EFFICIENT VM LOAD BALANCING ALGORITHM FOR A CLOUD COMPUTING ENVIRONMENT

Jasmin James,
38 Sector-A,
Ambedkar Colony, Govindpura,
Bhopal M.P
Email:james.jasmin18@gmail.com

Dr. Bhupendra Verma,
Professor and Head Department of Computer Science & Engineering,
Director PG Course,
Technocrates Institute of Technology,
Bhopal, M.P., India

ABSTRACT

Cloud computing is a fast growing area in computing research and industry today. With the advancement of the Cloud, there are new possibilities opening up on how applications can be built and how different services can be offered to the end user through Virtualization, on the internet. There are the cloud service providers who provide large scaled computing infrastructure defined on usage, and provide the infrastructure services in a very flexible manner which the users can scale up or down at will.

The establishment of an effective load balancing algorithm and how to use Cloud computing resources efficiently for effective and efficient cloud computing is one of the Cloud computing service providers' ultimate goals.

In this paper firstly analysis of different Virtual Machine (VM) load balancing algorithms is done. Secondly, a new VM load balancing algorithm has been proposed and implemented for an IaaS framework in Simulated cloud computing environment; i.e. 'Weighted Active Monitoring Load Balancing Algorithm' using CloudSim tools, for the Datacenter to effectively load balance requests between the available virtual machines assigning a weight, in order to achieve better performance parameters such as response time and Data processing time.

Keywords

Cloudsim, DataCenterController, Virtualization, Virtual Machine, Load Balancing.

1.0 INTRODUCTION

The virtualization forms the foundation of cloud technology where Virtualization is an emerging IT paradigm that separates computing functions and technology implementations from physical hardware.
Cloud computing, for example, is the Virtualization of computer programs through an internet connection rather than installing applications on every office computer.

Using virtualization, users can access servers or storage without knowing specific server or storage details. The virtualization layer will execute user request for computing resources by accessing appropriate resources.

Virtualization can be applied to many types of computer resources: Infrastructure such as Storage, Network, Compute (CPU / Memory etc.), Platform (such as Linux/ Windows OS) and Software as Services.

Cloud computing in computing research and industry today has the potential to make the new idea of 'computing as a utility' in the near future. The Internet is often represented as a cloud and the term "Cloud Computing". Cloud computing is the dynamic provisioning of IT capabilities/IT services (hardware, software, or services) from third parties over a network [1][2][9]. These IT services are delivered on demand and they are delivered elastically, in terms of 'able to scale out' and 'scale in'. The sections below briefly details different types of cloud computing and how Virtual Machines (VMs) can be provided as cloud Infrastructure as a Service(Iaas).

2.0 CLOUD COMPUTING ENVIRONMENT

It is generally supposed that cloud computing can be classified into three basic types: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS) [2][3]. In IaaS grids or clusters, virtualized servers, its computational resources- CPUs, memory, networks, storage and systems software are delivered as a service. Perhaps the best known example is Amazon's Elastic Compute Cloud (EC2) and Simple Storage Service (S3) which provide (managed and scalable) resources as services to the user [2][4][8]. Platform as a Service (PaaS) typically makes use of dedicated APIs to control the behaviour of a server hosting engine which executes and replicates the execution according to user requests Eg. Force.com, Google App Engine. In Software as a Service (SaaS) standard application software functionality is offered within a cloud. Examples: Google Docs, SAP Business by design. Load balancing is one of prerequisites to utilize the full resources of parallel and distributed systems.

In IaaS, the physical resources can be split into a number of logical slices called Virtual Machines (VMs). All VM load balancing methods are designed to determine which Virtual Machine is assigned to the next cloudlet[4] task units. These VMs are modeled using different tools of Cloudsim-Simulation framework for its allocation to the application.

3.0 MODELING THE VM ALLOCATION [5][6]

Cloud computing infrastructure is the massive deployment of virtualization tools and techniques as it has an extra layer i.e. Virtualization layer that acts as an creation, execution, management, and hosting environment for application services.

The modeled VMs in the above virtual environment are contextually isolated but still they need to share computing resources- processing cores, system bus etc.

Hence, the amount of hardware resources available to each VM is constrained by the total processing power ie. CPU, the memory and system bandwidth available within the host. The choice of virtual machine, meaning that you can select a configuration of CPU, memory, storage, bandwidth etc. that is optimal for an application.

