

Roulette Wheel Selection Model based on Virtual Machine Weight for Load Balancing in Cloud Computing

Rashed Al-Marhabi, Mohamed Haggag, Amal Elsayed Aboutabl

Computer Science Department, Faculty of Computers and Information, Helwan University, Egypt

Abstract: The increasing use of cloud computing resources have placed a large workload on the data centers. Solving the load balancing problem among all virtual machines (VMs) increases performance, throughput and decreases the request delay. In this paper, Roulette Wheel Selection algorithm is applied on all VMs for balancing the load in the cloud computing environment based on virtual machine weight. In our experiments, measures such as response time, data center request servicing time and processing cost are used. Simulation results show that Roulette Wheel Selection algorithm reduces data transfer cost when compared to three other load balancing algorithms.

Keywords: Cloud computing, Roulette Wheel Selection, VM load balancer.

I. Introduction

Cloud computing is a new computing model that changed our perspective to the technology and moved our information and applications away from personal computers to a “cloud” of computers. Today’s emerging cloud computing, users only need to be concerned with the computing services being requested and all services will be performed by cloud computing provider[1][12][22].

1.1. Characteristics of Cloud Computing

There are four basic characteristics of cloud computing [1] [3]:

- a. Elasticity and scalability.
- b. Self-service provisioning and automatic de-provisioning.
- c. Standardized interfaces.
- d. Billing self-service based usage model.

1.2. Types of Deployment Models of Cloud Computing

Cloud services can be deployed in different ways depending on the organizational structure and the provisioning location [1] [13]. Four deployment models are usually distinguished, namely as follows.

1. **Public cloud:** it is freely available for access based on the standard cloud computing. The service provider makes resources, such as applications and storage, available to the general public over the Internet.
2. **Private cloud:** it allows the usage of services by a single client on a private network. The importance benefits of this model are data security, corporate governance, and reliability concerns. The private cloud is used by the organization when it has a huge, well-run data center having a lot of spare capacity. It is also used when an organization is providing IT services to its clients and the data of organization is highly important. It is best suited when the requirements are critical.
3. **Community cloud:** It provides a number of benefits, such as privacy and security. This model, which is quite expensive, is used when the organizations having common goals and requirements are ready to share the benefits of the cloud service.
4. **Hybrid cloud:** It consists of multiple service providers. This model integrates various cloud services for Hybrid Web hosting. It is basically a combination of private and public cloud features. It is used by the company when a company has requirements for both the private and public clouds.

The rest of this paper is organized as follows. The literature review is discussed in section 2. Section 3 introduces the Roulette Wheel Selection Model, section 4 tells the Virtual Machine Weight definition and section 5 present the design model of the proposed algorithm. Section 6 discusses the simulation setup and results. Finally, section 7 concludes this paper.

II. Background

The cloud computing environment has a large number of resources and users which requires managing those resources using an efficient algorithm such as load balancing algorithms. Sometimes, there is an extremely large load over some VMs while others are idle. An idle VM is assigned load from a busy VM. In this work, our proposed model is compared with three other VM Load Balancing algorithms [2][18][20]:

1. Round Robin Algorithm: is a random selection algorithm, it used the concept of the time slot, when the time is divided into multiple slots and each VM is given a particular time slot and in this slot the VM will perform its task. The resources of the cloud computing provider are provided to the users on the basis of this time slot. In Round Robin load balancing, the time slot plays a very important role because if the time slot is very large then Round Robin balancing Algorithm is the same as the FCFS balancing. It selects the load on random basis and leads to the situation where some VMs are heavily loaded and some are lightly loaded. Though the algorithm is very simple, there is an additional load on the balancer to decide the size of the slot and it has longer average waiting time, higher context switches, higher turnaround time and low throughput [20][21].

2. Equally Spread Current Execution Algorithm (ESCE): In this algorithm, the load balancer makes effort to preserve equal load on all the virtual machines connected with the data center. ESCE Load balancing maintains an index table of the VM as well as number of requests currently assigned to the VM. If the request comes from the data center to allocate the new VM, it scans the index table for the least loaded VM. In case more than one VM is found, the VM identified first is selected for handling the request of the user. The load balancer also returns the VM id to the data center controller. The data center communicates the request to the VM identified by that id. The data center revises the index table by increasing the allocation by a count of identified VM. When VM completes the assigned task, a request is communicated to the data center which is further notified by the load balancer. The load balancer again revises the index table by decreasing the allocation count for the identified VM by one. There is an additional computation overhead to scan the queue again and again [18][20].

