International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Load Balancing By Max-Min Algorithm in Private Cloud Environment

S M S Suntharam

Department of Cloud Computing, SRM University, Chennai, India

Abstract: ***Cloud computing is a latest emerging technology because of its high availability, high performance, low cost and pay for use model. Wherein IT infrastructure and applications are provided as services to end-users. It enables On-Demand services where resources can be commissioning and decommissioning according to user needs. Private cloud, public cloud, hybrid cloud are the three main deployment model of cloud computing. Private cloud that can be building within the organization and the data security is more when compared to the other model clouds. In which storage nodes that can be increased when there will be an increase in the storage demand. During such increase, storage nodes in the private cloud have to be balanced with load in order to avoid the traffic in private cloud. Cloudsim is a simulation toolkit which ensures simulating, modeling and experimenting on cloud computing Design. It supports and to model the behavior component of cloud system such as data centers, virtual machines, hosts, service brokers, Scheduling and allocation polices. The idea behind this paper is to use max-min algorithm in cloudsim to show how to balance the load across the different storage nodes in the private cloud, which reduce the make span and data traffic. Max-Min algorithm is also used to reduce idle time and so efficient in mapping the load across the nodes.***

Keywords: Storage nodes, Virtual machine, Cloudsim, Max-Min algorithm.

1. Introduction

Renting the computing resources like hardware and software as service through internet is called cloud computing. Virtualization is the Key enabling technique and backbone of Cloud Computing. It helps to abstract the software from underlying hardware. Virtualization that can be applied to all computing resources named such as storage virtualization, memory virtualization, network virtualization and computing virtualization. Cloud computing provides the shared pool of resources to the end-user. All the resources that can be efficiently utilized by the cloud according to the customer requirements by the concept of elasticity and scalability. The service provided by the cloud computing are Infrastructure as a service(IAAS),Platform as a service(PAAS) and Software as a service(SAAS).Private cloud that can provisioned for single organization and it can be accessed anywhere within that organization Public cloud is provisioned for open use by all customer and it is maintained by third party. Hybrid cloud is a combination of private cloud and public cloud. Even though the public cloud can be accessed from anywhere at any time, but it has certain significant security related risks due to data remnants, unencrypted data and shared multi-tenant environments.

Private cloud storage is controlled by an organization. It is built by commodity machines within the organization where various users store and access their data. There may be increasing storage due to the including of new customers or abundant storage of data by exciting customers when they cross beyond their limits. Due to this storage nodes automatically get increased according to scalability property of cloud. During such expansion of the storage nodes load should be maintain across the nodes to avoid the traffic and load imbalance

Load balancing algorithm that can be implemented in this paper which attempts to balance the load when load is storing on nodes.

The rest of this paper is organized as follows. Section II discusses the related works. Section III describes the Problem Description. Section IV discusses the proposed system architecture. Section V describes the proposed algorithm. Section VI describes the Analysis and Report. Section VII Concludes and provides Future work.

2. Related Works

Martin Randles [1] proposed the optimization of network topology in which he used clustering and the honeybee foraging algorithm at the top of application layer. For making optimum resources allocation he used simulation of a self organizing and beehive-based load-balancing algorithm at the top of application layer. The work examined the allocation of servers in a large-scale SOA, he also proposed load balancing Biased Random Sampling approach. Connectivity in a virtual graph is used to represent the load.

Klaithem Al Nuaimi [2] proposed the most known contributions in the literature for load balancing in cloud computing. He classified the load balancing algorithms into two types: static algorithms and dynamic algorithms. Static load-balancing algorithms that have been developed for cloud computing. Based on the ability of the node to process new requests and tasks are assigned. The process is based on prior knowledge of the nodes properties and capabilities. These would include the node's processing power, memory and storage capacity, and most recent known communication performance.

Haozheng Ren [3] proposed a dynamic migration algorithm in cloud computing environment under considering the heterogeneity of environmental resources on cloud computing applications. He proposed fractal-based load balancing trigger strategy. Traditional trigger strategies are based on specific threshold value. Virtual machine migration strategy is called as Trigger strategy. It determines the

migration timing judged by the overloaded node in the system.

