International Journal of Advance Research in Science and Engineering
Vol. No.6, Issue No. 06, June 2017
www.ijarse.com
IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

A New Approach Towards Virtual Machine Scheduling Evaluation of Three Different VM Scheduling Algorithms

Sharon Justine[1], Deepa S Kumar[2]

[1]*Computer Science and Engineering, College of Engineering Munnar, (India)*

ABSTRACT

Scheduling is a balancing scenario in which processes or tasks are scheduled as per given requirements. It is a method that is used to distribute valuable computing resources, usually processor time, bandwidth and memory, to the various processes, threads, data flows and applications that need them. Scheduling is done to balance the load on the system and ensure equal distribution of resources and give some prioritization according to set of rules. This ensures that a computer system is able to serve all requests and achieve a certain quality of service. Virtual Machine (VM) scheduling algorithms are used to schedule the VM requests to the Physical Machines (PM) of the particular Data Center (DC) as per the requirement fulfilled with the requested resources.VM Scheduling determines how many processing cores of a host are allocated to virtual machines and how many processing cores will be delegated to each VM. It also determine how much of the processing core's capacity will effectively be attributed for a given VM. Cloud Simulation is a situation in which a particular set of condition is created artificially in order to study that could exit in reality. CloudSim is a cloud simulation tool. It is a library for the simulation of cloud scenarios. It provides essential classes for describing data centers, computational resources, virtual machines, applications, users, and policies for the management of various parts of the system such as scheduling and provisioning. Using these components, it is easy to evaluate new strategies governing the use of clouds, while considering policies, scheduling algorithms, load balancing policies, etc. This project is intended to evaluate three VM scheduling algorithms viz. First-Come First-Serve VM Scheduling algorithm, Round Robin VM Scheduling algorithm and Genetic VM Scheduling algorithm using CloudSim and throughput calculation is made for the above mentioned algorithms and trying to find out maximum throughput for the VM scheduling algorithms. In this project three VM scheduling algorithm are implemented and throughput calculation is made, best algorithm is found on the basis of throughput. Comparing the three algorithms, experiments show that Genetic algorithm has maximum throughput.

Keywords* : *CloudSim, First-Come First-Serve VM Scheduling Algorithm, Genetic VM Scheduling Algorithm, Round Robin VM Scheduling Algorithm, Virtual Machine Scheduling

I. INTRODUCTION

Cloud computing is a pay per use model for providing convenient and on demand network access to sharable and configurable computing resources like networks, servers and applications that can be easily managed with minimal effort or service provider interaction.

Virtualization is a way of abstracting the physical resources to improve their utilization. It logically divides the resources between different users. A virtual machine is a logical implementation of a computer that operates in

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

the similar manner as the computer system. A cloud computing environment requires a suitable algorithm for executing the various jobs provided to the system in a cost effective manner based on certain constraints. This task is performed by a VM scheduler using a suitable scheduling algorithm VM scheduling policy is implemented in two levels in CloudSim namely Host level and VM level. At the host level, it is possible to specify how much of the overall processing power of each core in a host will be assigned to each VM. At the VM level, the Virtual Machines assign specific amount of the available processing power to the individual task units that are hosted within its execution engine. In Host level allocation the available VM's are allotted to the free PE (CPU) or hosts and in VM level allocation the available cloudlets are allotted to the free VM's.

There are two types of policies based on which scheduling is done in CloudSim – Space Shared scheduling policy and Time Shared scheduling policy. In Space Shared scheduling policy for Host level one VM is assigned at a time to a CPU core, when this VM finishes its task then it schedules another VM to a CPU core. In Space Shared scheduling policy for VM level one task is scheduled at a time to a virtual machine, when this task is completed it schedules another task to the virtual machine [1].

CloudSim provides a generalised and extensible simulation framework that enables seamless modeling and simulation of app performance. By using CloudSim, developers can focus on specific systems design issues that they want to investigate, without getting concerned about details related to cloud-based infrastructures and services.