CloudSim supports VM provisioning at two levels:-

- At the host level – It is possible to specify how much of the overall processing power of each core will be assigned to each VM. Known as VM policy Allocation
- At the VM level – the VM assigns a fixed amount of the available processing power to the individual application services (task units) that are hosted within its execution engine. Known as VM Scheduling.
  Note that at each level CloudSim implements the time-shared and space-shared provisioning policies.

In this paper, we have proposed the VM load Balancing algorithm at the VM level (VM Scheduling-time shared) where, individual application services is assigned varying (different) amount of the available processing power of VMs This is because- in the real world, it's not necessary all the VMs in a DataCenter has fixed amount of processing powers but it can vary with different computing nodes at different ends.

And then to these VMs of different processing powers, the tasks/requests (application services) are assigned or allocated to the most powerful VM and then to the lowest and so on. They are given the required priority weights. Hence, the performance parameters such as overall response time and data processing time are optimized.

3.1 Simulation

Simulation is a technique where a program models the behavior of the system (CPU, network etc.) by calculating the interaction between its different entities using mathematical formulas', or actually capturing and playing back observations from a production system. The available Simulation tools in Cloud Computing today are: simjava, gridsim and CloudSim.

3.2 CloudSim [1][3][12]

CloudSim is a framework developed by the GRIDS laboratory of University of Melbourne which enables seamless modeling, simulation and experimenting on designing Cloud computing infrastructures. CloudSim is a self-contained platform which can be used to model data centers, host, service brokers, scheduling and allocation policies of a large scaled Cloud platform. This CloudSim framework is built on top of GridSim framework which is also developed by the GRIDS laboratory. Hence, the researcher has used CloudSim to model datacenters, hosts, VMs for experimenting in simulated cloud environment.

This paper introduces a new VM load balancing algorithm:"Weighted Active Monitoring Load Balancing Algorithm"; to handle service request from user base. [7]. Below- Section 4.0 introduces the Contemporary VM load balancers; and Section 5.0 include the purposed VM load balancing algorithm for better response time and data processing time of cloudlets and results in Section 6.0: Research Setup & Results.

4.0 CONTEMPORARY VM LOAD BALANCERS

Virtual machine enables the abstraction of an OS and Application running on it from the hardware. The interior hardware infrastructure services interrelated to the Clouds is modeled in the Cloudsim simulator by a Datacenter element for handling service requests. These requests are application elements sandboxed within VMs, which need to be allocated a share of processing power on Datacenter's host components. DataCenter object manages the data center management activities such as VM creation and destruction and does the routing of user requests received from User Bases via the Internet to the VMs. The Data Center Controller [7], uses a VmLoadBalancer to determine which VM should be assigned the next request for processing. The contemporary Vmloadbalancer are Round Robin, throttled and active monitoring load balancing algorithms.

A. Round Robin Load Balancer (RRLB)

In this, the datacenter controller assigns the requests to a list of VMs on a rotating basis. The first request is allocated to a VM- picked randomly from the group and then the DataCenter controller assigns the subsequent requests in a circular order. Once the VM is assigned the request, the VM is moved to the end of the list. In this RRLB; there is a better allocation concept known as **Weighted Round Robin Allocation** in which one can assign a weight to each VM so that if one VM is capable of handling twice as much load as the other, the powerful server gets a weight of 2. In such cases, the DataCenter Controller will assign two requests to the powerful VM for each request assigned to a weaker one.

The major issue in this allocation is this that it does not consider the advanced load balancing requirements such as processing times for each individual requests.[1]

B. Throttled Load Balancer (TLB)

The TLB maintains a record of the state of each virtual machine (busy/ideal). If a request arrived concerning the allocation of virtual machine, the TLB sends the ID of ideal virtual machine to the data center controller and data center controller allocates the ideal virtual machine.

C. Active Monitoring Load Balancer (AMLB)

THE AMLB maintains information about each VMs and the number of requests currently allocated to which VM. When a request to allocate a new VM arrives, it identifies the least loaded VM. If there are more than one, the first identified is selected. ActiveVmLoadBalancer returns the VM id to the Data Center Controller. The data Center Controller sends the request to the VM identified by that id. DataCenterController notifies the ActiveVmLoadBalancer of the new allocation and cloudlet is sent to it.

5.0 PROPOSED –WEIGHTED ACTIVE MONITORING LOAD BALANCING ALGORITHM

The ‘Weighted Active Monitoring Load Algorithm’ is implemented; modifying the Active Monitoring Load Balancer by assigning a weight to each VM as discussed in Weighted Round Robin Algorithm of cloud computing in order to achieve better response time and processing time.

