www.ijecs.in
*International Journal Of Engineering And Computer Science ISSN:2319-7242*
*Volume 3 Issue 11 November, 2014 Page No. 9197-9200*

# Cloud Computing: VM placement & Load Balancing

***Adeel Hashmi***

Maharaja Surajmal Institute of Technology, Department of Computer Science
Janakpuri, New Delhi, India
*ashashmi10@gmail.com*

**Abstract:** ***In recent years, cloud computing has become a popular paradigm for hosting and delivering services over the internet. The key technology that makes cloud computing possible is server virtualization, which enables dynamic sharing of physical resources. Through virtualization, a cloud service provider can ensure QoS delivered to the user while achieving higher server utilization and energy efficiency. Virtualization introduces the problem of virtual machine placement and also increases the overheads in load balancing. This paper discusses the various the various algorithms dealing with VM placement and load balancing in cloud environment.***

**Keywords:** Cloud computing, Virtualization, Virtual Machines, Load balancing.

## 1. Introduction

Clusters [1] are distributed systems under the supervision of single administrative domain. Grid [1] is a geographically dispersed collection of distributed systems. Cloud is a collection of parallel and distributed system where the nodes are virtualized whereby a single physical server can run multiple virtual servers, thus reducing the resources as well as the cost.

A cloud can be public, private or hybrid [2]. Private clouds are setup by enterprises for their internal use only. Public clouds are setup for public use by the enterprises. The users of a public cloud must agree to the Service Level Agreement (SLA) specified by the cloud provider. Hybrid cloud is a combination of private and public cloud. OpenStack is one the most popular software used to setup a cloud, others being Eucalyptus, OpenNebula, etc.

There are three major types of services provided on a cloud: Infrastructure as a Service (IaaS), Platform as a Service (PaaS), Software as a Service (SaaS).

*Software-as-a-Service* is software which is deployed over the internet and used by someone on a personal computer or local area network. Popular example is Microsoft Office 365.

*Platform-as-a-service* in the cloud is defined as a set of software development tools hosted on the provider's infrastructure. Applications are created on the provider's platform over the internet. Popular examples are Google App Engine, Windows Azure.

*Infrastructure-as-a-Service* provides the customer with storage and virtual server instances, as well as APIs that allow the customer to configure their virtual servers and storage. This model allows the organization to pay for only as much capacity as is needed, and scale up/down as soon as required. Popular example is Google Drive, Amazon EC2 (Elastic Compute Cloud).

All the given examples i.e. MS Office 365, Google App Engine, Windows Azure, Google Drive, Amazon EC2 are public clouds, so any user can access them through an internet connection. Organizations can also setup private clouds for their internal use, which run on their private network.

## 2. Virtualization

Virtualization (also called Server Virtualization) is the key technology that makes cloud computing possible. Virtualization enables multiple applications to run on a single physical server simultaneously, inside performance-isolated platforms called Virtual Machines. Creation of a virtual machine requires software known as Hypervisor or Virtual Machine Monitor (VMM). There are two types of hypervisors: Bare metal (Type-1) and Hosted (Type-2). Type-1 hypervisors run directly on the top of the hardware, whereas Type-2 run inside an operating system. Examples of Type-1 hypervisors are Xen, Microsoft Hyper-V. Examples of Type-2 hypervisors are Virtualbox, VirtualPlayer.

Internet hosting service companies use VMMs to provide virtual private servers. A virtual private server which is dynamic (can be moved to other hardware according to load while it is still running) is referred to as cloud server.

Amazon EC2, IBM SoftLayer use Xen as the primary VMM for their product offerings.

## 3. Cloud Architecture

A cloud can be said to have 4 layers: Tasks are executed on Virtual Machines, several Virtual Machines reside on a host (physical server), and several hosts at a single physical location aggregate to form a single datacenter entity. The data-centers may be dispersed at different geographical locations all over the world.

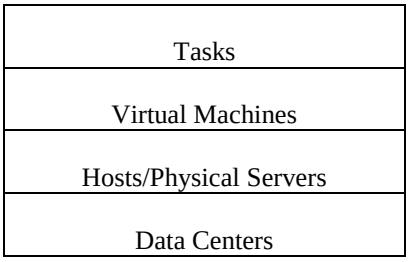


**Figure 1:** Cloud Architecture

Host is the physical server which can have multiple cores for parallel processing; it is responsible for assigning processing cores to the virtual machine. More than one instance of virtual machine can be mapped onto a single host. Tasks are executed on a virtual machine. Tasks must be mapped onto appropriate virtual machine based upon its configuration and availability.

There are few simulation tools available now which can be used to simulate a cloud. The most popular ones are CloudSim (and related projects like CloudAnalyst), iCanCloud, GDCSim (Green Data Center Simulator), etc.