II. CLOUDSIM

CloudSim provides a generalised and extensible simulation framework that enables seamless modelling and simulation of app performance. By using CloudSim, developers can focus on specific systems design issues that they want to investigate, without getting concerned about details related to cloud-based infrastructures and services.

CloudSim, which is -a toolkit for the modelling and simulation of Cloud computing environments- comes to the rescue. It provides system and behavioural modelling of the Cloud computing components. Simulation of cloud environments and applications to evaluate performance can provide useful insights to explore such dynamic, massively distributed, and scalable environments.

CloudSim is a simulation tool that allows cloud developers to test the performance of their provisioning policies in a repeatable and controllable environment, free of cost. It helps tune the bottlenecks before real-world deployment. It is a simulator; hence, it doesn't run any actual software. It can be defined as 'running a model of an environment in a model of hardware', where technology-specific details are abstracted.

CloudSim is a library for the simulation of cloud scenarios. It provides essential classes for describing data centres, computational resources, virtual machines, applications, users, and policies for the management of various parts of the system such as scheduling and provisioning. Using these components, it is easy to evaluate new strategies governing the use of clouds, while considering policies, scheduling algorithms, load balancing policies, etc. It can also be used to assess the competence of strategies from various perspectives such as cost, application execution time, etc. It also supports the evaluation of Green IT policies. It can be used as a building block for a simulated cloud environment and can add new policies for scheduling, load balancing and new scenarios. It is flexible enough to be used as a library that allows you to add a desired scenario by writing a Java

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

program. By using CloudSim, organisations, R&D centres and industry-based developers can test the performance of a newly developed application in a controlled and easy to set-up environment.

2.1 Scheduling in CloudSim

CloudSim is a simulating tool that has been widely used for simulating cloud applications and related algorithms. It is an open source tool which is free to use. In CloudSim, scheduling has been performed at two different levels, viz. First between the Hosts and Virtual Machines for allocation of Processing Elements (PE's) and second between the Virtual Machines and the Cloudlets, i.e., applications or processes to be executed over cloud. Commonly, the first is known as VM Scheduling and the second is known as Cloudlet Scheduling. Basically supports two types of scheduling techniques at both levels, viz. Time Shared Scheduling and Space Shared Scheduling. CloudSim also supports another type of VM Scheduling known as Time Shared over Subscription VM Scheduling and another type of Cloudlet Scheduling known as Dynamic Workload Cloudlet Scheduling [1].