In this proposed Load balancing algorithm using the concept of weights in active monitoring, the VM are assigned varying (different) amount of the available processing power of server/ physical host to the individual application services. To these VMs of different processing powers; the tasks/requests (application services) are assigned or allocated to the most powerful VM and then to the lowest and so on according to its weight and its availability. Hence optimizing the given performance parameters.

WEIGHTED ACTIVE MONITORING LOAD BALANCER

(Algorithm)

STEP 1: Create VM’s of different Datacenter according to computing power of host/physical server in terms of its core processor, processing speed, memory, storage etc.

STEP 2: Allocate weighted count according to the computing power of the VM’s in Datacenter. If one VM is capable of having twice as much load as the other, the powerful server gets a weight of ‘2’ or if it can take four times load then server gets a weight of ‘4’ and so on.

For example:

- Host server with single core processor, 1GB of memory, 1TB of Storage space, 1000000 bandwidth will have weighted count=1
- Host server with 2 core processor, 4GB of memory, 2TB of Storage space and 1000000 bandwidth will have weighted count=2
- Host server with quard core processor, 8GB of memory 4TB of Storage space and 1000000 bandwidth will have weighted count=4 and so on..

STEP 3: WeightedActiveVmLoadBalancer maintains an index table of VMs, associated weighted count and the number of requests currently allocated to the VM. At start all VM's have 0 allocations.

STEP 4: When a request to allocate a new VM from the DataCenterController arrives, it parses the table and identifies the least loaded VM.

STEP 5: After Identifying the least loaded VM's in different datacenters, it allocate requests to the most powerful VM according to the weight assigned. If there are more than one, the first identified is selected.

STEP 6: WeightedActiveVmLoadBalancer returns the VM id to the DataCenterController.

STEP 7: The DataCenterController sends the request to the VM identified by that id.

STEP 8: DataCenterController notifies the WeightedActiveVmLoadBalancer of the new allocation.

STEP 9: WeightedActiveVmLoadBalancer updates the allocation table increasing the allocations count for that VM.

STEP 10: When the VM finishes processing the request, and the DataCenterController receives the response cloudlet, it notifies the WeightedActiveVmLoadBalancer of the VM de-allocation.

STEP 11: The WeightedActiveVmLoadBalancer updates the allocation table by decreasing the allocation count for the VM by one.

STEP 12: Continue from step 4.

The purpose of algorithm is to find the expected Response Time of each Virtual Machine because virtual machine are of heterogeneous capacity with regard to its processing performance, the expected response time can be found with the help of the following formulas:

Response Time = Fin_t - Arr_t + TDelay (1) Where, Arr_t is the arrival time of user request and Fin_t is the finish time of user request and the transmission delay can be determined by using the following formulas:

TDelay = T + T(2)latencytransfer Where, TDelay is the transmission delay Tlatency is the network latency and T transfer is the time taken to transfer the size of data of a single request (D) from source location to destination.

Ttransfer = D / Bwperuser (3) Bwperuser = Bwtotal / Nr (4) Where, Bwtotal is the total available bandwidth and Nr is the number of user requests currently in transmission. The Internet Characteristics also keeps track of the number of user requests in-flight between two regions for the value of Nr.

6.0 RESEARCH SETUP & RESULTS

The proposed algorithm is implemented through simulation package CloudSim based tool [7][10][11]. Java language is used for develop and implement the new 'Weighted VM load balancing Algorithm'. Assuming the application is deployed in one data center having 3 virtual machines running on each physical hosts (3 in numbers); then the **Parameter Values are as under:**

Table 1: Parameter Value Simulation duration: 60 min.

PARAMETER	VALUE
Data Center OS	Linux
Data Center Architecture	X86
Service Broker Policy	Optimize Response Time
Physical H/w units (physical hosts)	3
No. of VMs	3

Each physical hosts has 3 number of VMs, having configuration as:

Id	Memory (Mb)	Storage (Mb)	Available BW (Mb)
0	1024	1048576	1000000
1	2048	2097152	1000000
2	4096	4194304	1000000

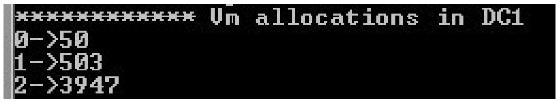
No.of processors	Processor Speed (MIPS)	VM Policy
1	10000	TIME-SHARED
2	20000	TIME-SHARED
4	40000	TIME-SHARED

Followings are the experimental results based on Efficient Weighted Active VM Load Balancing Algorithm:

Table 2: Result Detail 1

Performance Parameters	**Avg (ms)**	**Min (ms)**	**Max (ms)**
Overall Response Time	694.82	40.31	1222.07
Data Processing Time	0.18	0.01	0.81

Result Detail 2



```
************ Vm allocations in DC1
0->50
1->503
2->3947
```

VMs of physical host Id-2 is most powerful than Id- 0 and Id-1. So no. of requests allotted to that host (set of VMs) is given the first priority with the highest weight has the highest as response and thus showing less 'Over all Response Time' 694.82 ms and 'Data Processing Time' as 0.18ms.

