Minimization Response Time Task scheduling Algorithm

Maria Maqsood¹, Saima Anwar Lashari², Murtaja Ali Saare³, Sari Ali Sari⁴, Yaqdhan Mahmood Hussein⁵, Hatem Oday Hatem⁶

¹Department of Computer Science, University of Lahore, Sargodha Campus, Pakistan.
²College of Computing and Informatics, Saudi Electronic University, Dammam, Kingdom of Saudi Arabia.
³Human-Centered Computing Research Lab, School of Computing, Universiti Utara Malaysia, Sintok, 06010, Kedah, Malaysia
⁴Faculty of Computer Science and Information Technology Universiti Tun Hussein Onn Malaysia, 86400 Parit Raja, Batu Pahat, Johor, Malaysia.
⁵⁶Communication Engineering Department, Faculty of Electrical Engineering, University Technology Malaysia, 81310 Johor Bahru, Johor, Malaysia

s.lashari@seu.edu.sa

Abstract. Cloud computing is a type of computing that base on shared computer resources for delivering information technology services. Distributed Computational system has been considered as best systematic model for sharing resources. Minimization of response time is the requirement in the distributed computational system. Previous Algorithm belongs to the optimization in which every new Virtual Machine reaching as a request to the cloud computing manager. This creates disturbance in the processing during life cycle of others Virtual Machines. For overcome this problem, there is a proposed an algorithm for minimization response time in distributed computational system by using the multi agents cloud structure. The overall system is architecture as a distributed over multi layered model. This leads to a situation that demands high care in allocating a Virtual Machine (VM) to a server machine in order to improve allocation time and efficiency of resources. Focus of this work remains to decrease time of allocation to optimize response time using a Virtual Machine (VM) placement and allocation algorithm that accounts Virtual Machines (VM), their types and available resources among the data center server machines. Presented algorithm is implemented by using cloud simulator tool, through this tool, the calculated results of presented algorithm have been quite better with respect to traditional algorithm, and the difference presented with graphs for the both algorithms. Results support proposed methodology.

1. Introduction
Cloud computing is a type of computing technique which plays an important role for increasing the efficiency of data sharing and storage capacity for the business in the world. Cloud computing improves the capabilities of services like as IaaS (Infrastructure as a service), Paas (Platform as a service) and Saas (Software as a service) but there are many security and management problems that are faces by Cloud computing [1]. With increasing advancement of information technology, the
Computer has become the essential utility of human life [2]. Cloud computing is introducing the capability for data access, interconnection between resources and exchangeable data with different resources through cloud computing [3]. For organizations and enterprises data sharing is available within low cost and with high capacity that is provided by new technology of cloud computing. Cloud computing is open base environment that provide the facility of sharing in which main aspect of security issues during sharing of resources [4]. In past years many methodologies are presented for achieve the high quality of service from server to client in the cloud infrastructure with fast response time in different ways. Such as traditional methods is bin package algorithm for placement of VM to host. Earlier algorithms belong to the optimization of every new virtual machine reaching as a request to the cloud computing manager. This disturbs the processing during the life cycle of a virtual machine. Traditional policies has not been contained such method that accounts time optimization in placement policies of VM. This situation needs improvement of design towards efficient methodologies that can lead to a dynamic computing technological improvement. A standard cloud terminology adopted by Cloud simulator tool which is used for executed with different specifications of virtual machine with modified algorithm. Figure 1. presents the cloud computing infrastructure.

![Cloud Computing](image)

**Figure 1 Cloud Computing**

2. Cloud Services Models

There are three types of cloud computing service models.

1. Software as a service (SaaS)
2. Platform as a service (PaaS)
3. Infrastructure as a service (IaaS)

2.2.1 Software as a Service (SaaS)

User uses the services of cloud computing by executing their software and application properly with its subscription [5]. The other option is that user accesses the services of cloud computing through Software as a Service (SaaS) without not knowing about the architecture and backend of the applications and software’s that run behind the frontend. User does not concern with the installation and maintains of application [6]. The main advantage and rapidly increasing use of Software as a Service (SaaS) is its scalability, easy configuration and compatibility because there is no need user to concern with the hardware and software of applications. If we compare with traditional system the user faces many problems of configurations and there is no level of scalability, compatibility and accessibility [7]. Example of Software as a Service (SaaS) providers are Google Docs, Microsoft Office 36, Amazon etc [8].