3. Throttled Load Balancing Algorithm (TLB): The user first requests the load balancer to find a suitable VM to perform the required task. TLB algorithm maintains an index table of virtual machines as well as their states (Available or Busy). The data center queries the load balancer for allocation of the VM. The TLB algorithm scans the index table from top until the first available VM is found or the index table is scanned fully. The data center communicates the request to the VM identified by the id. Further, the data center acknowledges the load balancer of the new allocation and the data center changes the index table accordingly. While processing the request of a client, if appropriate VM is not found, the load balancer returns -1 to the data center. The data center queues the request with it. When the VM completes the allocated task, a request is acknowledged to the data center, which is further informed to load balancer to deallocate the same VM whose id is already communicated [12][18].

The problem of the above load balancing algorithms is that they do not check the current state (available resources) of the VMs in the data center so as to assign a request from a Userbase to a suitable VM in the data center. As such, every time a request is received from the Userbase, the request is queued until the VM state indicates that it is available. This increases the total response time of the requests. In this research, the Roulette Wheel Selection Algorithm is used in the Virtual Machine to address this problem.

III. Roulette Wheel Selection Model

In this algorithm, all the VM weights in all the data centers are placed on the virtual roulette wheel according to their specification (weight value). Each individual VM is assigned a segment of roulette wheel. The size of each segment in the roulette wheel is equivalent to the VM available specification (a higher specification value means better weight) [16].

Then, the virtual roulette wheel is pinned. The individual corresponding to the segment on which roulette wheel stops is then selected. Individuals with higher weight have more probability of selection. The average VMs weight of the population for the i^{th} VM in roulette wheel selection is calculated as follows [14]:

Suppose there are P individual VMs with weight w_1, w_2, \dots, w_P ;

$$\text{Let } S = \sum_{j=1}^n P_j \tag{1}$$

The Average weight of the population in i^{th} is

$$P_i = \frac{w_i}{\sum_{k=1}^P w_k} \tag{2}$$

Where:

P is the total probability of VM

N is the number of individuals VM

w is the VM weight

S is the sum of the VMs probability values (=1)

Eq. 1 presents the sum of the VMs weight values over all the individuals VM in the data center. The circumference of the roulette wheel is the sum of all weight values of all VM in the data centers. The suitable VM occupies the largest interval, whereas the least weight has correspondingly smaller intervals within the roulette wheel. To select an individual, a random number is generated in the interval [0, S] and the individual whose segment spans the random number is selected.

IV. Virtual Machine weight

The Virtual Machine Weight is defined according to the VMs' specifications, particularly available resources at the current time. When the user request arrives to the load balancer, the algorithm updated the information about all VMs in the system according to available resources so as to check the new states of the system. Using virtual machine weight is an explorative technique of selection. Virtual machine weight prevents too quick convergence and differs from roulette wheel selection in terms of selection pressure. Virtual Machine weight overcomes scaling problems such as stagnation or premature convergence. Ranking controls selective pressure by uniform method of scaling across the VMs. Mentoring the Virtual Machine weight makes the system more robust and increases the system performance. Virtual Machine Weight is computed by giving weight for all VMs in the system depending on the specification of the VM in the data center and bandwidth available between the data center and user base.

V. Proposed Algorithm: Roulette Wheel Selection Algorithm with Virtual Machine weight

In this paper, we apply and experiment the Roulette Wheel selection algorithm for load balancing in cloud computing according to the available resources of virtual machines to balance the work load over all virtual machines in the pool of our cloud computing environment. The proposed load balancing algorithm is to move the VM selection criteria from exploration to exploitation so as to obtain the perfect mix of the two techniques. In this algorithm, weight value of each VM is computed. Depending upon the current weight of VM, selection pressure is changed, new weight contribution of each VM is computed and selection probability of each VM is also computed. As the weight of the VM changes, the probability of each individual VM also changes. In this algorithm there are two basic activities:

1. Information collection: This activity deals with collecting information about all VMs in the data centers and assigning weight to each VM depending on the specification of the VM and available bandwidth between the Userbase to the VM. The weight value of the VM guiding the decision on assigning new requests to appropriate VMs depends on the current state of the VM in the system.

