Reverse Host Allocation Approach for Virtual Machine Cloud Computing Environment

Divya Chaudhary
M.Tech Scholar
Department of Computer Science and Applications,
M.D. University, Rohtak, Haryana, India

Rajender Singh Chhillar
Professor and Head
Department of Computer Science and Applications,
M.D. University, Rohtak, Haryana, India

ABSTRACT

Cloud Computing is one the latest and evolving technology in the coming times. It has evolved as a balloon filled up with immense potential of making the world bloom with its applications. Cloud provides us with the higher degree of capability by using its resources and fulfilling the needs of the future generations. It provides greater resource utilization with the help of load balancing principles for the completion of the tasks. This paper discusses the concept of load balancing in the virtual machine cloud computing environment highlighting the procedures of the resource i.e., host and virtual machine allocation and de-allocation. This paper exhibits a new load balancing approach in cloud environment and its comparison with the existing approaches.

Keywords: Load Balancing, Cloud Computing, Reverse Host, Reverse Filling, Round Robin Scheduling, CloudSim.

1. INTRODUCTION

Cloud Computing is one of the latest and forthcoming paradigm in the world of computing. It has evolved greatly from cluster and utility computing. It provides an extensive potential for computational purposes and generates efficient results in the system. In future it has the capacity of making "computing as a utility" a success. Cloud Computing and Grid Computing has evolved gradually from the distributed computing. As we know that, in distributed systems resources can be accessed from far away and distant locations. It targets the capability of provoking greater usage and development of applications for the betterment of the mankind. It is used to describe both a platform and a type of application. It supports distributed system design and application to support user oriented server applications. In the IT industry it would be highly beneficial for the designing of the numerous application systems [1]. Cloud computing presents a distributed system in which the data is transferred from a personal computer to servers or computers located at remote areas with the help of high speed networks.

Cloud Computing consists of two words i.e., Cloud and Computing. A Cloud is defined as expandable or scalable distributed system where the resources, systems and storage spaces are distributed diversely through the network. Secondly, Computing as defined by ACM Computing Curricula 2005 states as a goal oriented activity requiring, benefitting from, or creating computers. Thus, Computing includes designing and building hardware and software systems for a wide range of purposes: processing, structuring and managing various kinds of information, doing scientific studies using computers, making computer systems behave intelligently, creating and using communications and entertainment media, finding and gathering information relevant to any particular purpose, and so on. It is synonymous to counting and calculating. Cloud Computing offers dynamic stability of the system. It provides the services of accessing a huge amount of computer resources and storage spaces providing subscription based services supporting the concept of on demand computing. A Cloud represents virtual data centres or hubs containing bulk of huge information warehouses. The data or information is mined from the warehouse using the large number of tools and applications designed [2]. The computing strategy in cloud offers scalable and timely allocation of the resources on the basis of the amount of usage in the system. It has the potential warehouse huge data and resources. NIST states the cloud computing definition as:-

"Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, services) that can be rapidly provisioned and released with minimal management effort or service provider interaction."

The main idea driving the cloud computing is the managing and scheduling uniformly computing resources connected by a network and supporting user resources and services in correspondence to the needs. It provides the ability to use the resources according to the requirements of the users and payment according to the usage. It helps in transmission of data and resources across remote locations and fulfilment of the tasks. Cloud Computing offers the services at various levels namely: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). The SaaS using standardised interfaces offers the services as application to the users. The PaaS provides the concept of operation and development of platforms to the users. Lastly, IaaS provides the capability of low service models, supplying raw data storage, network capacity and processing power [3]. These offer a huge amount of deployment possibilities for the users and developing a naïve framework. The rest of this paper is organized as follows. The load balancing concepts are discussed in Section II. The existing load balancing algorithmic approach is presented in Section III. The proposed work is specified in Section IV. Section V describes the experimental setup for implementation of the proposed algorithm in simulation environment, CloudSim. Finally, the conclusions and the future works are discussed in Section VI.