2.2.2 Platform as a Service (PaaS)

The Configurations, implementation and maintenance of applications, Platform as a service (PaaS) is an essential part in the cloud computing for application developers to maintain their application. Service providers need Platform as a service (PaaS) to maintain the whole services of cloud computing so there is necessary to implement the Platform as a service (PaaS) for successfully execution of services for cloud computing [9]. Service provider provide platform which enable the user to access the software, operating system, servers, development techniques and the database but all
of this there is no too used to need of concern of user about the hardware of system like as processing unit, memory and storage etc. Platform as a service (PaaS) allow the application developers to configure and implementation of applications on cloud. The differentiation between SaaS and PaaS is that Platform as a service PaaS provides full life cycle for implementation of software and user can easily implemented the software directly on cloud while by using the SaaS, user can just host the executed applications [10].

2.2.3 Infrastructure as a Service (IaaS)
The third layer of service model is Infrastructure as a service (IaaS). IaaS deal with hardware resources and infrastructure such as storage, processing unit and connectivity of computer in the network. For execution of application there is need of infrastructure that is provided by IaaS. Infrastructure as a service (IaaS) is completely new concept of virtualization that only provides an infrastructure for application execution [11]. Infrastructure as a service (IaaS) work like a middle layer between two services which act as a physical or a virtualization machine for provide the infrastructure to help the user for execution the application. Infrastructure as a service (IaaS) enable the user to execute the application without taking the responsibility of implementation, administration and maintains. Example includes Amazon EC2 of IaaS [12].

3. Related Work
There is critical issue in cloud computing of management of resources and for the fast response time the placement of multiple virtual machines. For gaining the better results, many efforts and techniques are applying for solving this problem. Virtual machines work on the physical layer which is share available resources for the better productivity and effectiveness of computing resources. For solving the virtual machines placement problem the approached which has been modeled as Constraint Satisfaction Problem to increases the effectiveness of computing resources. With solving the virtual machine placement problem, the main focus on decreasing the cost of using the Computing resources [13]. These approaches focus is not on performance of network and effectiveness of computing resources that has a great impact on the whole effectiveness and efficiency of cloud architecture layers. For improving the response time and quality of service there is necessary to manage the computing resources and virtual machines in best way. In the case of migration of virtual machines few approaches [14][15][16][17] are declaring for gaining the effectiveness and efficiency of performance from one physical machine to another. Some approaches which focus on the performance of layers and response time of cloud for clients with the high quality of service. However, the existing approach is heavily focused on increasing the response time from cloud to client with managing the quality of service in better way. The drawback is that migration of virtual machines and their placement taking time has affected the response time to the front end and also effect on the quality of service. In the existing approach the algorithm is managing the placement of virtual machine in such a way one new appointed virtual machine does not disturb the life cycle of already working virtual machine. As the result the response time is increased from lower layer to upper layer and also it has good effect on the quality of service.

4. Traditional Scheduling Algorithm
A Virtual Machine allocation policy performs as an outline for provisioning process. The detail of the algorithm is given in the following lines. This is a one dimensional policy that accounts one Virtual Machines specification given in the Virtual Machine creation demand sent by the user to Data Center Manager. Data Center Manager further enlists the host that fulfills the demand of mentioned policy. This policy further tries to create a Virtual Machine after it finds a proper host. Output of this algorithm is either true or false depending upon availability of resources.

Step1: If (Not VM)
Step2: Do
Step3: For each (host_i in Hosts);
Step4: Search_Host(Demanded_Processing_cores);
Step5: End For;
Step6: Results_list = Index_of (host_i, j, k……);
Step 7: Counter = Counter +1;
Step 8: If (Results_list)
Step 9: Modify;
Step 10: VM_Table (VM);
Step 11: Set_Used_Processing_Cores(demanded_processing_cores);
Step 12: Set_free_Processing_Cores(demanded_processing_cores);
Step 13: Acknowledge_Flag = True;
Step 14: End if;
Step 15: Else host=host (where host index is minimum);
Step 16: While(Results_flag=False && Counter<demanded_processing_cores);
Step 17: Return Acknowledge_Flag

5. Proposed Response Time Minimization Algorithm
Step 1: If (!VM_Created)
Step 2: Do: Send_Vm_Specs(DCM);
Step 3: For Each(host i in Hosts)
Step 4: If(Host i>= Processing_Cores_Req && Host i >= Req_Mips)
Step 5: Calculate_remaining_capacity(host i)
Step 6: Calculate_Serving_ratio(host i)/after current VM
Step 7: If(remaining capacity>serving ratio)
Step 8: Index=host_index;
Step 9: Remaining_capacity=serving_ratio;
Step 10: End If
Step 11: End if
Step 12: End for
Step 13: If (! (index)
Step 14: Break;
Step 15: End if
Step 16: Else
Step 17: Host_list=index
Step 18: Acknowledge_Flag=create_VM
Step 19: Try_counter++; 
Step 20: If(Acknowledge_flag)
Step 21: Update:
Step 22: Vm_table(VM i)
Step 23: Scheduler_Queue(VM i)
Step 24: Used_processors(Required_VM_specs);
Step 25: Free_Cores(Used_processors)
Step 26: Acknowledge_Flag=true
Step 27: Break;
Step 28: Else
Step 29: Free_cores(main_value)
Step 30: Used_processors(main_value)
Step 31: Remaining_capacity(main_value)
Step 32: End if
Step 33: While(Result && try_counter<free_cores)
Step 34: End if
Step 35: Return Acknowledge_flag