2. VM assignment. It deals with efficient ways to choose and assign the Userbase requests to the highest weight VM in the system.

The roulette wheel selection algorithm with Virtual Machine Weight is selected by sorting the VMs first according to their weights at the current time. Then, every VM is allocated a selection probability value with respect to its weight. The VMs are selected as per their selection probability. The steps of the roulette wheel selection algorithm with Virtual Machine Weight are as follows:

1. Efficiently compare the information gathered by the algorithm about each VM in the system and weigh each VM in all DC depends on the available resources.
2. Put the information about all VMs in the virtual sorted roulette wheel
3. The user request consults the load balancer to know which VM will perform the requests ,
4. Information about all VMs in the virtual Sorted Roulette wheel is updated.
5. According to the VM status on the system and the roulette wheel stops , probability are selected.
6. Assign the selected VM an ID within its data center and decrease the weight of the selected VM in the virtual roulette wheel.
7. The VM processes the user requests then sends a message to the virtual roulette wheel algorithm indicating that it has finished.
8. go back to 4

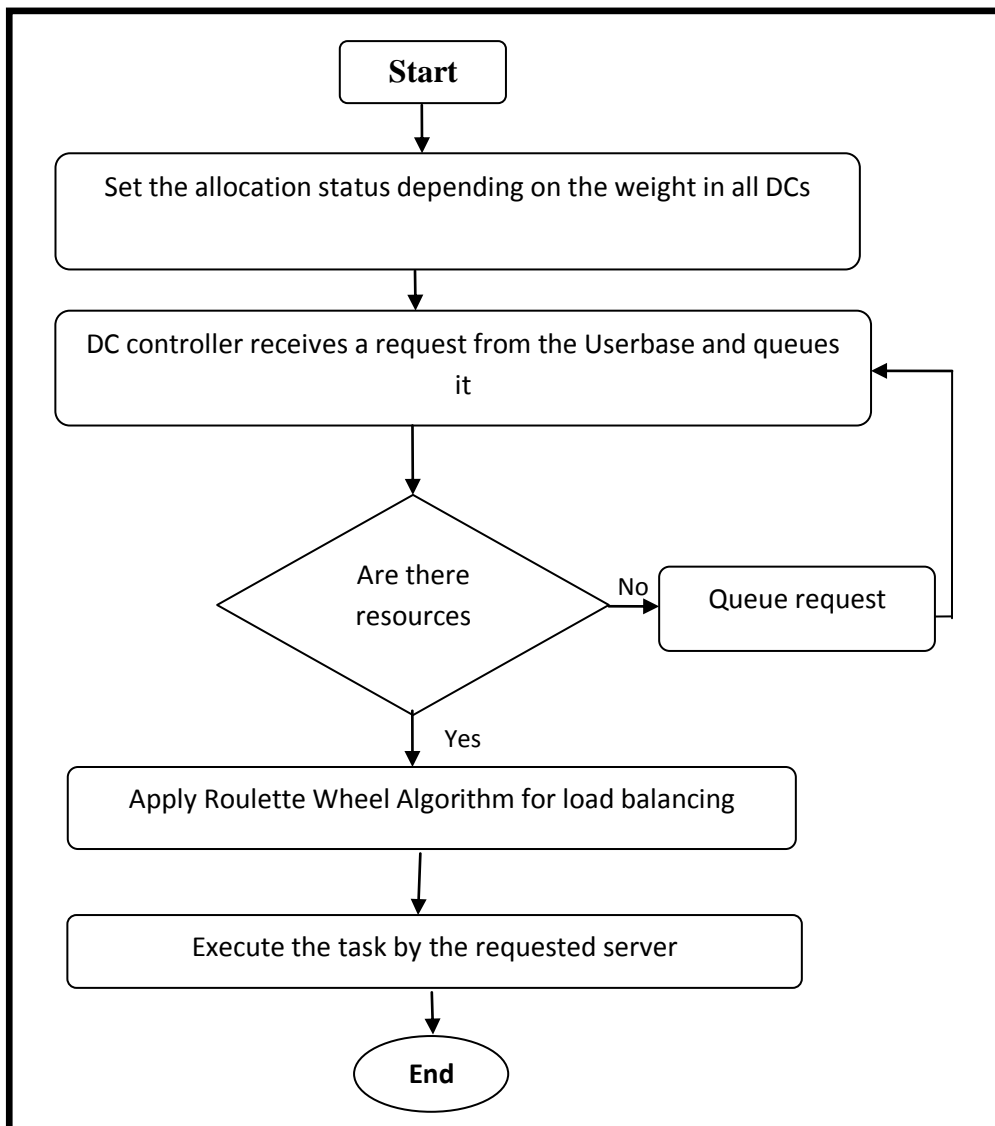


