A COMBINED BIN PACKING VM ALLOCATION AND MINIMUM LOADED VM MIGRATION APPROACH FOR LOAD BALANCING IN IAAS CLOUD DATACENTERS

Arya M B[1], Ajay Basil Varghese[2]

[1] *M-Tech Student , Department of Computer Science, Adi Shankara Institute of Engineering and Technology, Kalady, India*

[2] *Asst.Professor, Department of Information Technology, Adi Shankara institute of Engineering and Technology, Kalady, India*

Abstract

Day by day increasing online computation and migration to the cloud, significantly increase the need of proper load balancing. Load balancing ensure service availability and performance to users. Downtime experienced by users is great issue which leads to violation of QoS requirements for users. Improving response time is one of the solution and some extend it helps to meet the SLA. The proposed system has the intention of achieving an improved load balancing performance to cloud data center, and a satisfying service response to users. By focusing on virtualization technology here consider both virtual machine allocation and migration processes to achieve a better load balancing solution. A VM allocation policy called Bin packing and a Minimum loaded VM migration policy based on load threshold are used together to achieve the goal in Infrastructure as a Service cloud environment. A cloud simulation tool called CloudReport is used to perform the simulation of the system. The experimental results show that the approach provides comparatively improved response time and completion time for users.

Key Words: *Cloud computing, VM Allocation, VM Migration, Load Balancing, Infrastructure as a Service, Datacenters, Quality of Services*

----------***----------

1. INTRODUCTION

The concept of cloud computing has been around us for a few decades now. It gains lots of popularity among inter-net users and radically change the face of information and communication technology provisioning. Dynamic business industries adopt cloud due to its flexibility for redesigning capacity, according to changing business needs. Infrastructure as a Service (IaaS) of cloud facilitate the users to select com-puting instance with varying combinations of CPU, memory, storage, and networking capacity which offers flexible use of an appropriate mix of resources and pay for what they use [1]

Poor load balancing mechanisms lead to a significant amount of performance degradation and unavailability of ser-vices. A cloud service provider is responsible for ensuring the demanded Quality-of-Service (QoS) to users based on service level agreement. According to Ardagna *et al*. [2] QoS assurance is the issue of allocating resources to the application to guarantee a service level along dimensions such as performance, availability and reliability. One of the reasons for performance and availability issues is lack of load balancing. Unbalanced load leads to hotspot condition and under utilization of resources. Most of the time load balancing leads to a significant amount of downtime to users. Downtime refers the unavailability of services to users and it costs money.

By surveying 38 cloud services including Amazon, Microsoft Windows Azure, IBM etc. Cerin *et al.* [3] reveals that there experience a cumulative down time 2218.67 Hours to users in the year period 2007 to 2013. It states that down time of the majority of providers has grown from 2012 to 2013.One of the biggest cloud providers, Windows Azure, has almost tripled its downtime in 2013, standing at a total of 272.04 hours a year.

Virtualization technology offers a decoupling nature between physical machines and virtual machines in cloud datacenters and which yields efficient state (more correctly memory state) capture of VMs, which enables migration and restoration of virtual machines across physical machines [4]. This redistribution nature of virtual machines among physical machine after their first allocation opens a way to balance the load among the servers. So a load imbalance condition in cloud datacenters can be managed by this migration to some extent. The initial placement of virtual machines to physical machines also responsible for optimized resource utilization and initiating migration.

The main objective of this paper is to propose a method for achieving load balancing in cloud datacenters by combing an optimized VM allocation policy [5] and a VM migration policy. The performance of the method is analyzed through the considering completion time of requested job execution and response time experienced by the users. The remaining

sections of this paper organized as follows. Section 2, deals with related works on both VM allocation and migration in the cloud. Section 3 describes the proposed method and algorithms. The experimental setup and performance of the proposed system are evaluated in section 4. Finally, section 5 presents the conclusion and future work.

2. RELATED WORK

Virtual machine allocation or VM allocation [6] is the process of allocating virtual machines above the suitable physical machine (PM/host). During placement, hosts are selected on the basis of hardware and resource requirements of virtual machines and the expected resource usage. There are different policies existing for virtual machine placement. Mills *et al.* [7] discuss about 3 policies. In first fit policy virtual machines are allocated on the first host with sufficient available resources. Next fit policy chooses next host with sufficient resource of last chosen one. Randomly a physical machine is chosen for placing the virtual machine in random fit policy. Both three policies are simple and traditional. The main drawbacks of these policies are load imbalance and independent decision taking on resource capacities of virtual machines and resource utilization in physical machines.