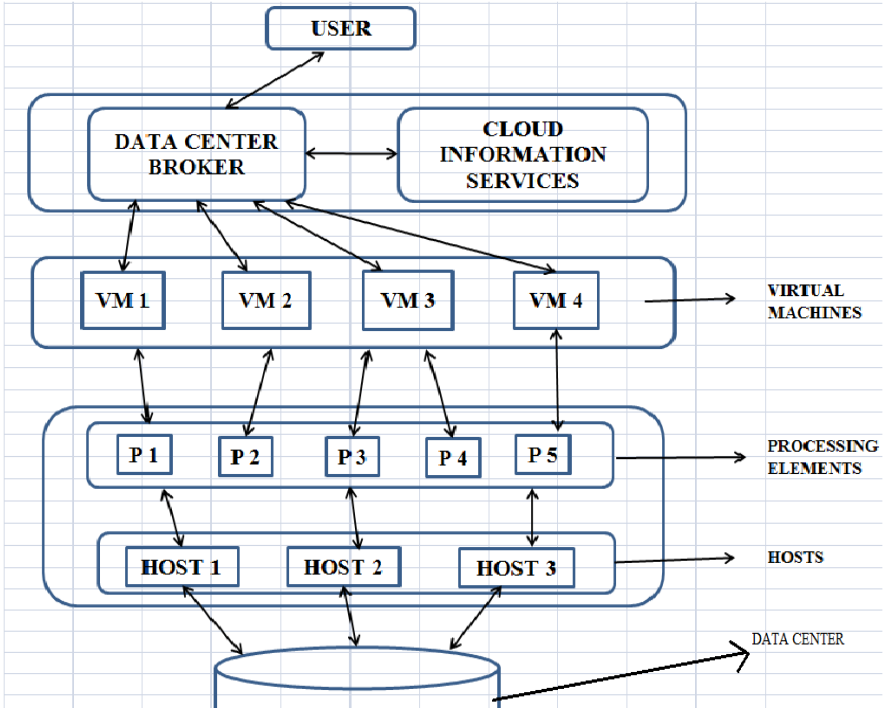

Fig.2.1 Vm scheduling in cloud computing environment

In Space Shared Scheduling at any level, the sharing of Processing Elements between tasks or VM's is not possible. A PE can be allotted to new VM or a new task can be taken for execution only after completion of the previous task or successful execution of all tasks b a VM. While in Time Shared Scheduling, tasks can share the PE or the VM's can also share the PE's for execution. Time Shared with Over subscription VM Scheduling is basically used to improve the utilization of the cloud resources to a higher extent like improving the utilization of memory, PE's, etc. Dynamic Workload Cloudlet Scheduling is commonly employed in cases when there is only one cloudlet for execution that can be used to execute as an online web service[13].

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

III. VM SCHEDULING ALGORITHMS

VM Scheduling determines how many processing cores of a host are allocated to virtual machines and how many processing cores will be delegated to each VM. It also determine how much of the processing core's capacity will effectively be attributed for a given VM.

There are different VM scheduling policies are there viz. First-Come First-Serve(FCFS), Shortest Job First, Round Robin, Genetic Algorithm etc. VM scheduling policies like FCFS, Round Robin and Genetic Algorithm are evaluated in this project.

First-Come First-Serve is for parallel processing and targets the resource having the least waiting line up time and is chosen for the received job. It is the simplest VM scheduling algorithm. It is a non-preemptive scheduling algorithm.

The Round Robin algorithm focuses on distributing the load equally to all the nodes. Using this algorithm, the Scheduler allocates one VM to a node in a cyclic manner. The main advantage of this algorithm is that it utilizes all the resources in a balanced order [2].

Genetic algorithms are stochastic search algorithms based on the mechanism of natural selection strategy. It starts with a set of initial solution, called initial population, and will generate new solution using genetic operators. The advantage of this technique is it can handle a large searching space, applicable to complex objective function and can avoid trapping by local optimum solution [3].

3.1 First- Come First- Serve VM Scheduling Algorithm

First-come First-serve (FCFS) scheduling algorithm in which the process that requests the resource first is allocated first. The implementation of the FCFS policy is easily managed with a FIFO queue. When a process enters the ready queue, its PCB is linked onto the tail of the queue. When the CPU is free, it is allocated to the process at the head of the queue. The running process is then removed from the queue. The code for FCFS scheduling is simple to write and understand.

It is for parallel processing and targets the resource having the least waiting line up time and is chosen for the received job. The CloudSim toolkit backs FCFS scheduling plan for interior scheduling tasks. Distribution of app-specified VMs to Hosts within a Cloud-based datacenter is the work of the virtual machine stipulated element. The default policy adopted by the VM stipulated is a simple policy that distributes a VM in FCFS method. The limitations of First come first serve is that it is non preemptive. The shortest errands which are based at the back of the line-up must wait for long errands at the front to complete. It is turn around and reaction is fairly minimal.

The FCFS scheduling algorithm is non-preemptive. Once the CPU has been allocated to a process, that process keeps the CPU until it releases the CPU, either by terminating or by requesting I/O. The FCFS algorithm is thus particularly troublesome for time-sharing systems, where it is important that each user get a share of the CPU at regular intervals. It would be disastrous to allow one process to keep the CPU for an extended period.