6. Experimental Setup
CloudSim is a simulator that supports simulation of cloud data center using java based classes. It supports two decision based function, one works as coordinator of data center that is only one for each data center. And the second one represent host. Second agent function is associated exactly one with each host. Agent layers and their communication can be ordered and maintain but it increases the complexity of simulation. To keep the simulation simple, two layers architecture is used for the
purpose of simulation. CloudSim allows the pooling of resources by using intelligent allocation at the
time of Virtual Machine creation.

In figure 2, the distributed Computational system show with multi agents dividing them into layers. For the purpose of simulation, the configurations specification that is used in implementation show in the table 1 with quantity.

| Configurations                  | Details                                      |
|---------------------------------|----------------------------------------------|
| No of Data Centers              | 1                                            |
| No of hosts                     | 5                                            |
| Cores on each host:             | 1 (total 5 cores)                            |
| Storage capacity: 1 Tb          | (100 GB=100000 MB)                           |
| Bandwidth:                      | 1 mbps =8000 kbits/sec                       |
| No of Broker instances:         | 1                                            |
| VMs                             | 10                                           |
| Tasks per VM                    | 50                                           |
| Instruction length              | 50000                                       |
| Input size                      | 300 kb                                       |
| Output file size                | 400 kb                                       |
| clock Speed                     | 1000 mips per core                           |
| Storage per VM                  | 20 GB                                        |
| RAM per VM                      | 512MB                                        |
| CPU core per VM                 | 1 virtual core                               |
| MIPS Per VM                     | 1000 mips                                    |
| Bandwidth per VM                | 1000 kb/s                                    |

| Table 2. Specifications of Different Virtual Machine categories |
|---------------------------------------------------------------|
| Spec  | VM 1     | VM 2     |
|-------|----------|----------|
| Storage | 512 mb  | 1024 mb  |
| Ram    | 256 mb   | 512 mb   |
| Cores  | 1        | 1        |
| Mips   | 1000     | 1000     |
| Bandwidth | 512      | 512      |
7. Results

Results are observed in two different ways. Impact of both algorithm (default & modified) are noted down in two different observed variable. Increasing number of VMs has been observed for their impact on Allocation time & resources pooling. The observations are given in the form of graphs. Figure 3. is a graph which shows the difference of time allocation between default VM placement algorithm and modified algorithm.

![Figure 3. Time allocation difference between default algorithm and modified algorithm.](image3)

Figure 3. Graph of time allocation between default (traditional) Virtual Machine placement algorithm and modified algorithm. With respect to response Time, the modified algorithm has been quickly response to the maximum Virtual Machines requests. During the second scenario, different kinds of Virtual Machines (with different specs) are used to test the effect of placement method on allocation time. Four different kinds of specification are defined for Virtual Machines (show in table 6.2) and then their observations are made. Data center configurations are kept same as in the above example.

![Figure 4. Throughput Comparison](image4)

Figure 4. Throughput Comparison

According to the observed results, when number of Virtual Machines requests crossed over 225, the difference between allocation times began to clear. In case of default (traditional) algorithm, the allocation time started rising abruptly as number of demanded resources bypassed number of available resources. On the other hand, agent based scenario with modified algorithm smoothly increased its allocation time with increasing numbers of virtual machines requests. The overall ratio
with increasing the number of virtual machines with respect to time is less than the traditional algorithm that is show in figure 4.

In the case of maximum virtual machine request, the failures of request have been happened in both traditional and modified algorithm but with quite difference.

Figure 5. Ratios of Virtual Machine Request Failures

Figure 5 shows the ratio of Virtual Machines Failures Request. Modified Placement Algorithm is quite less ratio of failures of Virtual Machines Request than traditional Algorithm.

8. Time Complexity of Algorithm

The algorithm’s complexity is dependent on number of Virtual Machine creation requests and the time needed by the creation attempts, time needed for communication and scheduling time. So the Complexity is described as

\[ F(n) = T_c \times n + T_{ca} + T_{com} + T_{sch} \]

Time Complexity = Time of creation * num of virtual machine + time of communication + time of scheduling

\[ F(n) = O(n) \]

All of these are time constants therefore we can say that time complexity of this algorithm is linear.