First Fit Decreasing (Single Dimension) [8] perform sorting of physical machines and virtual machines based on capacity (high to low) and it applies first fit policy. Here consider all resources and assure resource utilization. Based on residual capacity of the host and by using dot product a method is described which also consider the virtual machine resource demand [9]. A minimizing angle method is discussed in [9] where the target host is selected based on the angle between the sum of physical machines resource utilization vector (RUVi) and virtual machine v resource demand RD with the total capacitance vector. It offers resource utilization and load balancing, but require a comparatively high computational cost for selecting PM and VM.

Live Virtual Machine (VM) migration is a unique capability of system virtualization, which allows a virtual machine (VM) to transparently move from one Physical Machine (PM) to another. Live migration is an essential management activity in datacenters for load balancing, server consolidation and server maintenance [4]. Load-based Controlling Scheme of Virtual Machine Migration is introduced by Zou *et al.* [10] which improve overall load balancing performance, avoid the unnecessary migration time and reduces the amount of data transferred. When compared to the CPU based strategy, it ensures well balanced storage and bandwidth load. Archer *et al.* [11] propose a strategy which periodically checks CPU and RAM utilization to figure out the load status among VMs. Migration performs in such a situation where even after scaling up the available resources, there is a peak utilization of re-sources. It provides improved performance of the applications running in virtual machine in terms of response time and distributes the load across the servers. Razali *et al.* [12] describe a strategy for improving overall load balancing performance by implementing the migration of virtual machines across multiple hosts, in which utilization of CPU resources can be optimized. The results provide a minimum migration of virtual machines and efficient utilization of resources. Chen *et al*. [13] discuss the VM migration time overhead issue and propose a network topology aware parallel migration. The strategy helps to speed up the load balancing process. Xu *et al.* [14] focus on avoiding violation of SLA insisted by cloud application through an interference-aware VM live migration strategy called iAware. They're finding that iAwar-Sandpiper can balance the CPU utilization of all PMs across the cluster in a better way compared to original Sandpiper strategy.

3. PROPOSED SYSTEM

The proposed system (figure 1) "A combined VM allocation and migration approach for load balancing in IaaS cloud datacenters" give focus to load balancing in cloud datacenters through VM migration. To achieve optimum VM allocation system use a Bin packing policy [5].

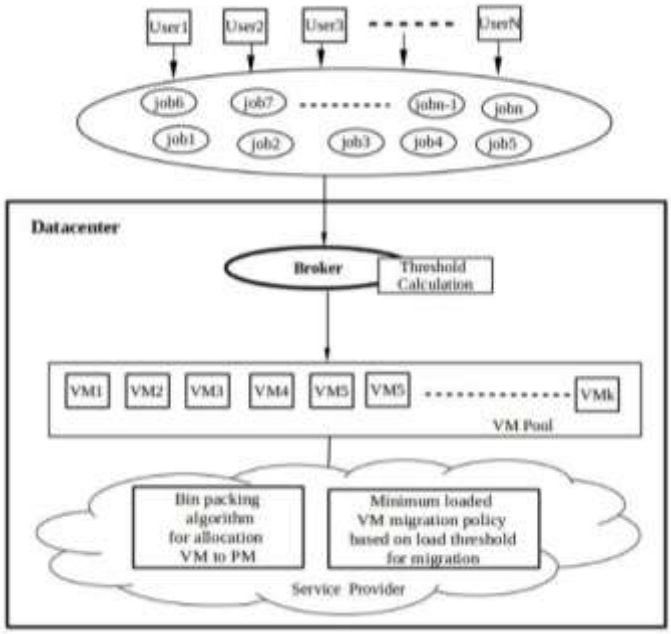

Figure1: System Architecture

The system is designed for IaaS cloud platform where the users have the choice for selecting virtual machine instances according to their requirement. When users specify their VM requirements, service providers create the virtual machines and allocate it to physical machines. Once VMs allocated it get started working by receiving workload in the form of user requests. The workload is distributed among the VMs, as they arrive by the datacenter broker. Dynamic workload in cloud leads to load imbalance among the PMs. When the datacenter broker detect under loading or overloading of PMs it perform a VM migration strategy called Minimum loaded VM migration policy based on load threshold. A threshold function is used to calculate marginal value for initiating migration. The performance of the system measured by taking response time and completion time of workloads. System architecture is given in figure 1