3.2 Round Robin VM Scheduling Algorithm

The Round Robin algorithm focuses on distributing the load equally to all the nodes. Using this algorithm, the Scheduler allocates one VM to a node in a cyclic manner. The round robin scheduling for VM scheduling is similar to the round robin scheduling for process scheduling. The scheduler starts assigning VM to each node and move further for next VM to place in to next node. This algorithm is repeated for all the nodes until each

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

node have at least on VM. After placing VMs to all nodes it will go to the first node and repeat this process for next VMs requests. Hence, in this case, the scheduler does not wait for the exhaustion of the resources of a node before moving on to the next. As an example, if there are three nodes and three VMs are to be scheduled, each node would be allocated one VM, provided all the nodes have enough available resources to run the VMs.

The main advantage of this algorithm is that it utilizes all the resources in a balanced order. An equal number of VMs are allocated to all the nodes which ensure fairness. However, the major drawback of using this algorithm is that the power consumption will be high as many nodes will be kept turned on for a long time. If three resources can be run on a single node, all the three nodes will be turned on when Round Robin is used which will consume a significant amount of power. The CloudSim toolkit supports RR scheduling strategy for internal scheduling of jobs [14].

3.3 Genetic VM Scheduling Algorithm

Genetic algorithm is for scheduling sets of independent VM's, the objective of genetic algorithm is to minimize the make span. Initially in GA many individual solutions are (usually) randomly generated to form an initial population. The population size depends on the nature of the problem i.e. type and no of VM's to be run effectively on system. During each successive generation, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through a fitness-based process, where fitter solutions (VM's schedule likely to give effective response time) are typically more likely to be selected. The next step is to generate a second generation population of solutions from those selected through genetic operators: crossover and mutation. This generational process is repeated until a termination condition has been reached i.e. a solution is found that satisfies minimum response time criteria.

GA will increase the cost of time, space, throughput and improve the quality of service of the entire. The goal of GA is to reduce the scheduled time of VM. Genetic algorithm provides both improved response time to VM via parallel execution. A state of the system and through genetic algorithm the migration cost becomes a problem [15].

IV. RESULT

Throughput is the number of processes that complete their execution per time unit. Throughput is a measure of how many units of information a system can process in a given amount of time. It is applied broadly to systems ranging from various aspects of computer and network systems to organizations. Throughput calculation for the VM scheduling is calculated using the formula given below:

Throughput = Number o f VMs/ Total Excecution Time (1)

Algorithm with maximum throughput is considered as the better algorithm. For the analysis of algorithms like First-Come First-Served, Round Robin and Genetic algorithm throughput is calculated.

Throughput of First-Come First-Served Algorithm= 0.56

Throughput of Round Robin Algorithm= 0.015

Throughput of Genetic Algorithm= 1.5

Hence concluded that genetic algorithm has better performance among the three.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

V. CONCLUSION

Virtual Machine Scheduling determines how many processing cores of a host are allocated to virtual machines and how many processing cores will be delegated to each VM. Virtual machine scheduling algorithms like First-Come First-Serve, Round Robin and genetic algorithm are implemented in cloud simulation environment using the simulation tool CloudSim. Throughput calculation is made for each above mentioned algorithm. Genetic algorithm posses the higher throughput value. The higher throughput value algorithm is the better one. Hence came in a conclusion that Genetic algorithm is the best scheduling algorithm among the three.

REFERENCES

[1] S. Santra and K. Mali, "A new approach to survey on load balancing in vm in cloud computing: Using cloudsim," in Computer, Communication and Control (IC4), 2015 International Conference on. IEEE, 2015, pp. 1-5.

[2] J. T. Piao and J. Yan, "A network-aware virtual machine placement and migration approach in cloud computing," in Grid and Cooperative Computing (GCC), 2010 9th International Conference on. IEEE, 2010, pp. 87-92.

[3] K. D. Prajapati, P. Raval, M. Karamta, and M. Potdar, "Comparison of virtual machine scheduling algorithms in cloud computing," International Journal of Computer Applications, vol. 83, no. 15, 2013.

[4] S. S. Masoumzadeh and H. Hlavacs, "Integrating vm selection criteria in distributed dynamic vm consolidation using fuzzy q-learning," in Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013). IEEE, 2013, pp. 332-338.