## 4. Why Virtualization?

Consider we have 7 non-virtualized servers hosting 7 applications labeled App1 to App7. For each server, we have application resource requirements (memory requirements, etc.) over a period of time. We have to move applications from these servers to virtualized servers on a cloud, where each host has a quad-core processor which is capable of executing up to 4 VMs in parallel.

As we can see from the figure 2, the non-virtualized hosts are not fully utilized and we are using 7 of them. By virtualizing the servers, the numbers of hosts required have reduced to 3, and each of these hosts is being better utilized.

For efficient resource utilization, and reduction of cost and energy consumption, it is important that the VMs are consolidated to minimal number of physical nodes.

The following are few approaches used to handle VM placement problem:
*Bin packing*. The VM placement problem is often considered as a variant of bin-packing problem, which is a NP-hard problem. Several heuristics have been proposed to handle this problem [4-8].

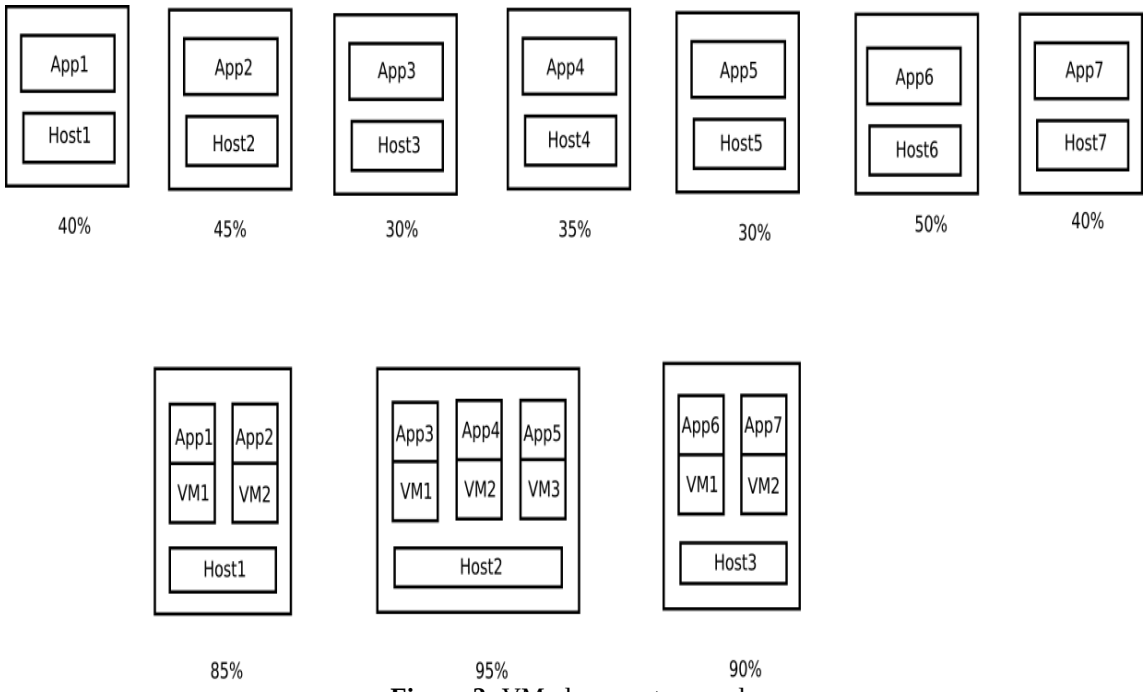


**Figure 2:** VM placement example

*Constraint programming*. Constraint programming methods have also been used to handle VM placement problem [9-10].
*Linear programming.* Several approaches based on linear programming have been proposed [11-13].

Genetic algorithm. Another approach to VM allocation problem is to use genetic algorithm [14-15].

## 5. Load Balancing

Suppose all the hosts and VMs across all the data centers across the world are supporting the same social networking application. Then a task initiated by a user at some location in the world has to be assigned a specific VM in a specific data center.

There are various algorithms for data center allocation (called service-broker policies) like *closest data center*, *optimize response time*, *reconfigure dynamically with load* as well as for load distribution across multiple VMs in a single data center like *round robin*, *equally spread current execution load*, *throttled*.

The "closest data center" policy simply routes the traffic to the closest data center. In "optimize response time" policy, the service broker monitors the performance of each data center, and directs traffic to the data center with best response time. The "reconfigure dynamically with load" policy is an extension to closest data center policy, where the routing logic is similar, but the service broker has one more responsibility of increasing or decreasing the number of virtual machines allocated in the data centers based on the load.

