International Journal of Scientific & Engineering Research Volume 9, Issue 3, March-2018
ISSN 2229-5518

AN IMPROVED ROUND ROBIN LOAD BALANCING ALGORITHM IN CLOUD COMPUTING USING AVERAGE BURST TIME

[1]Abdulrahman Abdulkarim [2]Souley Boukari [3]Ishaq Muhammed [3]Fatima Ahmed Abubakar

1. M.sc Student, Department of Mathematical Sciences, Abubakar Tafawa Balewa University Bauchi, Bauchi State, Nigeria
2. Professor, Department of Mathematical Sciences, Abubakar Tafawa Balewa University Bauchi, Bauchi State, Nigeria
3. Department of Computer Science, Federal Polytechnic Bauchi, Bauchi State Nigeria.

ABSTRACT

Cloud Computing is the collection of large number of resources like hardware and software that are provided by the cloud providers to the consumers as a service over the internet. In cloud computing every task requires to be executed by available resource to achieve minimum waiting time, reduce makespan, best performance and maximum utilization of resources. Considering the growing importance of cloud, finding new way to improve cloud services is an area of concern and research focus Therefore, an efficient task scheduling in the cloud would be required to improve the performance of the cloud. In this paper, a model was proposed that seeks to improve load balancing algorithm for virtual machine load balancing. A method by which average burst time would be used as the time quantum for the round robin load balancing algorithm is also proposed for more effective time sharing. These might result to a more effective load balancing and therefore achieving a better response time of tasks, data center processing time and overall throughput.

Keywords- Cloud computing, Round robin, Virtual Machine, Load balancing, Burst time, Time quantum, Response time.

1. INTRODUCTION

Cloud computing consists of a cluster of computing resources that are delivered over a network, which is accomplished by utilizing virtualization technologies to consolidate and allocate resources suitable for various different software applications. [3] defined cloud computing as a parallel and distributed computing system consisting of a collection of inter-connected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service level agreement established through negotiation between the service provider and consumers. According to [1], Cloud computing refers to both the application delivered as services over the internet and the hardware and software of systems in the datacenters that provide these services. The primary goal of cloud computing is to provide well organized way into remote and geographically distributed resources with the help of virtualization service [12]. Scheduling is the process of allocating tasks to available resources on the basis of tasks' qualities and need [9]. Scheduling can be done at task level, resource level or workflow level. Scheduling is performed on the basis of different parameters so that it increases the overall cloud performance [13]. Users send request to the datacenter for computing job named task. A task is a small piece of work that should be executed within a given period of time [13]. A task may include entering data, processing, accessing software or storage functions. Task Scheduling of cloud computing refers to dispatch the computing tasks

to resource pooling between different resource users according to certain rules of resource use under a given cloud circumstances [7]. It has been about for many years and provides essential provision for both grid computing and cloud computing to growth the system performance and competence [15]. It dispatches the tasks provided by the cloud to the cloud provider on available resources. The aim of task scheduling is to assign tasks to available resources such that the overall time compulsory to execute all tasks is minimized. Round Robin scheduling Algorithm is one of the task scheduling algorithm mostly used in cloud computing. The reason for this is that round robin algorithm employs time-sharing, giving each job a time slot or quantum thus result to be less complexity and maintains fairly load balancing [13].

Although, round robin scheduling algorithm is one of the standard and widely used algorithm in cloud computing, it has a potential drawback. The performance level of round robin scheduling on cloud depends on the slice/quantum time [18]. Based on the work by [14], round robin scheduling approach was selected for VM load balancing on the cloud environment but without a specific method of assigning slice/time quantum. In this paper, an improvement would be made to Round Robin Load Balancing Algorithm by [14] that would ensure an efficient scheduling of tasks by considering a better way to determine a suitable time quantum for more effective time sharing.

The remaining parts of this paper are organized as follows: Section 2 presents some related work and overview on the related work. Section 3 presents the methodology where analysis of the existing algorithm would be discussed and presentation of the proposed algorithm. Section 4 is the discussion of the proposed algorithm. Finally, Section 5 concludes the paper and presents future work.

2. RELATED WORK

In this section, previous work on scheduling algorithms on cloud computing are overviewed, some of the important considerations in performance comparison considered by the previous work are identified.

In [6], the authors have proposed a task scheduling algorithm based on task length and speed of VMs. They have designed a rank based task scheduler which effectively utilizes resources and provides high performance. This algorithm has been tested using CloudSim toolkit with varying lengths of tasks and varying MIPS (speed) of VMs. The results have shown that Rank based task scheduler gives high performance than existing space-shared and time shared task schedulers.