Yupeng Zhang and Wing Shing Wong [4] proposed a model which is to presented to study the distributed load balancing problem. There have three basic assumption: first one is users independently distribute their tasks to servers according to pre-assigned binary sequences .Second is no real-time synchronization is required, this is no centralized controller after the initial sequences assignment.

Chun-Wei Tsai, Wei-Cheng Huang [5] proposed a high-performance hyper-heuristic algorithm to find better scheduling solutions for cloud computing systems, their algorithm has two detection operators that automatically determine when to change the low-level heuristic algorithm and a perturbation operator to fine-tune the solutions obtained by each low-level algorithm to further improve the scheduling results in terms of makespan.

Z.Zhang and Xu. Zhang [6], Proposed a load balancing algorithm in which ant colony concepts and complex network theory has described in cloud computing and improved many related ant colony algorithm that balance the load in cloud computing system.

H. Mehta, P. Kanungo [7], proposed workload and client aware policy.it is a method in which hybrid approach of client that can be described with the workload that can be used as a request distribution policy.

Y. Lua, Q. Xiea, G. Kliotb, A. Gellerb, J. R. Larusb, and A. Greenber [8], proposed algorithm named as Join- Idle -queue for balancing load in large system. No communications that can be occur between dispatchers and processors when the job arrived at node.

X. Liu, Pan, C. Wang, and J. Xie,[9], proposed a lock-free multiprocessing load balancing solution which eliminated the use of shared memory with other multiprocessing load balancers in which lock concepts that can be used to maintain the user section.

H. Liu, S. Liu, X. Meng, C. Yang, and Y. Zhang [10], proposed load balancing virtual storage strategy which offers storage as service on cloud storage and also large data storage model which proposed a two-level task scheduling to meet load balancing criteria.

S. Wang, K.Yan , W. Liao and S. Wang [11], proposed algorithm which combines Opportunistic Load Balancing and Load Balance Min-Min scheduling algorithms which can offers a good efficiency and maintain the load balancing in the system.

A. Bhadani, and S. Chaudhary [12], proposed Central Load Balancing Policy for Virtual Machines for load balancing in the virtualization and cloud computing models.

Z. Zhang, H. Wang, L. Xiao and L. Ruan [13], proposed an algorithm named as Statistic based Load Balance which has used for online resource allocation decision with the help of statically prediction mechanism.

Bhathiya [14], proposed two virtual machine load balancing algorithms in which first algorithm is Active Monitoring Load Balancing algorithm, which distributes the load equally to all virtual machines and second algorithm is throttled load balancing algorithm which allows the allocation of task for each VM, with help of this better load balancing algorithm can be achieved to all VM.

M. Randles, D. Lamb and A. Bendiab [15], proposed three load balancing algorithm which are large-scale complex system, biased random sampling on walk procedure and Active Clustering. Dynamic network system is created which provided the load distribution status in biased random sampling. Active Clustering is used for rewire the network. In these applications all the nodes are aware of other nodes which can be easily help for load balancing.

3. Problem Description

In a Load balancing problem, where jobs are to be assigned immediately to the resources in heterogeneous environment. The allocation should be as fast as possible, while at the same time optimizing several criteria such as utilization and response time can also be consider. There are no dependencies between the different tasks. The arrival rate of jobs determines the system load, and there may be a different runtimes for a particular task on different machines. Max-Min is a simplest scheduling algorithm which is proposed to balance the load in the private cloud storage. In which mean completion time of all virtual machines storage nodes in particular private cloud can be calculated. Then the machine whose completion time is greater than the mean value is selected and those jobs will be rescheduled to the node or machine which has least completion time.

