RESEARCH ARTICLE OPEN ACCESS

A New Load Balancing Algorithm for Virtual Machine Allocation in Cloud Computing

Pandaba Pradhan[1], Prafulla Ku. Behera[2], B N B Ray[3]
[1]Asst. Professor& H.O.D, Department of Comp. Sc.
BJB (Auto) College, BBSR, Odisha, India.
[2] Reader & H.O.D, Department of Comp. Sc & Applications.
Utkal University,BBSR, Odisha, India.
[3] Reader, Department of Comp. Sc & Applications.
Utkal University, BBSR, Odisha, India.

ABSTRACT
Cloud computing has been recognised as a cost-effective model for computingon the Internet. Virtualization that enables cloud computing, shares a pool of physical machines with a large number of virtual machines running massive user-applications. This necessitates efficient allocation of virtual machines to physical machines in order to balance the load on physical machines and ensure better quality of service. However, unpredictable and unequal load of user applications impose severe challenges on cloud datacenters in view of effective resource management. In this research, a new load balancing algorithm is proposed for efficient allocation of virtual machines to physical machines. The algorithm has been implemented in CloudSim and compared with the existing worst-fit algorithm. Resulted data shows the proposed algorithm outperforms the other algorithm in resource management.

Keywords: Virtual Machine Allocation, Load Balancing Algorithm, Cloud Computing, Datacenter Resource Management, Virtualization.

Date of Submission: 24-11-2020 Date of Acceptance: 07-12-2020

I. INTRODUCTION

Cloud computing offers on-demand access to virtually unlimited computing resources over the Internet. It provides metered computing services that meet changing computing needs of the users. The cost-effective pricing model is popularly known as “pay-as-you-go” pricing model. Users do not own the resources; rather they share it with other users in a secured multitenant environment. They can scale-in and scale-out the resources in terms of the number and computing power of the resources as and when they need [1,13]. Virtualization is the key enabling technique for cloud computing that allows creation of large number of virtual resources for a limited number of physical resources and makes the computing model fascinating.

Cloud computing offers three basic service models: software-as-a-service (SaaS), platform-as-a-service (PaaS) and infrastructure-as-a-service (IaaS). SaaS provides a cost-effective way of using application software to meet data processing requirements. It eliminates the need for buying and installing the application software in user’s computer thereby removes the maintenance and support costs. Similarly, PaaS delivers development and execution environments that include operating system, database, application server and development frameworks for users to develop and host their applications. On the other hand, IaaS enables users to create their own virtual resources such as virtual machines, storage and networks in order to configure and control their own computing infrastructure.

The adoption of cloud computing services has been manifold in recent times. In particular, the rising growth of artificial intelligence and machine learning technologies facilitates an increasing demand for cloud-based service. According a market forecast, the cloud computing global market size has been projected to reach 761 billion USD by 2027 from 199 billion USD in 2019, a CAGR (Compound Annual Growth Rate) of 18.6% during the period [12,14]. The popularity of cloud computing is due to the cost-effectiveness it offers. However, cloud computing is not devoid of challenges [2,15]. A major challenge is the service availability and quality of service when there is high load of service demand coming from millions of users. The challenge can be addressed through efficient task scheduling and load balancing to utilize effectively the physical computing resources in cloud computing datacenters. Consequently, it motivates to carry out research and investigations in the areas of cloud resource management [3,16,17].

In a cloud environment, virtual machines (VMs) are hosted on one or more physical machines (PMs) by provisioning right amount of physical resources such as processor, memory, storage, and networking elements etc. A large number of VMs are created by increasing number of cloud users to host their applications. The task demands on the VMs are dynamic and unequal in nature. Further, cloud datacenters are constrained with limited number of physical resources. Thus, anefficient mapping of VMs onto PMs is required for effective resource management in cloud datacenters.

II. RELATED WORKS

A large number of research efforts are available that propose load balancing algorithms for effective resource management in cloud computing. Some of the important works are reviewed here.

According to Rewehel et al. [6], the OLB (Opportunistic Load Balancing) scheduling algorithm is used to allocate the taskand divides a task into subtasks in a three level cloud computingnetwork (i.e., Request manager, Service manager, Service node)for assigning and solving the workload in the least time. It doesnot take additional calculations for the allocation and load balancingof tasks. Rather it considers overall expected completion time to executea task. They have measured the makespan of the systemthrough the algorithm. The merit of OLB is to keep all hosts busyas much as possible which shows better efficiency and maintainproper balancing of the load for the system. OLB is not suitablefor cloud environment due to poor make-span when multipleobjectives are considered simultaneously.