The authors of [4] proposed the Genetic Algorithm (G.A) which thrives to balance the load of the cloud infrastructure while trying minimizing the make span of a given task set. They find the problems in workload across the multiple nodes in the cloud environment and because of that resource utilization is not done properly. So they try to solve the load balancing problem using the task scheduling. Using the proposed G.A, the searching process has become more optimized and effective.

[10] proposed a work on load balancing in cloud computing environments using honey bee behavior. The proposed method achieves good load balancing across virtual machines which maximizes the throughput. Load is balanced according to the priorities of tasks so that waiting time of tasks in the queue is minimal. Compared to other traditional methods, this method is effective and overall execution time is improved.

The work of [2] have proposed a task scheduling technique for cloud computing. A new strategy of task scheduling is proposed in view of the aggregate order for allocation of resource to enhance the algorithm of Min-Min. The main concentration is on minimizing the aggregate executing time (makespan) of scheduling of task and expanding the utilization of assets. The results of their work shows that the proposed approach allows more versatile assets distribution for free occupations booking in the distributed computing environment.

[11] proposed a cloud task scheduling policy based on Load Balancing Ant Colony Optimization (LBACO) algorithm. The main contribution of this work is to balance the entire system load while trying to minimize the makespan of a given tasks set. The LBACO is inherited from the Ant Colony Optimization (ACO). They can carry out new task scheduling depending on the result in the past task scheduling so it is very helpful in the cloud environment. The scheduling problem aims to minimize the total execution time of the tasks as well as to achieve a well-balanced load across all VMs in the cloud. They consider that all the tasks are independent.

In [17], the authors developed an approach for scheduling workflow tasks over the available resources of cloud that minimized the execution time and execution cost under the given deadline and budget constraint. In their work, workflow tasks are executed in order of their priority which is basically computed using bottom level. The assigned priority is then used to initialize the Particle Swamp Optimization (PSO). After assigning the priority the tasks are sorted according to the descending order of bottom level. The tasks are then sent to different processors according to their order of execution for completing the workflow application. Experiment results shows that this algorithm has a promising performance when compared with PSO.

Also [16] proposed the approach of improved cost-based scheduling algorithm. The main objective of their work is to schedule groups of tasks in cloud computing platform, where resources are having different resource costs and different computation performance. When grouping of jobs is done, communication between jobs and resources optimizes computation/communication ratio. This algorithm measured performance of computation and cost of resources. This also increased the execution of tasks, transfer of data between tasks ratio by combining various tasks during execution. The process of combining task is usually done after analyzing the capability of different available resources and its processing. CloudSim has been used for performing the simulation. Results of their work show that for their algorithm, time taken to complete tasks after grouping is very less as compared to when grouping is not done.

2.1 Load Balancing

Many efforts have been made to provide a good load balancing structure. Load balancing in clouds is a technique that distributes the excess dynamic local workload evenly across all the nodes. It is used for achieving a better service provisioning and resource utilization ratio, hence improving the overall performance of the system [5]. Incoming tasks coming from different location are received by the load balancer and then distributed to the data center for the proper load distribution.

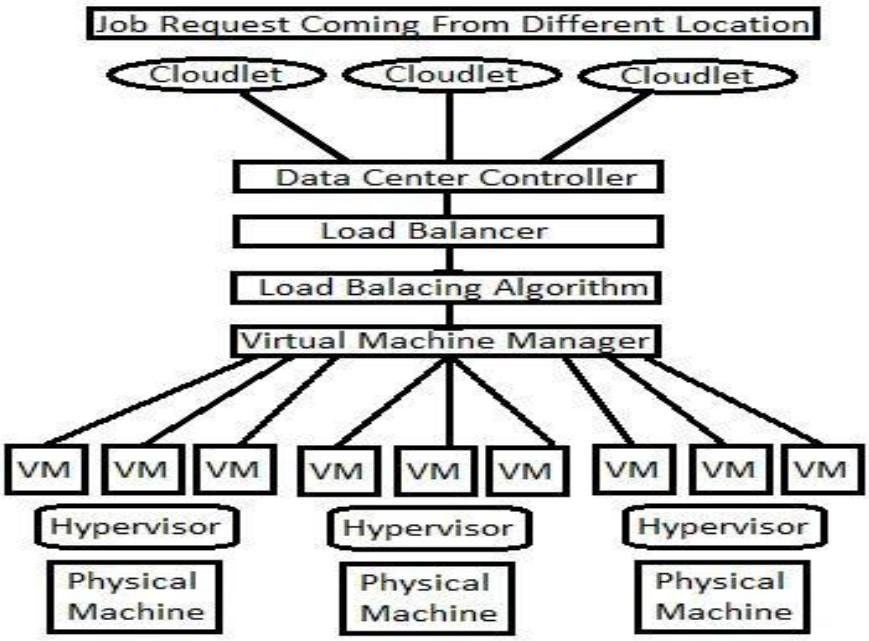

Figure 1. Load Balancing Structure.[5]

