Effective Virtual Machine Load Balancing Algorithm for Cloud Computing

Dr Suresh KOPPARTHI, Ms GODI SUSHMA, Mr ABHISHEK GAJULA

Professor and principal,[1] Asst. Professor[2,3]
Dept. of Computer Science and Engineeering,
Bhimavaram Institute of Engineering and Technology, Bhimavaram, Andhra Predesh, India.
E.mail id: sureshkgrl@gmail.com, sushma.godi@gmail.com, abhishek.gajula@gmail.com

ABSTRACT
The field of "cloud computing" is expanding rapidly in both academia and business. As the Cloud matures, it paves the way for novel approaches to application development and the delivery of a variety of services to end users through virtualization on the web. Cloud service providers provide scalable computing resources based on demand, with the added benefit of allowing their customers to easily expand or contract their resources as needed.
One of the ultimate aims of Cloud computing service providers is the construction of an effective load balancing algorithm and the learning of how to utilize Cloud computing resources properly for effective and efficient cloud computing.
First, this study compares and contrasts several load-balancing techniques for use with Virtual Machines (VMs). Second, in order to improve performance parameters like response time and Data processing time, a new VM load balancing algorithm called "Weighted Active Monitoring Load Balancing Algorithm" has been proposed and implemented for an IaaS framework in a simulated cloud computing environment using CloudSim tools.

Keywords

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

1.0 INTRODUCTION
Cloud computing relies on virtualization, a relatively new information technology paradigm that decouples software and operating system deployments from underlying hardware.
Instead of installing software on each workplace computer, cloud computing allows for the virtualization of software over an internet connection.
With virtualization, users don't need to know the specifics of a server or a storage system in order to access it. The virtualization layer will access the necessary hardware to carry out the user's request for computational resources.

Storage, networks, computation (CPU, RAM, etc.), platforms (Linux, Windows, etc.), and software as a service are all examples of computer resources that may be virtualized.
The widespread use of cloud computing in both academia and business might usher in a new era of "computing as a utility" in the not-too-distant future. The cloud metaphor and "Cloud Computing" are often used to describe the Internet. Cloud computing is the on-demand delivery of resources (either hardware, software, or services) to a user through the Internet.[2][9]. These IT services are offered both on-demand and elastically, meaning they may scale both up and down depending on demand. Cloud computing and the provision of Virtual Machines (VMs) as cloud Infrastructure as a Service (IaaS) are briefly discussed in the following sections.
2.0 INTERNET OF CLOUD COMPUTERS
Infrastructure as a service (IaaS), platform as a service (PaaS), and software as a service (SaaS) are the three main categories into which cloud computing is often divided. [2][3]. The central processing units, random access memories, networks, storage media, and operating systems software of a server are all virtualized in IaaS grids or clusters.

supply as a service. Amazon's Elastic Compute Cloud (EC2) and Simple Storage Service (S3) are two of the most well-known examples of such services, since they give (controlled and scalable) resources to the customer [2].[4][8]. PaaS often use application programming interfaces (APIs) to manage the actions of a server-hosted engine that performs and repeats the execution based on user requests. Force.com, Google App Engine, etc. SaaS refers to a model wherein commonly used program features are made available via the Internet. Google Docs and SAP's Business by Design are only two examples. Load balancing is a necessary component of parallel and distributed system architectures.
With Infrastructure as a Service (IaaS), real-world hardware may be partitioned into many virtual machines (VMs).

To decide which Virtual Machine should do the next cloudlet[4] operation, all load balancing techniques are created equal. These virtual machines are simulated using various Cloudsim-Simulation framework tools before being allotted to the program.

3.0 MODELLING VMWARE RESOURCE ALLOCATION [5][6]

The virtualization layer of a cloud computing infrastructure is responsible for providing a sandboxed setting in which applications may be developed, run, managed, and hosted.

Although each modeled VM in the preceding virtual environment has its own unique context, they must nonetheless share hardware components like CPUs, memory, and the network.

As a result, the host's overall processing power (CPU), memory, and system bandwidth determine how much hardware may be allocated to each VM. You get to choose whatever virtual machine to use, which means you may tailor its hardware specifications to meet the needs of a certain program.

CloudSim allows for two tiers of virtual machine provisioning:At the host level, you may control how much of each core's total processing power is allocated to each virtual machine. Virtual Machine Allocation Policy

Virtual machines (VMs) provide a certain portion of its execution engine's available processing power to the many application services (task units) running on it. Scheduling Virtual Machines. It is important to remember that CloudSim uses time-shared and space-shared provisioning strategies on all levels.

In this research, we present a VM load balancing technique (VM Scheduling-time shared) in which the available processing power of VMs is distributed unevenly across the various application services. This is because, in the real world, the processing capability of a DataCenter's virtual machines (VMs) may vary throughout its many computing nodes.

The most powerful VM is given the tasks/requests (application services), followed by the next most powerful VM, and so on down the line. They are assigned the necessary levels of importance. Therefore, measures of performance like response and processing times are fine-tuned.