2. LOAD BALANCING

The Cloud Computing is a scalable and distributed system requiring the allocation of the resources to particular client. This could lead to congestion or imbalanced allocation of the system [3]. So, to deal with the imbalance in the network we

need a load balancing strategy. Load Balancing is the mechanism that is used for distributing the load on the system processes or virtual machines for the optimal resource utilization. Load refers to the work needed to be done on the system. It can also be said as the handling or the management of the load. It is applied on the resources in the system which can be disks, drivers, memory buffers, processors, network simulators, etc. [4]. Load balancing is done to achieve maximum response time, maximum throughput and the reduction of the overheads produced by the system. The main aim is the processing of the user's request by the efficient and effective utilization of the resources present in the system.

Load Balancing is the task of effectively distributing the load on the different web servers. It is done to increase the performance of the particular system. Providing a way to the much anticipated Green computing having power efficient systems. It also provides cost effectiveness, efficient load distribution, system stability and many more [5]. A large number of algorithms are used for scheduling of load in the system. These algorithms on the basis of position of the noses or system topology are divided into two categories namely static and dynamic algorithms.

a) Static Approach

It is defined in the design and the implementation of the system. In this system the nodes are allocated in advance or predefined in the network. They maintain a state table which is provided to every node indicating the process of balancing of the load in the system. This table is provided in advance to every node of the system. Round Robin Algorithm (RR) and Weighted Round Robin Algorithm (WRR) are simplest of the static algorithms.

b) Dynamic Approach

This approach considers the current position or state of the system for the balancing purposes. All the nodes present in a particular system maintain the state table indicating the state of the particular nodes present. The table contains the path followed to reach a particular point and the future path to be followed for the efficient allocation of tasks to the nodes. This is suitably used in large distributed systems.

For the fulfilment of the goals and producing the optimized results and balancing the load of the system many load balancing algorithms have been designed. The main points to ponder [6] during the process of development of these algorithms are:

a) Estimation and comparison of load.
b) Performance of the system
c) Interaction and selection of nodes
d) Stability of the systems
e) Manner of the load or work transfer.

Load Balancing systems have found greater usage in a number of our day to day things [7]. Some of the few examples are: DNS (Domain Name Server), ZXTM LB (Zeus Extensible Traffic Manager Load Balancer) and AMAZON Load Balancing. These provide effective load balancing strategies which are widely seen [8]. The basic tasks needed to be fulfilled by the Load Balancing Algorithms are:

a) Distribution of data through virtual machine migration.
b) Distribution and management of the stored data.
c) Creation and development of small data centres.
d) Designing of power efficient systems through energy management.
e) Provisioning of resources and services automatically.

3. EXISTING APPROACH

Round Robin Algorithm follows the static approach in nature. It follows the principle of Decentralization. Here, the connections are established between a large number of systems namely clients and the servers. It follows a round robin mode of balancing for the load present in the system [9]. In Operating Systems, the round-robin (RR) scheduling algorithm is designed especially for timesharing distributed systems. It is similar to FCFS (First Come First Serve) scheduling, but pre-emption is added to enable the system to switch between processes. A small unit of time, called a time quantum or time slice, is defined. A time quantum generally starts from 10 milliseconds and goes to 100 milliseconds in length. The ready queue is treated as a circular queue. It is used for storing the processes and forwarding them for execution to the scheduler. The CPU scheduler goes around the ready queue, allocating the CPU to each process for a time interval of up to 1 time quantum. We keep the FIFO (First in First Out) queue of processes. New processes are added to the tail of the queue. The CPU scheduler picks the first process from the queue, sets a timer to interrupt after 1 time quantum, and dispatches the process. It is also known as processor sharing approach and helps in achieving higher response time.

The allocation order of the processes is maintained locally which are independent of the allocation order of the remote processors. The load that is distributed between the various processors is equal which means that load distribution strategy divides the processes equally for execution whereas it does not provide equal job processing time to the systems or processes. There are times when during processing some systems are heavily loaded and others are scantly loaded or remain idle by the scheduler. It is mostly seen in the case of the web servers. Another algorithm that is used is Weighted Round Robin Algorithm approach. In this algorithm the weights are assigned to the system for the efficient resource utilization. It solves the problem of inconsistent server's performance by adding weight. The weights are assigned and the tasks are completed in a proper prioritized manner as the loads are first arranged in an order and then the weights categorization is dine [3]. These two algorithms follow the strategy of having the path stored on the nodes in the beforehand the process of balancing of the nodes.