4. Proposed System Architecture

A private cloud is built with numerous virtual machines several storage nodes are reside inside the VM. Private cloud storage which contains Walrus controller, Storage controller, Load balancer, and Virtual Machines with Storage node. Walrus controller offers persistent storage to all of the virtual machines in the private cloud and can be used as a simple HTTP put/get storage as a service solution. There are no data type restrictions for Walrus, and it can contain images i.e., the building blocks used to launch virtual machines, volume snapshots and application data. Only one Walrus can exist per cloud storage. The storage controller communicates with the virtual machine and manages block volumes and snapshots to the instances within its specific cluster. If an instance requires writing persistent data to memory outside of the cluster, it would need to write to Walrus, which is available to any instance in any cluster. Load balancer which is used to balance the load with the help of Max-Min algorithm. During the start-up of the Load balancer, storage nodes register their details such as free space, network utilization, utilized space and IP address .Storage controller which forwards the load to the load balancer for distributing them across the various storage nodes in the cluster to avoid load imbalance

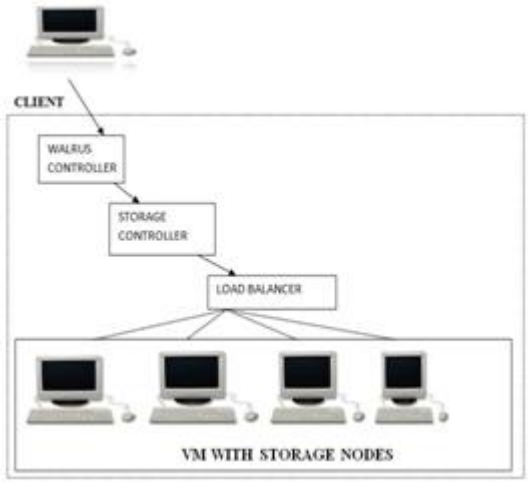

Figure 1

5. Proposed Algorithm

A. Storage Node Sub Module

The storage node sub module gets the status of the storage nodes. It contains RAM, bandwidth, capacity, utilized space, residual space, network utilization, current load of each storage node. Then it receives the free space, network utilization, utilized space and IP address from the load balancer.

B. Path Transmission Sub Module

Path transmission sub module path gets the status of path and path id is simulated, by which load can transfer from load balance to VM

C. Virtual Machine Sub Module

Virtual machine sub module which gets the status of Virtual machine and it contains virtual machine identification, RAM, bandwidth, and MIPS(Million Instruction Per Second).It also receives storage nodes and allocate those storage nodes into virtual machine in random process.

D. Job Provisioning Sub Module

Job provisioning sub module which gets status of jobs that are entering into a private cloud and job Id,file Size and particular path Id is simulated, from which is entered into Storage node.

E. Load Balancing Module

In this paper load balancing can be done with Max-Min Algorithm. It is one the scheduling algorithm. Allocation is done based on the maximum and minimum value. The mean completion times of all the machines are calculated and the machine whose completion time is larger than the mean value is selected. Then the tasks assigned to the selected machines are reassigned to the machines whose completion time is less than the mean value. The expected completion time of the task t_i on machine m_j is calculated by

$$\mathbf{Et\ (t_i, m_j) = Mch\ (m_j) + MT(t_i, m_j)}$$

Where Mch (m_j) is the machine availability time i.e. the time at which machine m_j completes any previously assigned tasks. MT (i,j) represents the estimated execution time for task t_i on machine m_j

Proposed related Definition of Max-Min Algorithm

- MT_{ij} the amount of time taken by machine M_j to execute $task_i$
- ET_j - the expected completion time of M_j
- Mch (m_j) - the machines availability time i.e. the time at which Machine j completes any previously assigned tasks.
- Cmb(ET,Machines)–The function "f1" is used to combine all the tasks and machines that has minimum.
- The best minimum task/machine pair (m, n) is selected from the Group