A. System Description

In Bin packing VM allocation policy hosts are considered as bins. Virtual machines are assumed to be the items that need to be filled in the bin. The aim of bin packing is to pack items to a minimum number of unit bins. The procedure for bin packing policy is given in algorithm1.

Algorithm 1 Bin packing procedure

Input : VMlist and PMlist

Sort VM in VMlist in ascending order based on storage requirement.

Sort PM in PMlist in descending order based on storage capacity.

for each VM in sorted VMlist

select first PM in sorted PMlist

if storage requirement of VM is less than or equal to storage availability of PM

allocate VM to PM

else choose next PM in sorted PMlist

end if

end for

In order to achieve load balancing migration are performed from overloaded PMs to under loaded PMs and perform Minimum loaded VM migration policy based on load threshold. To find, out load on each VM and PM there use the equations (1) and (2). Load on VM is finding out in terms of Mips. The load on each PM is found out by adding load on VMs among the PM.

Datacenter broker has the responsibility of finding out the threshold value of each PM. It is calculated based on equation (3). T_{PM} stands for threshold value of PM. This value changes with change in PM resource capacity. So different PMs has different threshold value. Find out the t% of each PM and compare it with the available capacity of PM. If a PMs load exceeds its threshold value it considered as overloaded. All other PMs are under the under loaded category. Here 't' takes a value 80. A source VM is selected from VMs running on the overloaded PMs. Based on load on each VM, they sorted in ascending order and select the top VM on list as source VM. For selecting destination PM here consider the under loaded PMs list. They sorted in ascending order based on load on it. So minimum loaded PM is at the top of the list. For each source VM the sorted PM list is scanned to meet the processing element requirement of VM. The search starts from the top of the list. When a PM with sufficient processing elements (CPU) is finding out add the VM to that PM. The overall migration process called " Minimum loaded VM migration policy based on load threshold" is described through algorithm 2 and 3.

$$VM_{load} = \frac{Submitted_{load}(Mips)}{Total\,CPU\,Capacity(Mips)} \quad (1)$$

$$PM_{load} = \sum VM_{load} \quad (2)$$

$$T_{PM} = PM_{capacity} * t/100 \quad (3)$$

Algorithm 2 Migration Source Selection Procedure

---

for each PM Calculate PM_{load} if $PM_{load} > T$

add PM to overload list

else add to underloaded list end for

Sort Overload PM list in descending order Sort underloaded PM in ascending order for each VM on overloaded PM

Calculate VM_{Load}

Sort VM based on load in descending order

Select first VM from sorted list. Add to VM migration list end for

---

Algorithm 3 Migration Destination Selection Procedure

---

for each VM in VM Migration list Calculate processing element number

end for

for each PM in sorted overloaded PM list Calculate available processing element number

end for

for each VM in VM migration list and a first PM in sorted overloaded PM list

if $VM_{PE} <= PM_{PE}$ add VM to PM

end if end for

---

4. EXPERIMENTAL SETUP AND RESULT

The experiment is carried out in CloudReport [15] frame-work. CloudReports is a graphic tool build on top of CloudSim [16] that simulates distributed computing scenarios on the basis of Cloud Computing paradigm and facilitate the simulation of Infrastructure as a Service (IaaS)

provider. In experiments carried out a host is characterized by properties like operating system, processing elements, RAM, storage, bandwidth and scheduling policies. Capacity of processing elements is defined in MIPS. A VM characterizes its operating system, processing elements, RAM, scheduling policies and hypervisor. A datacenter consist of several hosts and a broker to manage the datacenter.