In the existing Round Robin Load Balancing Algorithm, the data centre controller uses a load balancer named round robin load balancer for the allocation purposes. It is done to allocate the virtual machines to the clouds in the system. The load balancer maintains an index table of the virtual machines in the system. The data centre receives queries and processes the requests in the system for the allocation purposes. It then allocates the virtual machines to the cloudlets in the system in a round robin fashion. [12] The ids (identifiers) of the virtual machines are stored in the index table. The load balancer assigns the virtual machines and cloudlets along with updating of the allocation table. After the completion of the task the de-allocation process occurs in the similar manner as allocation. After studying the algorithm, it is observed that this needs improvement. Hence a new approach is designed in cloud computing environment for higher and better response time, performance and cost/ debt.

4. PROPOSED REVERSE HOST APPROACH

The proposed naïve approach follows the principle of Reverse Host Allocation approach for the task of load balancing. This is the advancement in the reverse filling approach [12] as described. As in the reverse filling approach the data centres manage the virtual machines that are created. The virtual machines and the ids are stored in the index table of the systems. The hosts and the virtual machines are then allocated in the reverse order to the cloudlets or the processes. Thus, in particular they follow a strategy of Last in First out Order (LIFO) and giving lesser debt, cost, waiting time and greater response time. Now, in this new and latest approach we take into consideration the reverse host approach of the system. The system shuffles the allocation strategy of the hosts in the distributed environment. [12] Here, in this approach we generally have a Load Balancer whose main task is to deal with the virtual machines and the cloudlets. The virtual machines take into reverse host allocation approach of the system as a whole. The algorithm is as follows:

1. NewVmLoadBalancer makes an index table of Virtual Machines and the state of the VM (BUSY/AVAILABLE). At the start all VM's are available.
2. DataCenterController receives a new request.
3. DataCenterController queries the NewVmLoadBalancer for the next allocation.
4. The NewVmLoadBalancer parses the table.
5. The NewVmLoadBalancer shuffles the table storing the virtual machine id's and list.
6. The NewVmLoadBalancer then creates a new allocation table containing the index table of the virtual machines.
7. The NewVmLoadBalancer now allocates the hosts in a reverse order in the Host List of the system.
8. The NewVmLoadBalancer allocates VMs to the resources.VM is found.
   If found:
   a. The NewVmLoadBalancer returns the VM id to the DataCenterController
   b. The DataCenterController sends the request to the VM identified by that id.
   c. DataCenterController notifies the NewVmLoadBalancer of the new allocation
   d. NewVmLoadBalancer updates the allocation table accordingly in the similar manner.
   If not found:
   e. The NewVmLoadBalancer returns -1.
   f. The DataCenterController queues the request
9. When the VM finishes processing the request, and the DataCenterController receives the response cloudlet, it notifies the NewVmLoadBalancer of the VM de-allocation.
10. The DataCenterController checks if there are any waiting requests in the queue. If there are, it continues from step 3.
11. Continue from step 2.

5. EXPERIMENTAL SETUP

For the execution of this algorithm a large number of the simulation packages are available namely: CloudSim, SimCloud, NS2 (Network Simulator), MATLAB and many others. CloudSim (Cloud Simulator) is an environment of the easier creation of the cloud networks. CloudSim have many versions for implementation ranging from CloudSim 1.0 to the latest CloudSim 3.0.3. The proposed naïve approach is executed and implemented through simulation package: CloudSim 3.0. [11]. CloudSim 3.0 is run on the Java Net Beans interface. The CloudSim consists of a large number of classes like Cloudlet, Scheduler, Data Center, Hosts, etc which are extended and inherited to implement a particular algorithm. It provides a simulated environment for performing research on virtual machines by the allocation and the broker policies. Java language is used for implementing VM load balancing algorithm. [13]Assuming the application is deployed in two data center's having 20 virtual machines (with 1024MB of memory in each VM running on physical processors capable of speeds of 250 MIPS) and Parameter Values are as given in Table 1 [12]:

Table 1: Parameter and Values

Parameter	**Values**
Data Center OS	Window 7
VM Memory	10000 MB
Data Center Architecture	X86
Service Broker Policy	Optimize Cost & Time
VM Bandwidth	1000
Users	2