An overview of related works on various task scheduling algorithms revealed that there are still lots of improvements that can be carried out. The major issues being observed on most proposed task scheduling algorithms are their ability to achieve better response time, makespan, cost, resource allocation, and energy consumption. Therefore, this paper would consider the work by [14] which would focus on improving the round robin load balancing algorithm to address the issue of time quantum so as to effectively utilize the principle of time sharing which may lead to better resource utilization.

3. METHODOLOGY

3.1 Analysis of Existing Algorithm

Based on the algorithm by [14], requests are being assigned to VM using round robin approach provided that the request has never been assigned to a virtual machine. That is, it's allocate a VM to a task (request) using round robin algorithm. The shortfall of this approach is that the method in which assigning time quantum for the time sharing principle is not specified.

Algorithm 1: Round robin load balancing algorithm by [14]

hash_map[entries for the last VM allocated to a request from a given user base]
VM_state[allocation status of each VM]
Round_robin_load_balancing_algorithm()
{
Initialize all the VM allocation status to AVAILABLE in the VM state list;
Initialize hash map with no entries;
While (new request are received by the Data Centre Controller)
Do {
Data Center Controller queue the requests;
Data Centre Controller removes a request from the beginning of the queue;

If (hash map contain any entry of a VM corresponding to the current requesting user base request && VM allocation status == AVAILABLE)
{
The VM is reallocated to the user base request;
}
Else
{
Allocate a VM to the user base request using Round Robin Algorithm;
Update the entry of the user base and the VM in the hash map and the VM state list;
}
}

3.2 PROPOSED ALGORITHM

The improvement is the addition of a technique to get a specified time quantum for an effective time sharing principle that the existing algorithm by [14] lacks. Therefore, in this approach, allocation of VM to requests would be using an improved round robin algorithm in which method of assigning time quantum is based on average burst time adopted by [8]. A pseudocode of the proposed model is given in algorithm 2

Algorithm 2

hash_map[entries for the last VM allocated to a request from a given user base]
VM_state[allocation status of each VM]
Algorithm()
{
Initialize all the VM allocation status to AVAILABLE;
Initialize hash map = null;
While (new request are received)
Do {
Data Center Controller queue requests;
Data Centre Controller removes a request from the beginning of queue;
If (hash map contain any entry of a VM corresponding to the current requesting user base && VM allocation status == AVAILABLE)
{
The VM is reallocated to the user base request;
}
Else
{
Initialize ready_queue, BT_list, TQ, VM_state, round;
While (ready_queue != NULL)
{
TQ= avg (BT_list);
Allocate VM to request in ready_queue,
Set VM_state to BUSY;
Execute request;
Deallocate VM and set VM_state = AVAILABLE;
round++;
New BT=BT-TQ;
}

Update the entry of the user base and the VM in the hash map and the VM state list;

}

}

4. DISCUSSION OF PROPOSED ALGORITHM

The block of algorithm added as improvement to the existing algorithm works in such a way that ready_queue holds set of tasks (requests) to be assigned to a VM. Therefore, the program would execute if there are available tasks to be assigned. Time quantum is set to be equal to the average of burst time of requests for context switching which is declared as the variable 'round'. The first request in the ready_queue would be assign a VM, there the VM is allocated to execute the request and the state of the VM is changed to Busy. After request is being executed, VM is de-allocated and the state changed back to Available. Then we have a context switch with a new burst time for the request.

4.1 Performance Parameters

There are some performance parameters that would be considered in this work. These parameters are the response time, data center processing time and throughput. These parameters would be evaluated based on both the round robin load balancing algorithm by [14] and the improved algorithm for performance analysis.

4.1.1 Response time

The elapsed time between the end of an inquiry or demand on a computer system and the beginning of a response.

4.1.2 Data Center Processing time

The exact time taken to complete the execution of a given task in the data center. The ultimate goal of any scheduling algorithm is minimizing the processing time.

4.1.3 Throughput

It is the number of task executed in the fixed interval of time. To improve the performance of the system, throughput should be high.