Max-Min Algorithm

1) While there are tasks to schedule
2) Every $Task_i$ to schedule
3) For all $Machine_j$
4) Compute $ET_{i,j}$; $ET_{i,j}$ = Mch (m_j) + T_{ij}
5) End for
6) Cmb (ET_i, Machines) =f1 (CT_i, 1, T_i, 2 ...)
7) End for
8) Select the best maximum pair (task, machine) from the combined Group
9) Compute level maximum $ET_{m,n}$
10) Reserve task m on n
11) End while
12) Calculate MeanET= (ΣET j)/No of machines
13) For all $Machine_j$
14) if (ET_j <=MeanET)
15) Select particular tasks reserved on the host
16) End for
17) Every $Task_i$ reselected
18) For all $Machine_j$ with (ET_j >=MeanET)
19) Compute $NewET_{i,j}$ = ET($task_i$, $host_j$)
20) if ($NewET_{i,j}$ >= MeanET)
21) Combine (ET_i, Machines) =f1 (ET_i, 1, ET_i, 2 ...)
22) End for
23) Choose the best maximum pair from the Combined Group
24) Reschedule ($task_i$ on $machine_j$)
25) Compute NewET
26) End for

6. Analysis and Report

Proposed algorithm is implemented with the help of simulation package like cloudsim and cloudsim based tool. Java language is used to implementing VM storage node load balancing algorithm. We assume that the cloudsim toolkit has

been deployed in 6 storage nodes with 4 virtual machines, where the parameter values are as under.

Parameter	Values
VM architecture	X86
VM bandwidth	Change according to each machine
VM memory	Change according to each machine
No. of processor each machine	6
Process element	5
MIPS	Randomly Varies

In simulation process job path is automatically simulated and assigned to VM by cloudsim according to the VM status, whenever the VM status changes job id and its path is also change according to it. A storage node gets automatically assigned into VM by simulation process The proposed Max-Min Algorithm is implemented for simulation.

Table 2 shows the result based on final Max-Min VM Load Balancing algorithm for overall balancing jobs on nodes in VM. In which Job length and current load of storage node for four VM that can analyze using cloudsim for our proposed algorithm.

Table 2

Virtual Machine ID	Job Length	Current Load
1	58715	0.099516
5	32529	0.151447
3	40154	0.124559
6	58073	0.195362

Analysis shows that Max-Min balancing algorithm consumes less time for storing jobs in node. When number of virtual machine or storage node increases it creates the respective jobId for reaching each and every node. It decreases the problem of deadlock by cloud environment job path provisioning policy in virtual macine. Figure 2 and Figure.3 describes the Max-Min produce the high sufficiency and scheduling efficiency to all jobs in the private cloud.

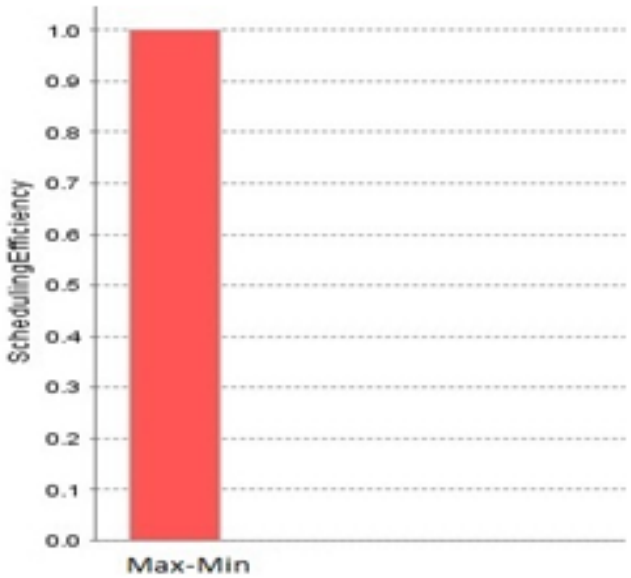

Figure 2

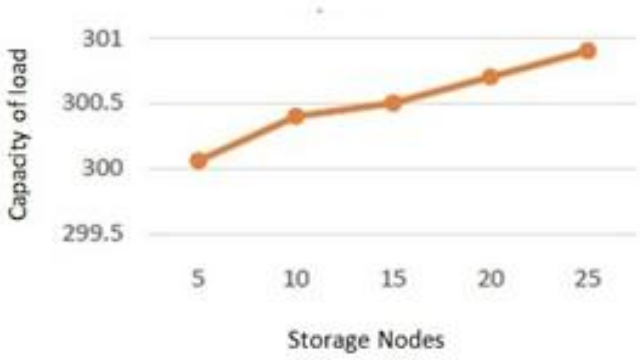

Figure 3

7. Conclusion and Future Work

The experimental results show that the Max-min algorithm which helps to store the jobs in quick and efficient manner to all the storage nodes in virtual machine and avoid traffic and load imbalance on storage nodes. The future research can also be extended to improving the complexity more and also fault tolerance can be considered, because after a job is submitted to the node, if the node may get affected which may lead to affect all jobs which submitted to that particular node.