Radojevic et al [7] proposed an improved algorithm over roundrobin called Central Load Balancing DecisionModel. It uses the basis of round robin but it alsomeasures the duration of connection between client andserver by calculating overall execution time of task ongiven cloud resource.

Liet al. [5] have proposed an enhanced Max–Min algorithm thatkeeps a task status table to measure the real-time load of virtual machines aswell as the expected completion time of tasks. After allocation of a task to a virtual machine following Max-Min procedure, that task is removed from the queue and the algorithm proceeds forward for the distributionof the rest all unallocated tasks. The algorithm proposed them is betterthan the round robin technique for the consideration of averagetask pending time.

Chen et al. [4] have introduced animproved Min-Min load balancing algorithm to optimize the makespan and enhance the resource utilization.The algorithm proposed by them splits all the tasks into two groups: higher priority tasks and lower prioritytasks. It schedules all the tasks of higher priority first andthen moves to the allocation of tasks in the lower priority group. Finally, the load balancingfunction is operated to optimize the particular load of eachmachine to generate the final schedule.

Lee et al [8] proposed a load balancing technique indynamic environment based on weighted least connection. It allocates theresource with least weight to a task and takes into accountnode capabilities. Based on the weight and capabilitiesof the node, task is assigned to a node.

The loadbalancing algorithm proposed in [9]uses three level frameworks for resource allocation indynamic environment. It uses opportunistic loadbalancing algorithm as its basis. Since cloud is massivelyscalable and autonomous, dynamic scheduling is betterchoice over static scheduling.

Kim et al. [10] have used minimum compilation time technique where they considered both ready-to-execute timeand the expected execution time of the tasks for load balancing purpose.In that, they allocated the task that has least completiontime to an appropriate core.

III. THE PROPOSED ALGORITHM

The task of virtual machine allocation to physical machines can be viewed as an optimization problem in cloud datacenters. The allocation of VMs onto PMs should be such that the load on a PM is optimized while ensuring the quality of service delivery. The optimization problem can be stated as follows:

Let V1, V2, V3 . . . represent VMs and P1, P2, P3 . . .represent PMs in a datacenter. Let req_Pes represents the required processing elements of a VM, and free_Pes represents the free processing elements available in a PM.Now, a virtual machine V_i is assigned to a physical machineP_i only if P_i has the minimum number of free_Pes such that free_Pes of P_i is greater than or equal to req_Pes of the virtual machine V_i.The algorithm of the above assignment problem is presented here using the pseudo code.

```
Input: a set of virtual machines $V_i$ and a set of physical machines$P_j$
Output: a set of ($V_i$, $P_j$), i.e. VMs to a PM allocation pair
1. For each virtual machine $V_i$ that is not assigned to a physical machine$P_i$
```

```
Calculate req_Pes($V_i$)
   For each physical machine$P_j$
      If (free_Pes($P_j$) >= req_Pes($V_i$) )
          Find min( freePes($P_j$) - req_Pes($V_i$) )
      End if
   End for
  Assign the virtual machine Vi to the physical machinePj
      Update free_Pes(Pj) = free_Pes(Pj) – req_Pes(Vi)
End for
```

IV. IMPLEMENTATION

The implementation of the proposed algorithm has been carried out using CloudSim simulator, a popular Java based simulation environment for cloud computing [11].The open source software, CloudSim emulates all the characteristics of a cloud environment through its 12 packages and a large number of classes it contains. Out of 12 packages it has, the most relevant toa cloud researcher is org.cloudbus.cloudsim. This class contains the codes for modeling the various cloudentities like Datacenter, physical machine known as Host, task known as Cloudlet and VM. They also define various resource scheduling and provisioning policies. One can extend or overwrite these classes to define new cloud entities, change existingcloud entities and create new policies.

The class vmAllocationPolicy can be extended to accommodate user-defined policies for VM allocation to host for load balancing. By default, CloudSim provides a class called vmAllocationPolicySimple that implements a worst-fit algorithm as the default host selection policy. According to this algorithm, once a VM execution task comes in, all the available hosts are scanned and the host that is having maximum number of processing elements (Pes) or CPUs is selected for the VM execution. To test the efficacy of the proposed algorithm, the available class vmAllocationPolicy has been extended and a new class called vmAllocationPolicyOptimized is created that implements the algorithm.