3.1 Simulation

By recording and replaying real observations from a production system, or by mathematically modeling the interaction between the many components of the system (CPU, network, etc.), simulation mimics its behavior. The current crop of Cloud Computing simulation software includes simjava, gridsim, and CloudSim.

In the Cloud Simulation (3.2) [1][3][12]

The University of Melbourne's GRIDS lab has created a framework called CloudSim that facilitates the modeling, simulation, and experimentation necessary for the design of Cloud computing infrastructures. CloudSim is an independent environment for simulating Cloud infrastructure at scale, including data centers, hosts, service brokers, and scheduling and allocation mechanisms. The GRIDS lab also created the foundational framework GridSim upon which CloudSim was constructed. In order to conduct tests in a controlled setting, the researcher has utilized CloudSim to create models of datacenters, hosts, and virtual machines.

This article presents a novel virtual machine (VM) load balancing technique called the "Weighted Active Monitoring Load Balancing Algorithm" to manage the influx of service requests from the user base. [7]. Section 4.0 presents modern VM load balancers, Section 5.0 details an optimized VM load balancing technique for faster cloudlet responses and data processing, and Section 6.0 summarizes the study's methodology and findings.

4 MODERN VM LOAD BALANCERS

By executing an OS and its applications in a virtual machine, it is possible to isolate them from the underlying hardware. The Cloudsim simulator uses a Datacenter component to represent the internal hardware infrastructure services connected to the Clouds and to process service requests. Requests of this kind come from the application components housed in isolated virtual machines (VMs) and need resources from the Datacenter's host components. DataCenter object handles the routing of user requests and other data center administration tasks including the creation and removal of virtual machines.

access the VMs via the Internet from User Bases. A VmLoadBalancer is used by the Data Center Controller [7] to decide which VM will be given the next processing request. Modern Vmloadbalancer use load balancing methods including Round Robin, throttling, and active monitoring.

This is where the RRLB comes in.

The requests are distributed across a pool of virtual machines (VMs) at a data center in a random fashion. After the first request is sent to a virtual machine at random, the DataCenter controller allocates following requests in a recursive fashion. After a virtual machine (VM) receives a request, it is pushed to the end of the queue. Weighted Round Robin Allocation is a superior allocation idea in this RRLB. It allows you to give each virtual machine a relative importance, giving the more powerful server a weight of 2 if it can do twice as much work as the others. When this occurs, the DataCenter Controller will provide the more powerful VM two requests for every request sent to the less powerful VM.

The main problem with this allocation is that it ignores important factors like request processing times while attempting to balance loads.[1]

Load-sharing device that may be throttled

The TLB tracks the availability and use of all virtual machines. When a request is received for a virtual machine's allocation, the TLB communicates the optimal virtual machine's ID to the data center controller.

Balancer (AMLB) with Active Monitoring

THE AMLB keeps track of which virtual machines are being used for which requests. In response to a request for a new virtual machine, the system will look for the least busy one available. If more than one is found, the one that is found first will be used. The Data Center Controller receives the VM id from ActiveVmLoadBalancer. The Data Center Manager then forwards the request to the specified Virtual Machine. The cloudlet is then forwarded to the ActiveVmLoadBalancer once the DataCenterController has informed it of the new allocation.

ALGORITHM FOR WEIGHTED ACTIVE MONITORING AND LOAD BALANCE

In order to improve response time and processing time, the 'Weighted Active Monitoring Load Algorithm' is put into place, which involves changing the Active Monitoring Load Balancer by giving a weight to each VM, similar to the Weighted Round Robin Algorithm of cloud computing.

Using the idea of weights in active monitoring, the suggested Load balancing algorithm distributes the server's or host's processing power across the VMs in a way that best suits each service's needs. The most powerful virtual machine (VM) is given the most important duties or requests (application services), followed by the next most powerful VM, and so on. Therefore, maximizing the potential of the existing specs.

WEIGHTED ACTIVE MONITORING LOAD BALANCER

(Algorithm)

In the first step, VMs from various Datacenters are created based on the host/physical server's processing capability (as measured by its core processor, processing speed, memory, storage, etc.).

The second step is to assign a weighted count based on the processing capacity of the Datacenter's virtual machines. When comparing two virtual machines (VMs), the one that can handle twice as much work is given a weight of 2, the one that can handle four times as much work is given a weight of 4, and so on.

Case in point:

• The weighted count for a host server with a single core CPU, one gigabyte of memory, one terabyte of storage capacity, and one million megabits of bandwidth is one.

• A host server with a 2-core CPU, 4-GB of RAM, 2-TB of storage, and 1-Gbps of bandwidth will have a weighted count of 2.

A weighted count of 4 indicates that a host server with a quad core CPU, 8GB of memory, 4TB of storage capacity, and 1,000,000 Mbps of bandwidth is optimal. to continue...

STEP 3: WeightedActiveVmLoadBalancer keeps track of the amount of requests assigned to each virtual machine as well as the weighted count associated with that virtual machine. All virtual machines start with a quota of zero.

Fourth, when a new VM allocation request comes in from the DataCenterController, the table is analyzed to determine which VM is the least busy.