After a task has been allotted a data center, the task has to be allotted a VM. Let VM = $\{VM_1, VM_2, \dots, VM_n\}$ be the set of virtual machines distributed on several hosts in a data center, which should process *m* tasks T = $\{T_1, T_2, \dots, T_m\}$. We need to schedule tasks to these VMs, taking into consideration the capacity of each VM in terms of its available memory and execution speed i.e. Million Instructions per Second (MIPS). The parameters on which we can judge the scheduling algorithm are response/turnaround time and cost. Let's study a few load balancing algorithms:

*Round-Robin*. VMs are allotted tasks in a round-robin manner i.e. task1 to VM1, task2 to VM2, and so on.

*First-Come-First-Serve*. Load Balancer maintains a queue of the tasks as they arrive at the data center, and assigns them to a node in the order the nodes become available.

*Equally Spread Current Execution Load* [16]. This algorithm attempts to maintain equal workload on each VM by allotting task to least loaded VM.

*Throttled* [16]. An index table of VMs is maintained. The table is parsed from the top for each request and the first available VM is allotted. This algorithm ensures that only a pre-defined number of internet cloudlets (grouping of user requests) are allocated to a single VM at any given time. If more cloudlets are present than the number of available VM's at a data center, some of the requests will have to be queued until the next VM becomes available.

*Min-Min* [17]. Each job is assigned to the resource which can complete it the earliest in order to spend less time completing all jobs.

*Max-Min* [17]. Similar to Min-min scheduling algorithm except that it gives the highest priority to the job with the maximum earliest completion time.

*Central Load Balancing* [18]. CLBDM algorithm handles the problem of overloaded servers in round robin algorithm, by providing a central load balancer which passes the load to other server if the response time of a server increases.

*Opportunistic Load Balancing* [19]. OLB assigns VMs to tasks in a random order.

All these algorithms are static where the decision related to distribution of load is made at compile time when resource requirements are estimated. These algorithms are not efficient as they can easily lead to overload of VMs.

In dynamic algorithms, the decision related to distribution of load is made at run-time, and the load can be shifted to another VM if needed.

*Weighted least connection* [20]. WLC algorithm maintains a weighted list of VMs with the no of connections and forward a new connection to the server based on its weight and number of connections.

*Exponential Smooth Forecast based on Weighted Least Connection* [21]. ESBWLC was presented in which selection of VM is based on parameters like CPU power, memory, performance etc.

*Load Balance Min-Min* [22]. The aim of this algorithm is to minimize the make-span (maximum of the completion times of all the jobs scheduled on their respective VM). In this algorithm, min-min is executed and makespan is calculated. Then the node with highest makespan value is selected, and the minimum execution time job on this node is selected. The completion time for the selected job is calculated on all the VMs. Maximum completion time of the selected job is compared with the makespan value. If it is less, then the selected job is allocated to the VM. Else, the next maximum completion time of the job is selected and the steps are repeated.

*Biased Random Sampling* [23]. In Biased Random Sampling, the network is represented in the form of a virtual graph. Each server is symbolized as a node in the graph, with each

in-degree of the node representing the free resources of the server. The increment and decrement of node's in-degree is performed via Biased Random Sampling. Random sampling can be defined as the process wherein the servers are randomly selected. The sampling begins at some fixed node, and at each step, it moves to an adjacent node, chosen randomly. In-degree refers to the free resources of the node. When a node is allocated a new job, it removes one edge to decrease its in-degree. Similarly, when a node completes a job, it will add an edge to itself to increase its in-degree.

A few soft computing techniques like Genetic Algorithm [24], Ant Colony [25], Particle Swarm [26], Honey Bee Foraging [27] are also reported in literature.

## Summary

In a cloud computing environment the aim is to fully utilize a host by using virtual machines. The two main tasks in cloud computing are VM allocation and task scheduling. The aim is to use minimal number of hosts, efficient load balancing (dynamic), low response/turnaround time, and low power consumption.

## Future Work

In this paper, all VMs are present in a single data center. Hence we are not considering the factors like network/internet bandwidth while calculating delay in response time (according to distance b/w user and the data center).

## References

[1] I. Foster, Y. Zhao, I. Raicu and S. Lu, Cloud computing and grid computing: 360-degree compared, in: IEEE Grid Computing Environment Workshops, 2008.