4.1Simulation Configuration

The experiment has been carried out by configuring VMs, Hosts and Tasks in a datacenter. The VMs are configured uniformly (e.g. MIPS=250; Image Size=10000; RAM=2048; Bandwidth=1000 and CPUs=1). Each VM carries a single cloudlet (task) of equal configuration (e.g. Length = 40000; FileSize = 300; Output Size = 300).However, the hosts have varied configurations (Table 1).

Table 1: Host Configuration

Id	MIPS	Storage	RAM	Bandwidth	No of CPUs
0	1000	100000	4096	10000	2
1	1000	150000	8192	10000	4
2	1000	100000	4096	10000	2
3	1000	100000	4096	10000	2
4	1000	100000	6144	10000	3

4.2 Simulation Results

Several simulation runs have been carried out by gradually increasing the number of VMs to impose load on available hosts. The incremental load depicts how an algorithm allocates the VMs to hosts and balances load among them. Both the algorithms (the existing and the proposed one) have been subjected to run under the above configurations to compared their performance. Host utilization is considered as a performance metric for the purpose. The data about VM allocation to host and host utilization have been recorded for each algorithm and simulation run. The average percentage of host utilization (based on its available processing elements) of each algorithm has been computed using the following formula:

Average % of Host Utilization = Total % of Utilization / No of Participated Hosts

Simulation results are tabulated in tables (Table 2 - 8) with VM allocation and Host utilization.

Table 2: Simulation No-1: (5 VMs, 5 Hosts)

	VM Allocation to Host				
Algorithm	Host0	Host1	Host2	Host3	Host4
Existing	VM3	VM0, VM1, VM4	Free	Free	VM2
New	VM0, VM1	Free	VM2, VM3	VM4	Free
	Host Utilization				
Existing	50%	75%	Free	Free	50%

New	100%	Free	100%	50%	Free

Table 3: Simulation No-2: (6 VMs, 5 Hosts)

	VM Allocation to Host				
Algorithm	Host0	Host1	Host2	Host3	Host4
Existing	VM3	VM0, VM1, VM4	VM5	Free	VM2
New	VM0, VM1	Free	VM2, VM3	VM4, VM5	Free
	Host Utilization				
Existing	50%	75%	50%	Free	50%
New	100%	Free	100%	100%	Free

Table 4: Simulation No-3: (7 VMs, 5 Hosts)

	VM Allocation to Host				
Algorithm	Host0	Host1	Host2	Host3	Host4
Existing	VM3	VM0, VM1, VM4	VM5	VM6	VM2
New	VM0, VM1	Free	VM2, VM3	VM4, VM5	VM6
	Host Utilization				
Existing	50%	75%	50%	50%	33.33%
New	100%	Free	100%	100%	33.33%

Table 5: Simulation No-4: (8 VMs, 5 Hosts)

	VM Allocation to Host				
Algorithm	Host0	Host1	Host2	Host3	Host4
Existing	VM3	VM0, VM1, VM4	VM5	VM6	VM2, VM7
New	VM0, VM1	Free	VM2, VM3	VM4, VM5	VM6, VM7
	Host Utilization				
Existing	50%	75%	50%	50%	66.66%
New	100%	Free	100%	100%	66.66%

Table 6: Simulation No-5: (9 VMs, 5 Hosts)

	VM Allocation to Host				
Algorithm	Host0	Host1	Host2	Host3	Host4
Existing	VM3, VM8	VM0, VM1, VM4	VM5	VM6	VM2, VM7
New	VM0, VM1	Free	VM2, VM3	VM4, VM5	VM6, VM7, VM8
	Host Utilization				
Existing	100%	75%	50%	50%	66.66%
New	100%	Free	100%	100%	100%

Table 7: Simulation No-6: (10 VMs, 5 Hosts)

	VM Allocation to Host				
Algorithm	Host0	Host1	Host2	Host3	Host4
Existing	VM3, VM8	VM0, VM1, VM4, VM9	VM5	VM6	VM2, VM7
New	VM0, VM1	VM9	VM2, VM3	VM4, VM5	VM6, VM7, VM8
	Host Utilization				
Existing	100%	100%	50%	50%	66.66%
New	100%	25%	100%	100%	100%

Table 8: Simulation No-7: (11 VMs, 5 Hosts)

	VM Allocation to Host				
Algorithm	Host0	Host1	Host2	Host3	Host4
Existing	VM3, VM8	VM0, VM1, VM4, VM9	VM5, VM10	VM6	VM2, VM7
New	VM0, VM1	VM9, VM10	VM2, VM3	VM4, VM5	VM6, VM7, VM8
	Host Utilization				
Existing	100%	100%	100%	50%	66.66%
New	100%	50%	100%	100%	100%

The averageutilization of hosts by each algorithm in each simulation run is computed and shown in Table 9. The corresponding graph is shown in Figure 1.