Reference

[1] Martin Randles, David Lamb, A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing" in 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops

[2] Klaithem Al Nuaimi, Nader Mohamed, Mariam Al Nuaimi and Jameela Al-Jaroodi," A Survey of Load Balancing in Cloud Computing: Challenges and Algorithms" in 2012 IEEE Second Symposium on Network Cloud Computing and Applications

[3] Haozheng Ren, Yihua Lan,Chao Yin "The Load Balancing Algorithm in Cloud Computing Environment" in 2012 2nd International Conference on Computer Science and Network Technology

[4] Yupeng Zhang and Wing Shing Wong " Distributed Load Balancing in a Multiple Server System by Shift-Invariant Protocol Sequences "in IEEE Wireless communication and networking conference

[5] Chun-Wei Tsai, Wei-Cheng Huang, Meng-Hsiu Chiang, Ming-Chao Chiang, and Chu-Sing Yang," A Hyper-Heuristic Scheduling Algorithm for Cloud" in IEEE Transaction on Cloud Computing, Vol. 2, NO. 2, April-June 2014

[6] Z. Zhang and Xu. Zhang "A Load balancing mechanism based on ant colony and complex network theory in open cloud computing federation", 2nd International Conference on Industrial Mechatronics and Automation (ICIMA), Wuhan, China, vol. 2, pp.240-243, May 2010.

[7] H. Mehta, P. Kanungo, and M. Chandwani, "Decentralized content aware load balancing algorithm for distributed computing environments", Proceedings of the International Conference Workshop on Emerging Trends in Technology (ICWET), pp. 370-375, February 2011.

[8] Y. Lua, Q. Xiea, G. Kliotb, A. Gellerb, J. R. Larusb, and A. Greenber, "Join-Idle-Queue: A novel load balancing algorithm for dynamically scalable web services", An international Journal on Performance Evaluation, vol. 68, pp.1056-1071, November 2011.
[9] X. Liu, Pan, C. Wang, and J. Xie, "A lock-free solution for load balancing in multi-core environment", 3rd IEEE International Workshop on Intelligent Systems and Applications (ISA), pp. 1-4, May 2011.
[10] H. Liu, S. Liu, X. Meng, C. Yang, and Y. Zhang, "LBVS: A load balancing strategy for virtual storage", International Conference on Service Sciences (ICSS), pp. 257-262, May 2010.
[11] S. Wang, K.Yan , W. Liao and S. Wang , "Towards a load balancing in a three-level cloud computing network", 3rd IEEE International Conference on Computer Science and Information Technology (ICCSIT), vol. 1, pp. 108 - 113, July 2010.
[12] A. Bhadani, and S. Chaudhary, "Performance evaluation of web servers using central load balancing policy over virtual machines on cloud", Proceedings of the Third Annual ACM Bangalore Conference (COMPUTE'10), Article No. 16, January 2010.
[13] Z. Zhang, H. Wang, L. Xiao and L. Ruan, "A statistical based resource allocation scheme in cloud", IEEE International Conference on Cloud and Service Computing (CSC), pp. 263-273, December 2011.
[14] W. Bhathiya, "CloudAnalyst a CloudSim-based tool for modelling and analysis of large scale cloud computing environments", MEDC Project, Cloud Computing and Distributed Systems Laboratory, University of Melbourne, Australia, pp. 1-44, June 2009.
[15] M. Randles, D. Lamb, and A. Bendiab, "A comparative Study into distributed load balancing algorithms for cloud computing", IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA), pp.551-556, April 2010