7.0 CONCLUSION

In this paper a new VM Load Balancing Algorithm is proposed and then implemented in Cloud Computing environment using CloudSim toolkit, in java language. In this algorithm, the VM assigns a varying (different) amount of the available processing power to the individual application services. These VMs of different processing powers, the tasks/requests (application services) are assigned or allocated to the most powerful VM and then to the lowest and so on. Hence we have optimized the given performance parameters such as response time and data processing time, giving an efficient VM Load Balancing algorithm i.e. 'Weighted Active Load Balancing Algorithm' in the Cloud Computing environment.

References /Bibliography

[1] Qi Zhang, Lu Cheng, Raouf Boutaba; Cloud computing: sate-of-art- and research challenges; Published online: 20th April 2010, Copyright : The Brazillian Computer Society 2010.

[2] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D.Joseph, Randy Katz; 'Above the Clouds: A Berkeley View of Cloud Computing'; The Regents of the University of California, 2009

[3] Christ of Weinhardt, Benjamin Blau, Jochen Stober; 'Cloud Computing- A Classification, Business Models, And Research Directions', Business and Information System Engineering'. Vol 5. 2009, Pg 391-399

[4] Rodrigo N.Calheiros, Rajiv Ranjan, Cesar A. F. De Rose, and Rajkumar Buyya; CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services, Grid Computing and Distributed Systems (GRIDS) Laboratory, Dept. of Computer Science and Software Engineering, The University of Melbourne, Austrailia; Pontificial Catholic University of Rio Grande do Sul Porto Alegre, Brazil. {Rodrigo, rranjan, raj} @csse.unimelb.edu.au, cesar.derose@pucrs.br

[5] Rodrigo N. Calheiros, Rajiv 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, Software: Practice and Experience (SPE), Volume 41, Number 1, Pages: 23-50, ISSN: 0038-0644, Wiley Press, New York, USA, January, 2011

[6] Rajkumar Buyya, Rajiv Ranjan and Rodrigo N. Calheiros,; Modeling and Simulation of Scalable Cloud Computing Environments and the CloudSim Toolkit: Challenges and Opportunities. Proceedings of the 7th High Performance Computing and Simulation Conference (HPCS 2009, ISBN: 978-1-4244-4907-1, IEEE Press, New York, USA), Leipzig, Germany, June 21-24, 2009.

[7] Bhathiya Wickremasinghe, Rodrigo N. Calheiros, Rajkumar Buyya,"CloudAnalyst: A CloudSim-based Visual Modeller for Analysing Cloud Computing Environments and Applications", 20-23, April 2010, pp. 446-452.

[8] Cloud computing insights from 110 implementation projects; IBM Academy of TechnologyThought Leadership White Paper, October 2010.

[9] IoannisPsoroulas,IoannisAnagnostopoulos,VassiliLoumos, Eleftherios Kayafas, "A Study of the Parameters Concerning Load Balancing Algorithms", IJCSNS International Journal of Computer Science and Network Security, Vol. 7, No. 4, 2007, pp. 202-214 .

[10] Sandeep Sharma, Sarabjit Singh, Meenakshi Sharma "Performance Analysis of Load Balancing Algorithms", World Academy of Science, Engineering and Technology, 38, 2008 pp. 269- 272.

[11] CloudSim 2.1.1 API (Application Programming Interface) , The Cloud Computing and Distributed Systems (CLOUDS) Laboratory, The University ofMelbourne, http://www.cloudbus.org/cloudsim/doc/api/index.html

Dr. Bhupendra Verma, M.Tech, Ph.D. in Computer Science and Engineering,.Presently working as Professor & Head, Department of Computer Science & Engineering, Director PG Course, Technocrates Institute of Technology, Bhopal, M.P., India.

M/s. Jasmin James, B.E.,R.G.T.U., Bhopal, M.P.,India. Presently pursuing dissertation of M.Tech.C.S.E, under the supervision of Dr. Bhupendra Verma, Technocrates Institute of Technology, Bhopal, M.P. India. Email: james.jasmin18@gmail.com