Table 2 shows the experimental results based on Reverse Host Approach VM Load Balancing Algorithm and its comparison to the other two algorithms [12]:

Table 2: Results of Round Robin, Reverse Filling [12] and Reverse Host Approach VM Load Balancing Algorithm

Values	**Round Robin**	**Reverse Filing Algorithm**	**Reverse Host Approach**
Arrival Time	0.3 ms	0.3 ms	0.3 ms
Finish Time	480.3 ms	800.3 ms	1600.3 ms
Cloudlets	40	40	40
Virtual Machines	20	20	20
Response Time	160.5 ms	160.5 ms	160.5 ms
Minimum Response Time	12 ms	20 ms	40 ms
Processing Elements	1	1	1
DataCenters	2	2	2
File / Output Size	300	300	300
Cost Of Processing	3.0	3.0	3.0
Cost Per Memory	0.05	0.05	0.05
Cost Per Bandwidth	0.1	0.1	0.1
Cost Per Storage	0.1	0.1	0.1
Debt/ Total Cost	10256	4102.4	2051.2

Figure 1 and Figure 2 below highlights the graphical comparison between the existing Round Robin load balancing algorithm and Reverse Filling load balancing algorithm [12] on the time taken for the task to complete and the cost incurred in its completion. And Secondly, Figure 2 [12] below highlights

their graphical comparison with the proposed Reverse Host Approach.

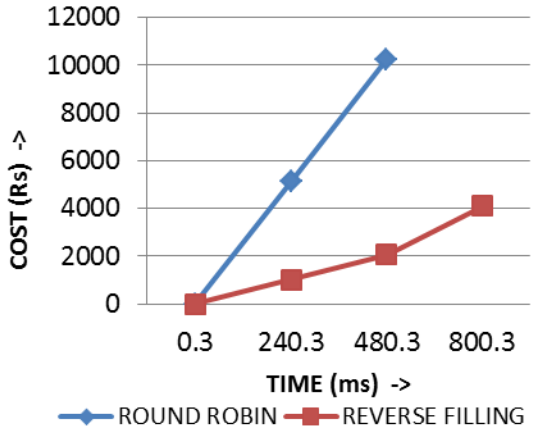

Figure 1: Graphical comparison of the existing Round Robin and Reverse Filling algorithm [12].

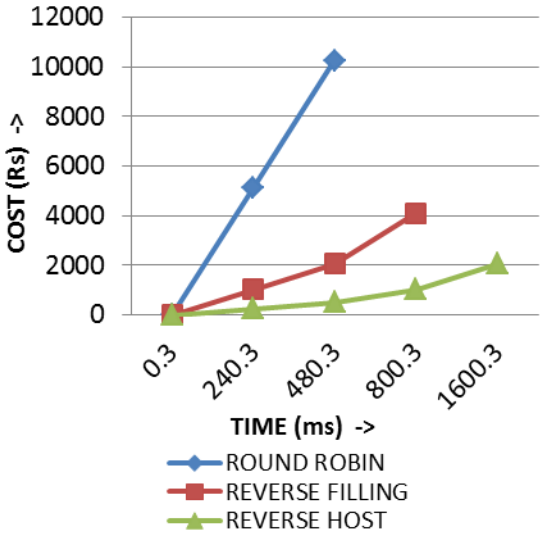

Figure 2: Graphical comparison of the existing Round Robin, Reverse Filling algorithm and proposed Reverse Host Approach

Thus, it is observed that the implementation of the proposed reverse host algorithm yields much lesser cost as compared to the existing two algorithms i.e., Round Robin and Reverse Filling algorithms. It also reduces the amount of the overheads and other additional costs incurred in the system. Thereby reducing the waiting time in the system and increasing the efficiency of the system providing more efficient services to users or clients. It also generates greater resource utilization in the system with less debt or cost.