Figure 3: Flow diagram of roulette wheel dynamic load balancing algorithm in cloud computing

VI. Experiments And Results

In this paper, we use the CloudAnalyst simulation tool to execute and analyze the Roulette Wheel Selection Algorithm with Virtual Machine Weight. Using the CloudAnalyst tool, various components and parameter need to be configured such as Userbase configuration, application deployment configuration, data center configuration and load balancing policy. The same configuration has been used for each load balancing algorithm. Average response time (table 1 and fig. 2), data center service time (table 2 and fig. 3) and Data Center Data Transfer Cost (table 3 and fig. 4) are shown.

Table 1: Average response time using the different load balancing policies

UBs	Round Robin	Throttled	ESCEL	RWSALB
UB1	65.69	65.70	65.70	66.35
UB2	57.65	57.63	57.66	58.03
UB3	314.06	313.58	313.79	314.27
UB4	55.80	55.82	55.87	56.12
UB5	511.57	511.73	512.35	512.58
UB6	53.70	53.72	53.66	53.50

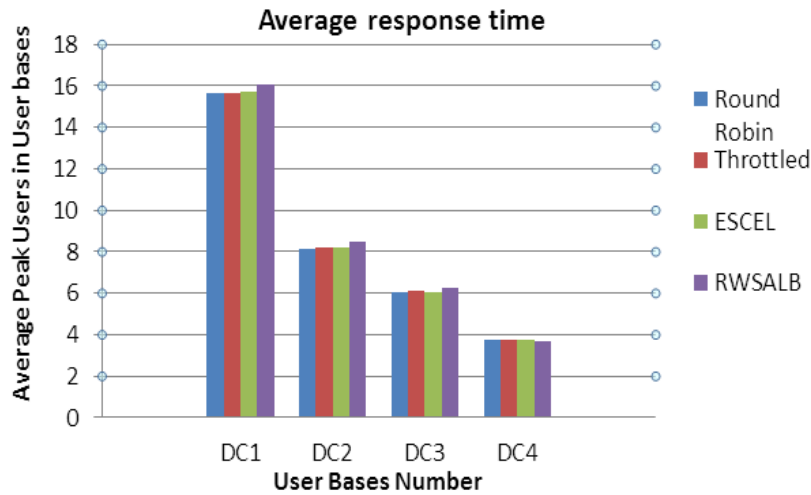


Figure 4 : Results showing the average peak users vs. average response time

Table 2: Average Data Center Request Servicing Times

DCs	Round Robin	Throttled	ESCEL	RWSALB
DC1	15.67	15.65	15.69	16.04
DC2	8.16	8.18	8.17	8.51
DC3	6.04	6.10	6.04	6.28
DC4	3.74	3.78	3.77	3.71