5. CONCLUSION AND FUTURE WORK

This paper proposed an improvement to the round robin load balancing algorithm by [14], where by a method was adopted to use average burst time of requests as the time quantum for the time sharing principle on the round robin algorithm. It is expected that this work would result to more effective load balancing by achieving a minimal response time of user base tasks, minimal data center processing time, increase overall throughput and therefore results to better resource utilization on the cloud.

In a future work, the improved algorithm would be implemented using cloudAnalyst toolkit whereby an evaluation on the performance of the improved algorithm using user base response time, data center processing time and the overall throughput would be carried out against the algorithm by [14].

REFERENCES

[1] Armbrust, M., Fox, A., Joseph, A., Randy, K., Andy, K., Gunho, L., . . . Ariel, R. (2010). A view of Cloud computing. *Communications of the ACM*, 50-58.

[2] Bey, K. B., Benhammadi, F., & Benaissa, R. (2015). Balancing Heuristic for Independent Task Scheduling in Cloud Computing. *12th International Symposium on Programming and Systems*, 7-12.

[3] Buyya, R., Venugopal, S., Brogberg, J., Yeo, C. S., & Brandic, I. (2009). Cloud Computing and Emerging IT Platforms: Vision, hype, and reality for Delivering Computing as 5th Utility. *Future Generation Computer Systems, Volume 25, No 6*, 559-616.

[4] Dasgupta, K., Mandal, B., Dutta, P., Kumar, J., & Dam, S. (2013). A Genetic Algorithm based Load Balancing Strategy for Cloud. *Computational Intelligence: Modelling Techniques and Applications* (pp. 340-347). Elsevier.

[5] Dharmesh, K., & Jaydeep, V. (2014). A Survey of various Load Balancing Algorithm In Cloud Computing. *International Journal of Scientific & Technology Research*, 115-119.

[6] Ettikyala, K., & Latha, Y. N. (2016). Rank Based Efficient Task Scheduling for Cloud Computing. *IEEE International Conference on Data Mining and Advanced Computing*, 343-346.

[7] Fazel, M., Shahram, J., & Masoud, B. (2014). Survey on Job Scheduling Algorithm in Cloud Computing. *Internation Journal of Emerging Trends & Technology in Computer Science, Volume 3, Issue 2*, 151-154.

[8] Gupta, A., & Nagpal, I. (2015). Revised Round Robin Scheduling Algorithm. *International Journal of Computer Science and Information Technology Reserach, Volume 3, Issue 3*, 164-168.

[9] Jangra, A., & Saini, T. (2013). Scheduling Otimization in Cloud Computing. *Internation Journal of Advance Research in Computer Science and Software Engineering, Volume 3, Issue 4*, 62-65.

[10] Krishna, P. V. (2013). Honey Bee Behavior Inspired Load Balancing of Task in Cloud Computing Environments. *Applied Software Computing* , 2292-2303.

[11] Li, K., Xu, G., Zhao, G., Dong, Y., & Wang, D. (2011). Cloud Task Scheduling based on Load Balancing Ant Colony Optimization. *Chinagrid Conference.* China: IEEE.

[12] Maniyar, B., & Kanani, B. (2015). Review on Round Robin Algorithm for Task Scheduling in Cloud Computing. *Journal of Emerging Technologies and Innovative Research, Volume 2, Issue 2*, 788-793.

[13] Mathew, T., Sekaran, K. C., & Jose, J. (2014). Study and Analysis of Various Task Scheduling Algorithm in Cloud Computing Environment. *International Conference on Advances in Computing* , 658-664.

[14] Nusrat, P., Amit , A., & Ravi, R. (2014). Round Robin Approach for VM Load Balancing Algorithm in Cloud Computing Environment. *Internation Journal of Advanced Research in Computer Science and Software Engineering* , 34-39.

[15] Patil, S., Rekha, A. K., Balaji, N., & Patil, S. H. (2015). Performance improvement in cloud computing through dynamic task scheduling algorithm . *Next Generation Computing Technologies (NGCT).* IEEE.

[16] Selvarani, S., & Sadhasivam, G. S. (2010). Improved cost-based algorithm for task scheduling in cloud computing. *Computational Intelligence and computing research*, 1-5.

[17] Verma, A., & Kaushal, S. (2014). Bi-Criteria Priority based Particle Swarm Optimization Workflow Scheduling for Cloud. *RAECS UIET*, (pp. 6-8). Chandigarh.

[18] Zannon, N., & Rawshdeh, D. (2015). STASR: A New Task Scheduling Algorithm for Cloud Environment. *Network Protocols and Algorithms* (pp. 81-95). Macrothink Institude.