A datacenter is created by the service provider (SP). SP create a number of hosts inside the datacenter. Here SP specifies which allocation policy and migration policy used in the datacenter. Customers have the capability to add the virtual machines of required properties. They specify the Broker policy and workload properties. The workload is termed as CloudLet in Cloud Report environment. A CloudLet is characterized in terms of RAM and CPU usage. An experiment carried out by using bin packing policy as the broker policy, especially for allocating VMs among hosts and minimum loaded VM migration policy based on the load threshold as migration policy. Experiments are carried out by varying workload and varying capacity of VMs and PMs.

The dynamic nature of cloud is set by initializing random load to each VM. Varying workload nature is established by using the DVFS policy [17] for workload scheduling. Results show that the system achieves a better load balancing status. To find out the efficiency of this system, it compares with two existing strategies in terms of response time and completion time.

Completion time is compared with Razali *et al*. [12] by configuring similar environment. In both methods the VM with minimum load are migrated. One data center with two physical hosts having 2048MB and 8192MB RAM are created. 5 Virtual machines of 2048MB RAM also created. Consider workload as Cloudlets of 300byte (Combination of RAM & CPU usage). Simulation result shown in figure 2. Through the experimental set up here compares the completion time of proposed strategy and Razali *et al*.'s strategy. Here proposed strategy exhibit slightly better result than the compared strategy.

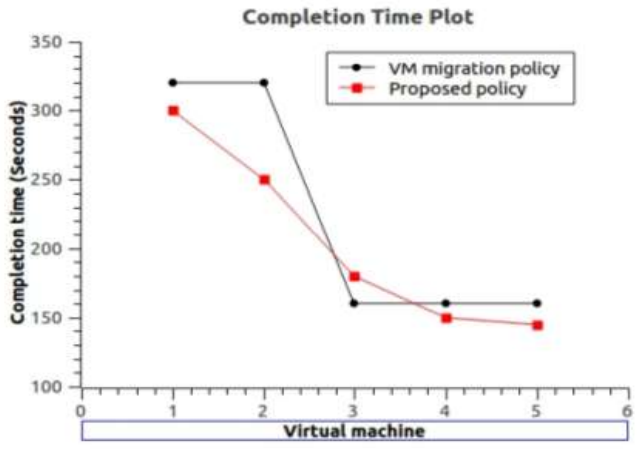

Figure 2: Completion time plot

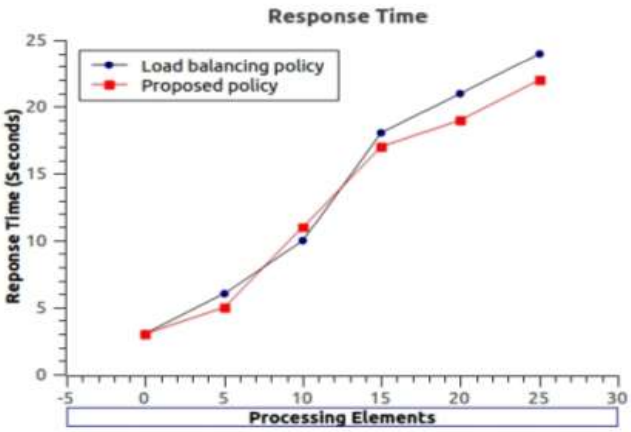

Figure 3: Response time

Proposed strategy is compared with the Achar *et al*. [11] method by configuring similar working environment. In this strategy of VM migration there randomly select a VM from overloaded VM as source. The experiments carried out on following condition. One data center with 3 physical hosts of 2.93 GHz processing capacity and 2 GB RAM is created. 3 Virtual machines of various capacities placed above the host. The studies reveal the efficiency of the proposed method in terms of completion time and response time. Figure 3 shows the result.

5. CONCLUSIONS

The proposed work presented a VM placement policy together with a VM migration policy to improve the performance of the applications running in virtual machine in terms of response time and completion time. Here conducted an experiment on CloudReport Platform. The strategy helps to achieve load balancing among cloud datacenters by mi-grating the VMs from an overloaded PM to an under loaded PM based on available resource capacity. DVFS policy and random generator function are used to stimulate the dynamic nature of cloud datacenters. By considering response time and completion time as QoS parameters the proposed system provides comparatively better result.