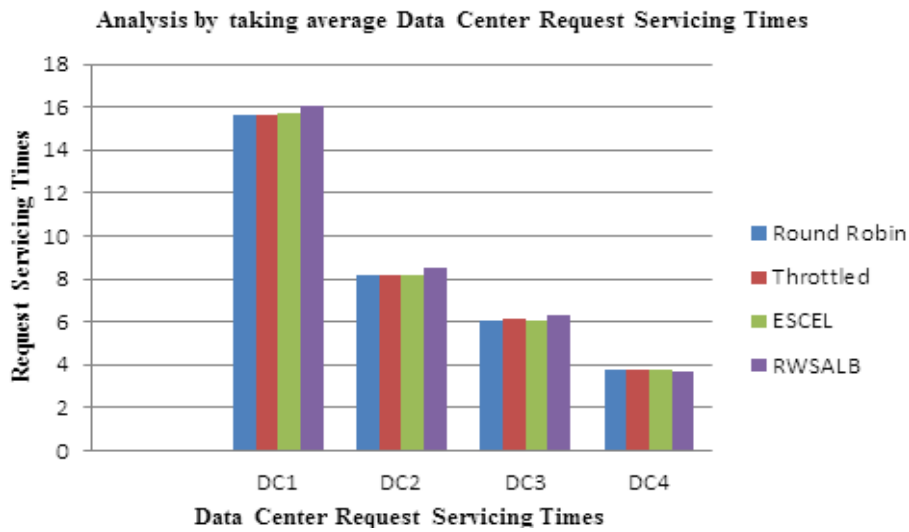


Figure 5: Average Data Center Request Servicing Times

Table 3: Data Center Data Transfer Cost (\$)

DCs	Round Robin	Throttled	ESCEL	RWSALB
DC1	0.78	0.78	0.76	0.12
DC2	0.36	0.36	0.36	0.36
DC3	0.32	0.33	0.34	0.32
DC4	0.12	0.12	0.12	0.12

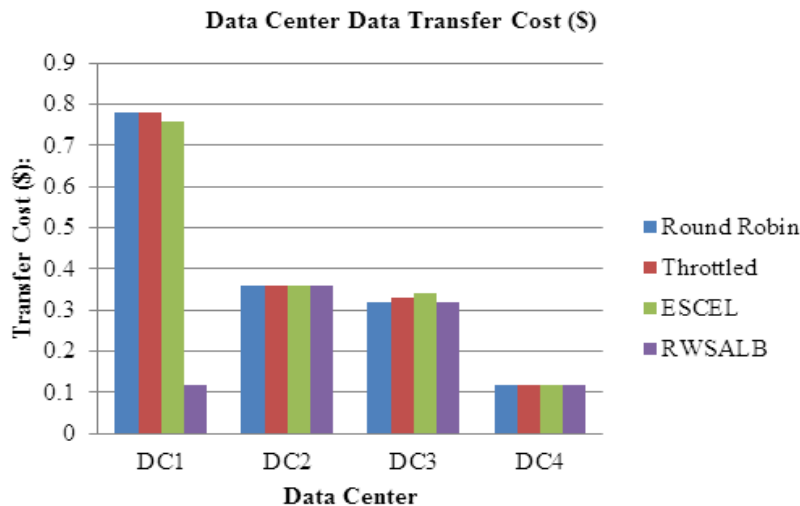


Figure 4: Data Center Data Transfer Cost (\$)

VII. Conclusion

In this research, we simulated four different load balancing algorithms including the Roulette Wheel Selection Algorithm. The average response time, data center service time and data transfer cost are computed for each algorithm. The Roulette Wheel Selection Algorithm has lower data transfer cost than other algorithms.. In the future, we will try to enhance this algorithm by applying some new parameters that increase the performance of the roulette wheel selection algorithm using virtual machine weight for load balancing in cloud computing.

