model more effective.

7. REFERENCES

- [1] Pranesh Das, Dr.Pabitra Mohan Khilar, “VFT: A Virtualization and Fault Tolerance Approach for Cloud Computing”, IEEE Conference on Information and Communication Technologies (ICT 2013), Kanyakumari, Tamil Nadu, in press, 2013.
- [2] Neeraj Rathore, Dr. Inderveer Chana ,“A Cognitive Analysis of Load Balancing and job migration Technique in Grid”, World Congress on Information and Communication Technologies (WICT), 2011,P 77-82, ISBN 978-1-4673-0127-5.
- [3] A.K, M.Z, EL Bakkali, Dafir EL Kettani,“Load Balancing Cloud Computing : State of Art”, ISBN 978-1-4673-1053-6 , IEEE 2012.
- [4] Jinhua Hu ,Jianhua Gu, Guofei Sun, Tianhai Zhao, “A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment”, 3rd International Symposium on Parallel Architectures, Algorithms and Programming, 978-0-7695-4312-3/10, 2010 IEEE,DOI 10.1109/PAAP.2010.65.
- [5] Xiaona Ren , Rongheng Lin, Hua Zou, “ A Dynamic Load Balancing Strategy for Cloud Computing Platform Based on Exponential Smoothing Forecast”, Proceedings of IEEE CCIS2011, 978-1-61284-204-2/11, 2011 IEEE.
- [6] M. Randles, D. Lamb, and A. Taleb-Bendiab,“A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing,” 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops, 2010, pp. 551–556.
- [7] Pranesh Das, Dr. Pabitra Mohan Khilar “A Randomized Searching Algorithm and its Performance analysis with Binary Search and Linear Search Algorithms”, Volume 1, No. 11, January 2013 ISSN – 2278-1080.
- [8] Derek L. Eager, Edward D. Lazowska, Jhon Zahorjan, “Adaptive Load Sharing in Homogeneous Distributed Systems”, IEEE Transactions on Software Engineering, Vol. SE-12,No 5,May 1986.
- [9] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César A. F. De Rose, and Rajkumar Buyya, “CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms”, Software: Practice 2011 - Wiley Online Library.
- [10] Rajkumar Buyya, Rajiv Ranjan and Rodrigo N. Calheiros,” Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities”, International Conference on High Performance Computing & Simulation, 2009. HPSC '09,P-1-11.