The introduced model is performed in a simulation environment. It is assumed that the workload as CloudLets, but in real time the workload characteristic is different based on the application. Real time implementation of the system is required to analyze the detailed behavior of the system. For Ensuring QoS requirement in addition to response time and can also implement a priority based scheduling policy for the workloads.

Green computing is a serious issue for cloud datacenters. So there requires reduction of number of migrations. When the threshold value used as 80 % it will lead to bad server consolidation and also energy wastage. So there requires a modification of strategy, avoiding instant migration of VMs. on meeting threshold. A load prediction mechanism will be a solution for this.

REFERENCES

[1] Amazon web services, http://aws.amazon.com/, 2015

[2] Danilo Ardagna , Giuliano Casale , Michele Ciavotta , Juan F Pérez and Weikun Wang, Quality-of-service in cloud computing: modeling techniques and their applications, Journal of Internet Services and Applications, Springer, 2014

[3] C. Cerin, C. Coti, P. Delort, F. Diaz, M. Gagnaire, Q. Gaumer, N. Guil- laume, J. Lous, S. Lubiarz, J.-L. Raffaelli, et al., "Downtime statistics of current cloud solutions," International Working Group on Cloud Computing Resiliency, 2013.

[4] Mayank Mishra, Anwesha Das, Purushottam Kulkarni, and Anirudha Sahoo, Dynamic Resource Management Using Virtual Machine Migra-tions, IEEE, 2012

[5] K R Remesh Babu and Philip Samuel, Virtual Machine Placement for Improved Quality in IaaS Cloud, Fourth International Conference on Advances in Computing and Communications, IEEE, 2014

[6] Georgiou, Stefanos, Konstantinos Tsakalozos, and Alex Delis. "Exploit-ing Network-Topology Awareness for VM Placement in IaaS Clouds." Cloud and Green Computing (CGC), 2013 Third International Confer-ence on. IEEE, 2013.

[7] K. Mills, J. Filliben and C. Dabrowski, Comparing VM-Placement Al-gorithms for On-Demand Clouds, Third IEEE International Conference on Coud Computing Technology and Science, IEEE, 2011

[8] S. Srikantaiah, A. Kansal, and F. Zhao, "Energy aware consolidation for cloud computing," in Proceedings of the 2008 conference on Power aware computing and systems, ser. HotPower'08. Berkeley, CA, USA: USENIX Association, 2008, pp. 10–10. 35.

[9] M. Mishra and A. Sahoo, "On theory of vm placement: Anomalies in existing methodologies and their mitigation using a novel vector based approach," in Proceedings of the 2011 IEEE 4th International Conference on Cloud Computing, ser. CLOUD '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 275–282.

[10] Chao Zou , Yueming Lu, Fangwei Zhang, Songlin Sun, Load-based Controlling Scheme Of Virtual Machine Migration, IEEE, 2012

[11] Raghavendra Achar, P. Santhi Thilagam, Nihal Soans, P. V. Vikyath , Sathvik Rao and Vijeth A. M, Load Balancing in Cloud Based on Live Migration of Virtual Machines, IEEE, 2013

[12] Rabiatul Addawiyah Mat Razali, Ruhani Ab Rahman, Norliza Zaini, Mustaffa Samad Faculty, Virtual Machine Migration Implementation in Load Balancing for Cloud Computing, IEEE, 2014

[13] Kun-Ting Chen, Chien Chen, Po-Hsiang Wang, Network Aware Load-Balancing via Parallel VM Migration for Data Centers Kun-Ting, IEEE, 2014

[14] Fei Xu, Fangming Liu, Linghui Liu, Hai Jin, Bo Li, and Baochun Li, iAware: Making Live Migration of Virtual Machines Interference-Aware in the Cloud, Ieee Transactions On Computers, 2014

[15] Thiago Teixeira Sa, Rodrigo N. Calheiros and Danielo G. Gomes, CloudReports: An Extensible Simulation Tool for Energy-Aware Cloud

[16] Ranjan , Anton Beloglazov , Cesar A. F. De Rose and Rajkumar Buyya, CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, Wiley Online Library, 2010

[17] Tom Guérout, Thierry Monteil, Georges Da Costa, Rodrigo Neves Cal-heiros, Rajkumar Buyyae, Mihai Alexandru, Energy-aware simulation with DVFS Tom, Elsevier, 2013.