[5] H. Teyeb, A. Balma, N. B. Hadj-Alouane, S. Tata, and A. B. Hadj-Alouane, "Traffic aware virtual machine placement in geographically distributed clouds," in Control, Decision and Information Technologies (CoDIT), 2014 International Conference on. IEEE, 2014, pp. 024-029.

[6] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. De Rose, and R. Buyya, "Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Software: Practice and Experience, vol. 41, no. 1, pp. 23-50, 2011.

[7] Z. T. He, X. Q. Zhang, H. X. Zhang, and Z. W. Xu, "Study on new task schedulingstrategy in cloud computing environment based on the simulator cloudsim," in Advanced Materials Research, vol. 651. Trans Tech Publ, 2013, pp. 829-834.

[8] P. Humane and J. Varshapriya, "Simulation of cloud infrastructure using cloudsim simulator: A practical approach for researchers," in Smart Technologies and Management for Computing, Communication, Controls, Energy and Materials (ICSTM), 2015 International Conference on. IEEE, 2015, pp. 207-211.

[9] S. Long and Y. Zhao, "A toolkit for modeling and simulating cloud data storage: An extension to cloudsim," in Control Engineering and Communication Technology (ICCECT), 2012 International Conference on. IEEE, 2012, pp. 597-600.

IJARSE
ISSN (O) 2319 - 8354
ISSN (P) 2319 - 8346

[10] B. Wickremasinghe, R. N. Calheiros, and R. Buyya, "Cloudanalyst: A cloudsim-based visual modeller for analysing cloud computing environments and applications," in 2010 24th IEEE International Conference on Advanced Information Networking and Applications. IEEE, 2010, pp. 446-452.

[11] D. Kliazovich, P. Bouvry, and S. U. Khan, "Greencloud: a packet-level simulator of energy-aware cloud computing data centers," The Journal of Supercomputing, vol. 62, no. 3, pp. 1263-1283, 2012.

[12] B. Madhusudhan and K. C. Sekaran, "A genetic algorithm approach for virtual machine placement in cloud," in Published in the proceeding of International Conference on Emerging Research in Computing, Information, Communication and Applications (ERCICA2013), 2013.

[13] H. Casanova, "Simgrid: A toolkit for the simulation of application scheduling," in Cluster computing and the grid, 2001. proceedings. first ieee/acm international symposium on. IEEE, 2001, pp. 430-437.

[14] P. Himthani, A. Saxena, and M. Manoria, "Comparative analysis of vm scheduling algorithms in cloud environment," International Journal of Computer Applications, vol. 120, no. 6, 2015.

[15] A. Pawar and D. Kapgate, "A review on virtual machine scheduling in cloud computing," International Journal of Computer Science and Mobile Computing, vol. 3, no. 4, pp. 928-933, 2014.

[16] R. Benali, H. Teyeb, A. Balma, S. Tata, and N. B. Hadj-Alouane, "Evaluation of traffic-aware vm placement policies in distributed cloud using cloudsim," in Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE), 2016 IEEE 25th International Conference on. IEEE, 2016, pp. 95-100.

[17] A. Bosilca, M.-C. Nita, F. Pop, and V. Cristea, "Cloud simulation under fault constraints," in Intelligent Computer Communication and Processing (ICCP), 2014 IEEE International Conference on. IEEE, 2014, pp. 341-348.

[18] W. A. Higashino, M. A. Capretz, and L. F. Bittencourt, "Cepsim: A simulator for cloud-based complex event processing," in Big Data (BigData Congress), 2015 IEEE International Congress on. IEEE, 2015, pp. 182-190.

[19] S. Swathi, S. Saravanan, and V. Venkatachalam, "Preemptive virtual machine scheduling using cloudsim tool," Int. J. Adv. Eng, vol. 1, no. 3, pp. 323-327, 2015.

[20] R. Buyya, R. Ranjan, and R. N. Calheiros, "Modeling and simulation of scalable cloud computing environments and the cloudsim toolkit: Challenges and opportunities," in High Performance Computing & Simulation, 2009. HPCS'09. International Conference on. IEEE, 2009, pp. 1-11.