6. CONCLUSION AND FUTURE SCOPE

Load Balancing is one of the prima facie of the cloud computing technology. It is quite helpful in reducing the workload of the system nodes for greater response time and reducing the cost thereby increasing performance. It helps in reducing the overheads i.e., costs incurred and increasing scalability of the cloud. This paper highlights the cloud computing fundamentals in brief. Its basic characteristics like on-demand access, resource provisioning, etc. are supplied. Secondly, it depicts the basic concept of load balancing defining its major tasks and algorithmic goals. Thirdly, it presents the review of the most prominent and static load balancing algorithms in the system. There features as well as the nature and way of working depicting stability. Then, it provides the proposed work on the simulated environment. Lastly the experimental setup used for the implementation of the virtual system and the resources used are specified. They depict that the system has greater resource utilization providing lesser overheads and waiting time. It showcases that the proposed approach is far better than the existing two approaches used as it provides lesser cost and greater response time. The future work includes modifications in the system leading to greater resource utilization and performance as well further reduction in the costs incurred.

7. REFERENCES

[1] Bhathiya Wickremasinghe, Rodrigo N. Calheiros, and Raj Kumar 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 (2010)

[2] Martin Randles, David Lamb, A. Taleb-Bendiab, “A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing”, 2010 IEEE 24th International Conference on Advanced Information Networking and Applications Workshops, pp. 551-556.

[3] Divya Chaudhary, Prof. Rajender Singh Chhillar, “Strategic Evaluation of Load Scheduling Techniques in Cloud Computing: A Review”, In the Proceedings of

[4] Volume-II, NCACT-2013, DCSA, M.D University, Rohtak, 548-552.

[5] F el. M a, Feng L1' U and Zhen L1' U, “Distributed Load Balancing Allocation of Virtual Machine in Cloud Data Centre”, 978-1-4673-2008-5/12/$31.00 ©2012 IEEE

[6] Shu-Ching Wang, Kuo-Qin Yan, Wen-Pin Liao, Shun-Sheng Wang, “Towards a Load Balancing in a Three-level Cloud Computing Network”, 2010 IEEE, pp. 108-113.

[7] Jinhua Hu, Jianhua Gu, Guofei Sun, Tianhai Zhao, “A Scheduling Strategy on Load Balancing of Virtual Machine Resources in Cloud Computing Environment”, 3rd International Symposium on Parallel Architectures, Algorithms and Programming, 978-0-7695-4312-3/10 $26.00 © 2010 IEEE DOI 10.1109/PAAP.2010.65

[8] A.KHIYAITA, EL BAKKALI, M.ZBAKH, Dafir EL KETTANI, “Load Balancing Cloud Computing : State of Art”, IEEE Transactions on Software Engineering, 978-1-4673-1053-6/12/$31.00 ©2012 IEEE

[9] Xiaona Ren, Rongheng Lin,Hua Zou, “A DYNAMIC LOAD BALANCING STRATEGY FOR CLOUD COMPUTING PLATFORM BASED ON EXPONENTIAL SMOOTHING FORECAST”, In the Proceedings of IEEE CCIS2011, 978-1-61284-204-2/11/$31.00 ©2011 IEEE

[10] Rashmi. K. S, Suma. V, Vaidehi. M, “Enhanced Load Balancing Approach to Avoid Deadlocks in Cloud”, Special Issue of International Journal of Computer Applications (0975 – 8887) on Advanced Computing and

Communication Technologies for HPC Applications - ACCTHPCA, June 2012

[11] Bhaskar. R, Deepu. S.R and Dr. B.S. Shylaja, "DYNAMIC ALLOCATION METHOD FOR EFFICIENT LOAD BALANCING IN VIRTUAL MACHINES FOR CLOUD COMPUTING ENVIRONMENT", Advanced Computing: An International Journal (ACIJ), Vol.3, No.5, September 2012 DOI : 10.5121/acij.2012.3506 53

[12] Calheiros Rodrigo N., Rajiv Ranjan, César A. F. De Rose, Raj Kumar Buyya (2009): CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services CoRR abs/0903.2525: (2009)

[13] Divya Chaudhary, Prof. Rajender Singh Chhillar, "A New Load Balancing Technique for Virtual Machine Cloud Computing Environment", International Journal of Computer Applications (0975 – 8887) Volume 69– No. 23, May 2013.

[14] CloudSim: A Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services, The Cloud Computing and Distributed Systems (CLOUDS) Laboratory, University of Melbourne, (2011) available from: http://www.cloudbus.org/cloudsim/.