Fifth, after it has determined which virtual machines (VMs) across data centers are the least busy, it will distribute requests to the most powerful VM based on the provided weight. If more than one is found, the one that is found first will be used.

Sixth, the DataCenterController receives the VM id from the WeightedActiveVmLoadBalancer.

In the seventh step, the DataCenterController forwards the request to the virtual machine with the specified id.

The eighth step is for the DataCenterController to inform the WeightedActiveVmLoadBalancer of the change.

The allocation table is then updated by WeightedActiveVmLoadBalancer, which adds further allocations for that VM.

In STEP 10, when the VM has completed its work and the DataCenterController has received the cloudlet with the answer, the DataCenterController will inform the WeightedActiveVmLoadBalancer that the VM should be released.

In STEP 11, the allocation table is modified by the WeightedActiveVmLoadBalancer by reducing the VM's allocation count by one.

Step 12: Proceed with step 4 from above.

The goal of the method is to calculate how long it should take for each Virtual Machine to respond. Response times may be calculated with the assistance of the following formulae, despite the fact that the processing power of individual virtual machines varies widely.

An appropriate response time is defined as: The transmission delay may be calculated using the following formulae, where Arr and Fin are the times at which the user's request arrived and completed, respectively.

T + T(2) = TDelaylatencytransfer If the delay in transmission is TDelay, then The time required to send a single request's worth of data (D) from its origin to its destination is denoted by Ttransfer, while network delay is denoted by Tlatency.

The formula for Ttransfer is: (3) D / Bwperuser (4) Bwperuser = Bwtotal / Nr Where Bwtotal represents the total bandwidth and Nr represents the number of requests being sent by users at the moment. For the value of Nr, Internet Characteristics additionally records the total number of user requests that have been sent between two areas.

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

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 research, we present a novel Virtual Machine (VM) Load Balancing Algorithm and show how it may be implemented in a Cloud Computing setting by means of the CloudSim toolkit written in Java. Using this method, the VM allots a distinct fraction of the total CPU time to each service in the application. Tasks and requests (application services) are distributed across these VMs in descending order of processing capability, starting with the most powerful. As a result, we have developed a powerful VM Load Balancing technique called the "Weighted Active Load Balancing Algorithm" for use in Cloud Computing by optimizing the specified performance criteria including response time and data processing time.

References /Bibliography

[1] Cloud computing: state-of-the-art and research problems; Published online: 20th April 2010, Copyright: The Brazillian Computer Society 2010.
[2] Above the Clouds: A Berkeley Perspective on Cloud Computing [2] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D.Joseph, Randy Katz; The Regents of the University of California, 2009.
[3] [3] Business and Information System Engineering, Volume 5, Issue 3, Pages 391–399; Christ oph Weinhardt, Benjamin Blau, and Jochen Stober; "Cloud Computing: A Classification, Business Models, and Research Directions."
[4]
[5] Grid Computing and Distributed Systems (GRIDS) Laboratory, Department of Computer Science and Software Engineering, The University of Melbourne, Australia; Pontificial Catholic University of Rio Grande do Sul Porto Alegre, Brazil; 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; [4]. @csse.unimelb.edu.au: "Rodrigo, rranjan, raj," @pucrs.br: "cesar.derose@pucrs.br"
[6] [5] Software: Practice and Experience (SPE), Volume 41, Number 1, Pages: 23-50, ISSN: 0038-0644, Wiley Press, New York, USA, January, 2011 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
[7] Modeling and Simulation of Scalable Cloud Computing Environments with the CloudSim Toolkit: Challenges and Opportunities[6] by Rajkumar Buyya, Rajiv Ranjan, and Rodrigo N. Calheiros. High Performance Computing and Simulation 2009: Proceedings, IEEE Press, New York, USA, 978-1-4244-4907-1, June 21-24, 2009, Leipzig, Germany.
[8] [7] The authors Bhathiya Wickremasinghe, Rodrigo N. Calheiros, and Rajkumar Buyya titled their paper "CloudAnalyst: A CloudSim-based Visual Modeller for Analysing Cloud Computing Environments and Applications" and published it in the proceedings of the 2010 conference held in Amsterdam, Netherlands, from April 20-23.
[9] IBM Academy of Technology Thought Leadership White Paper, October 2010. Cloud computing ideas from 110 implementation projects.
[10] Reference: [9] "A Study of the Parameters Concerning Load Balancing Algorithms," Ioannis Psoroulas, Ioannis Anagnostopoulos, Vassili Loumos, and Eleftherios Kayafas, IJCSNS International Journal of Computer Science and Network Security, Volume 7, Number 4, 2007, Pages 202-214.
[11] For example, see [10] "Performance Analysis of Load Balancing Algorithms" by Sandeep Sharma, Sarabjit Singh, and Meenakshi Sharma in World Academy of Science, Engineering, and Technology vol. 38, pages 269-272 (2008).
[12] [11] The Cloud Computing and Distributed Systems (CLOUDS) Laboratory's CloudSim 2.1.1 Application Programming Interface (API).