Table 9: AverageUtilizationofHostsin Percentage

Simulation No	No of VMs	Existing Algorithm	New Algorithm
1	5	58.33	83.33
2	6	56.25	100
3	7	51.67	83.33
4	8	58.33	91.67
5	9	68.33	100
6	10	73.33	85
7	11	83.33	90

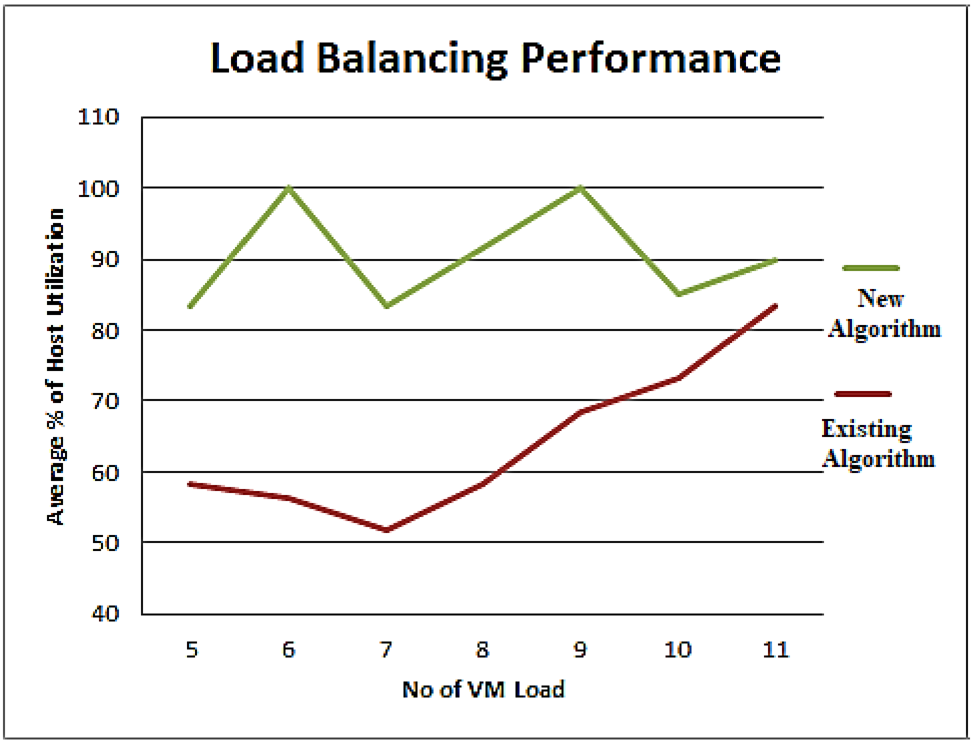

Figure 1: Load Balancing Performance Comparison

As per the simulation configuration, a host can accommodate a maximum number of VMs based on the number of CPUs the host has. It is clear from the tables (Table 2 - 8) that when the total VM load is less, the proposed algorithm performs better in balancing the load on minimum number of hosts, optimizing the utilization of processing elements, and making some hosts free. On the other hand, the existing algorithm allocates VMs to hosts making them underutilized. The underutilization of a host is not desirable as it increases energy consumption and also not economical.Figure 1 shows that the proposed algorithm performs better VM allocation and balances the load effectively compared to the existing algorithm. Thus, the proposed algorithm optimizes host utilization, makes host free when required and remains energy-efficient.

V. CONCLUSIONS

In this paper, load balancing algorithms available in the literature have been reviewed and a new load balancing algorithm to allocate VMs to PMs is proposed. The algorithm is implemented in CloudSim, a popular cloud computing simulator. Several simulation runs have been carried out to generate experimental data for both the proposed algorithm and the existing algorithm of CloudSim. The generated data suggest that theproposed algorithm performs load balancing better and improves host utilization in a cloud datacenter. The proposed algorithm is simple but effective.