References

- [1]. Alakeel, A. M. (2010). A Guide to Dynamic Load Balancing in Distributed Computer System. IJCSNS, VOL.10 No.6, pp. 153-160.
- [2]. CHANA, N. J. (2012, February 15). EXISTING LOAD BALANCING TECHNIQUES IN CLOUD COMPUTING: A SYSTEMATIC. Volume 3(Issue 1).
- [3]. Marinescu, D. C. (2012). Cloud Computing: Theory and Practice (Vol. 1). University of Central Florida, Orlando, FL 32816, USA: Newnes, 2013.
- [4]. Meenakshi Sharma, P. S. (2012). Efficient Load Balancing Algorithm in VM Cloud Environment. IJCSST, 3, pp. 439-441.
- [5]. Nidhi Jain Kansal, I. C. (2012, January). Cloud Load Balancing Techniques : A Step Towards Green Computing. IJCSI, No 1(Issue 1), 9.
- [6]. Pragati Priyadarshinee, P. J. (2012, June). Load Balancing and Parallelism in Cloud Computing. Volume-1(, Issue-5).
- [7]. S, R. K., V, S., & M, V. (2012). Enhanced Load Balancing Approach to Avoid Deadlocks in Cloud.
- [8]. Sarkar, S. R. (2012, October). EXECUTION ANALYSIS OF LOAD BALANCING ALGORITHMS IN CLOUD COMPUTING ENVIRONMENT. International Journal on Cloud Computing, 2(5), 1-13.
- [9]. Zenon Chaczko, V. M. (2011). Availability and Load Balancing in Cloud Computing. IPCSIT. vol.14. Singapore: IACSIT Press.
- [10]. Apte, A. A. (2012, August). A Comparative Performance Analysis of Load Balancing Algorithms in Distributed System using Qualitative Parameters. International Journal of Recent Technology and Engineering (IJRTE), 1(3), 175 - 179.
- [11]. Bhatiya Wickremasinghe, R. N. (20-23 April 2010). CloudAnalyst: A CloudSim-based Visual Modeller for Analysing Cloud Computing Environments and Applications. Advanced Information Networking and Applications (AINA) (pp. 446 - 452). Perth, WA : IEEE.
- [12]. Chandrasekaran K, a. U. (2013). Load Balancing of Virtual Machine Resources in Cloud Using Genetic Algorithm. National Institute of Technology Karnataka, 156-168.
- [13]. Gupta, R. (2014, February). Review on Existing Load Balancing Techniques of Cloud Computing. Volume 4(Issue 2), 168 - 171.
- [14]. Jain, R. M. (2013, September-October). Study and Comparison of CloudSim Simulators in the Cloud Computing. The SIJ Transactions on Computer Science Engineering & its Applications, Vol. 1(No. 4), 111-115.
- [15]. Jasmin James, B. V. (2012, Sep). EFFICIENT VM LOAD BALANCING ALGORITHM FOR A CLOUD COMPUTING ENVIRONMENT. International Journal on Computer Science and Engineering (IJCSE), Vol. 4(No. 09), 1658 - 1664.
- [16]. Komal Mahajan, A. M. (2013). Round Robin with Server Affinity: A VM Load Balancing Algorithm for Cloud Based Infrastructure. 2013 KIPS, Vol.9, pp. 379 - 394.
- [17]. Kumar, R. (2012, August). Blending Roulette Wheel Selection & Rank Selection in Genetic Algorithms. International Journal of Machine Learning and Computing, 2(4), 365-370.
- [18]. Mayanka Katyal, a. A. (2013, December). A Comparative Study of Load Balancing Algorithms in Cloud Computing Environment. International Journal of Distributed and Cloud Computing, 1(2), 5-14.
- [19]. Ms.NITIKA, M. M. (2012, May). Comparative Analysis of Load Balancing Algorithms in Cloud Computing. Volume 1(Issue 3), 120-125.
- [20]. P. Beulah Soundarabai, S. R. (2012, December). COMPARATIVE STUDY ON LOAD BALANCING TECHNIQUES IN DISTRIBUTED SYSTEMS. International Journal of Information Technology and Knowledge Management, 6(1), 53 - 60.
- [21]. Rajwinder Kaur, a. P. (2014). Load Balancing in Cloud Computing. Association of Computer Electronics and Electrical Engineers (pp. 374 - 382). Proc. of Int. Conf. on Recent Trends in Information, Telecommunication and Computing, ITC.
- [22]. Subashish Mohapatra, K. R. (2013, April). A Comparison of Four Popular Heuristics for Load Balancing of Virtual Machines in Cloud Computing. International Journal of Computer Applications, Volume 68(No.6), 33 - 38.
- [23]. Tejinder Sharma, V. K. (2013, March). Efficient and Enhanced Algorithm in Cloud Computing. International Journal of Soft Computing and Engineering (IJSCE), Volume-3(Issue-1), 385-390.