[2] Q. Zhang, L. Cheng and R. Boutaba, Cloud computing: State-of-the-art and research challenges, in: Journal of Internet Services and Applications, 2010.
[3] OpenStack: An Overview. Retrieved from www.openstack.org/downloads/ openstack-overview-data-sheet.pdf
[4] N. Bobroff, A. Kochut, K. Beaty, Dynamic placement of virtual machines for managing sla violations, in: Proceedings of the 10th IEEE Symposium on Integrated Management, 2007, pp. 119-128.
[5] S. Srikantaiah, A. Kansal, F. Zhao, Energy aware consolidation for cloud computing, in: Proceedings of HotPower'08 Workshop on Power Aware Computing and Systems, 2008.
[6] A. Verma, P. Ahuja, A. Neogi, pMapper: power and migration cost aware application placement in virtualized systems, in: Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware, 2008, pp. 243-264.
[7] B. Li, J. Li, J. Huai, T. Wo, Q.Li, L. Zhong, EnaCloud: an energy-saving application live placement approach for cloud computing environments, in: Proceedings of the IEEE International Conference on Cloud Computing, 2009, pp. 17-24.
[8] E. Feller, L. Rilling, C. Morin, Energy-aware ant colony based workload placement in clouds, in: Proceedings of the IEEE/ACM International Conference on Grid Computing, 2011, pp. 26-33.
[9] H. Van, F. Tran, J. Menaud, Performance and power management for cloud infrastructures, in: Proceedings of the IEEE 3rd International Conference on Cloud Computing, 2010, pp. 329-336.
[10] F. Hermenier, X. Lorca, J. Menaud, G. Mueller, J. Lawall, Entropy: a consolidation manager for clusters, in: Proceedings of the 2009 ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments, 2009, pp. 41-50.
[11] M. Bichler, T. Setzer, B. Speitkamp, Capacity planning for virtualized servers, in: Workshop on Information Technologies and Systems, 2006.
[12] S. Chaisiri, B. Lee, D. Niyato, Optimal virtual machine placement across multiple cloud providers, in: Proceedings of the IEEE Asia-Pacific Services Computing Conference, 2009, pp. 103-110.
[13] B. Speitkamp, M. Bichler, A mathematical programming approach for server consolidation problems in virtualized data centers, IEEE Trans. Services Comput., 2010, 266-278.
[14] H. Mi, H. Wang, G. Yin, Y. Zhou, D. Shi, L. Yuan, Online self-reconfiguration with performance guarantee for energy-efficient large-scale cloud computing data centers, in: Proceedings of the IEEE International Conference on Services Computing, 2010, pp. 514-521.
[15] J. Xu, J. Fortes, Multi-objective virtual machine placement in virtualized data center environments, in: Proceedings of the IEEE/ACM International Conference on Cyber, Physical, and Social Computing, 2010, pp. 179-188.
[16] B. Wickremasinghe, CloudAnalyst: A CloudSim-based Tool for Modelling and Analysis of Large Scale Cloud Computing Environments, MEDC Project Report, CSE dept., University of Melbourne, 2009.
[17] K. Etminani, M. Naghibzadeh, A Min-Min Max-Min Selective Algorithm for Grid Task Scheduling, in: The Third IEEE / IFIP International Conference on Internet (ICI 2007), September 26 - 28, 2007, Tashkent, Uzbekistan.
[18] B. Radojevic and M. Zagar, Analysis of issues with load balancing algorithms in hosted (cloud) environments, in: 34th IEEE International Convention on MIPRO, 2011.
[19] M. Maheswaran, S. Ali, H. J. Siegel, D. Hensgen and R. F. Freund, Dynamic Matching and Scheduling of a Class of Independent Tasks onto Heterogeneous Computing Systems, in: Journal of Parallel and Distributed Computing, vol. 59, no. 2, November 1999, pp.107-131.
[20] Zheng Qi.,et al, Load balancing algorithm based on dynamic feedback, in: Computer Age, pp: 49-51,2006.
[21] Ren, X., R. Lin and H. Zou, A Dynamic Load Balancing Strategy for Cloud Computing Platform Based on Exponential Smoothing Forecast, in: International Conference on Cloud Computing and Intelligent Systems (CCIS), IEEE, pp: 220-224, September 2011.
[22] O.A. Rahmeh, P. Johnson, A. Taleb-Bendiab, A Dynamic Biased Random Sampling Scheme for scalable and reliable Grid Networks, in: The INFOCOMP Journal of Computer Science, vol. 7, 1-10.
[23] C. Zhao, S. Zhang, Q. Liu, J. Xie, and J. Hu, Independent Tasks Scheduling Based on Genetic Algorithm in Cloud Computing, in: Wireless Communications, Networking and Mobile Computing, 2009. WiCom '09.
[24] K. Li, G. Xu, G. Zhao, Y. Dong and D. Wang, Cloud Task Scheduling Based on Load Balancing Ant Colony Optimization, in: Chinagrid Conference (ChinaGrid), 2011.
[25] V. Sesum-Cavic, E. Kuhn, Applying Swarm Intelligence Algorithms for Dynamic Load Balancing to a Cloud Based Call Center, in: 4th IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO), 2010 .