REFERENCES

[1]. Mell P. and GranceT. (2011)The NIST Definition of Cloud Computing, available at https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf, accessed on 6th June 2020.

[2]. Colin Ting Si Xue, Felicia Tiong Wee Xin. (2016), Benefits and Challenges of the Adoption of Cloud Computing, In Business, International Journal on Cloud Computing: Services and Architecture (IJCCSA) Vol. 6(6), pp.1-15.

[3]. Singh D., Banyal R.K., Sharma A.K. (2019) Cloud Computing Research Issues, Challenges, and Future Directions. In: Rathore V., Worring M., Mishra D., Joshi A., Maheshwari S. (eds) Emerging Trends in Expert Applications and Security. Advances in Intelligent Systems and Computing, vol 841. Springer, pp. 617-623.

[4]. Chen, H., Wang, F., Helian, N., Akanmu, G. (2013) User-priority guided Min-Min scheduling algorithm for load balancing in cloud computing. IEEE Natl. Conf. Parall. Comput. Technol. (PARCOMPTECH), 1–8.

[5]. Li, X., Mao, Y., Xiao, X., Zhuang, Y., (2014), June. ’An improved max-min taskscheduling algorithm for elastic cloud’, In: IEEE International Symposium on Computer, Consumer and Control (IS3C), pp. 340–343.

[6]. Rewehel, E.M., Mostafa, M.S.M., Ragaie, M.O. (2014), October. ’New Subtask Load Balancing Algorithm Based on OLB and LBMM Scheduling Algorithms in Cloud’, In Proceedings of the 2014 International Conference on Computer Network and Information Science, IEEE Computer Society, pp. 9–14.

[7]. Radojevic, B. &Zagar, M. (2011). Analysis of issues with load balancing algorithms in hosted (cloud) environments. In proceedings of 34th International Convention on MIPRO, IEEE.

[8]. Lee, R. &Jeng, B. (2011). Load-balancing tactics in cloud. In proc. International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), IEEE, pp. 447-454.

[9]. Wang, S. C., Yan, K. Q., Liao, W. P. & Wang, S.S. (2010). Towards a load balancing in a threelevelcloud computing network. Proceedings of 3rd International Conference on Computer Science and Information Technology (ICCSIT), IEEE, pp.108-113.

[10]. Kim, S.I., Kim, H.T., Kang, G.S., Kim, J.K. (2013), June. ’Using dvfs and task scheduling algorithms for a hard real-time heterogeneous multicore processor environment’, In: Proceedings of the 2013 workshop on Energy efficient high performance parallel and distributed computing, ACM, pp. 23–30.

[11]. CalheirosR. N., Ranjan R., BeloglazovA., De Rose C. F., and BuyyaR. (2011) “CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms,” Softw. – Pract. Exp., Vol. 41(1), pp. 23–50.

[12]. Cloud Computing Market Forecast 2020-2027 (2020), available at https://www.fortunebusinessinsights.com/cloud-computing-market-102697, accessed on 07th July 2020.

[13]. Pandaba Pradhan , Prafulla ku. Behera , BNB Ray(2016) , “Modified Round Robin Algorithm for Resource Allocation in cloud computing” , ELSEVIER Procedia Computer Science.

[14]. K Sumalatha, M. S. Anbarasi(2019), "A review on various optimization techniques of resource provisioning in cloud computing." International Journal of Electrical and Computer Engineering (IJECE), Vol.9, No.1.

[15]. Pandaba Pradhan , Prafulla ku. Behera , BNB Ray(2020), “Enhanced Max-Min Algorithm for Resource Allocation in Cloud Computing ” , International Journal of Advanced Science and Technology (IJAST), Vol.29,No.8.

[16]. C. Jiang, G. Han, J. Lin, G. Jia, W. Shi, W. Wan(2019), “Characteristics of Co-allocated Online Services and Batch Jobs in Internet Data Centers: A Case Study from Alibaba Cloud. “,IEEE Access.

[17]. Tram Truong Huu& John Montagnat.(2010), ”Virtual Resource Allocations distribution on a cloud infrastructure” ,IEEE,pp.612-617.

Pandaba Pradhan, et. al. “A New Load Balancing Algorithm for Virtual Machine Allocation in Cloud Computing.” *International Journal of Engineering Research and Applications (IJERA)*, vol.10 (12), 2020, pp 26-32.