9. Results of Observations

Different kinds of observations that can be made from simulation include response time and turnaround time. Response time depends on number of factors that include time to send request, number of tries, creation time of VM, time to send acknowledgement. Therefore mathematical notation of these two observations is given in the following.

\[ \text{Response\_Time} = T_{req} + \text{Counter}_{tries} \times T_{vm\_cr} + T_{ack} \]

The second observation that can be made is turnaround time. This is time is dependent on following factors.

\[ T_{ta} = T_{Response\_Time} + T_{cl\_res} + T_{ex} + T_{sr} + T_{dr} + T_{dev\_vm} \]

Here \( T_{ta} = \) turnaround time

\[ T_{Response\_Time} = \text{VM request response when the VM is properly created} \]

\[ T_{cl\_res} = \text{Response Time to cloudlets} \]

\[ T_{ex} = \text{Execution time for cloudlets} \]

\[ T_{sr} = \text{Time to request VM destruction} \]

\[ T_{dr} = \text{Time to send back results} \]
$T_{de}$ = time to destroy a VM Request

$T_{des\_vm}$ = time to destroy a VM.

10. Conclusion and Future Direction

While, in the improvement of response time depends on the interaction with cloud manager layer to the physical layer where multiple virtual machine share computing resources. At every client request, Cloud manger find new virtual machine according to the requirement of client request. Thus, it is necessary to control and manage the placement of virtual machine so that improve the response time and quality of service by selecting virtual machine according to the requirement in the short time. Some Virtual Machines placement approaches that work for distributed layer system in the cloud environment but for every new request arrival, cloud manger find the new virtual machine with considering the requirement of requesting task that disturb the life cycle of busy virtual machines. To address this issue, this paper present a Virtual Machines placement approach focus on the best and in short time selection of Virtual Machines by managing their placement in a way that new virtual machine does not disturb life cycle of working virtual machine. In addition, our approach has also considered the communication between distributed layers in the cloud architecture. Contract net protocol use for communication purpose in the distributed layers. Our simulation on cloud sim tool that proved purposing approach has been improved the task completion time. Results are showing in graphical representation with comparison traditional approach. This study can be further extended by using hybrid algorithm to participate the Virtual Machines migration and data centre joining aspects by considering energy saving.

References

[1] Xue, C.T.S. and Xin, F.T.W., 2016. Benefits and challenges of the adoption of cloud computing in business. International Journal on Cloud Computing: Services and Architecture (IJCCSA), 6(6).

[2] M. A. Saare and A. Hussain., 2018. “Successful Aging among Older Adults Using Computers: a Systematic Review,” Int. J. Eng. Technol., vol. 7, no. 20, pp. 50–54.

[3] Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J. and Brandic, I., 2009. Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. Future Generation computer systems, 25(6), pp.599-616.

[4] P.Dhir and S. Garg., 2017.“Survey on Cloud Computing and Data Masking Techniques”, International Journal of Innovation & Advancement in Computer Science 6, no. 4, pp.1-7.

[5] D. Assante, M. Castro, I .Hamburg and S. Martin., 2016.“The use of cloud computing in SMEs”, Procedia computer science 83, pp.7-12.

[6] K. K. M. Kumar., 2014. “Software as a service for efficient cloud computing”, International Journal of Research in Engineering and Technology, pp. 178-181.

[7] M. Al-Roomi, S. Al-Ebrahimi, S. Buqrais, and I. Ahmad., 2013 "Cloud computing pricing models: a survey," International Journal of Grid and Distributed Computing 6, no. 5, pp. 93-106.

[8] Q.Zhang, L. Cheng and R. Boutaba., 2010. “Cloud computing: state-of-the-art and research challenges”, Journal of internet services and applications 1, no. 1, pp.7-18.

[9] T.Devi, and R. Ganesan., 2015. “Platform-as-a-Service (PaaS): model and security issues”, Indonesian Journal of Electrical Engineering and Computer Science 15, no. 1, pp. 151-161.

[10] S. Kumar, and R.H. Goudar., 2012. “Cloud computing-research issues, challenges, architecture, platforms and applications: A survey”, International Journal of Future Computer and Communication 1, no. 4, pp. 356-360.

[11] G. Kulkarni, R.Sutar and J.Gambhir, “Cloud Computing- Infrastructure as Service – Amazon EC2”, International Journal of Engineering Research and Applications 2(1), pp. 117-125, 2012.

[12] S.R.Aniruddha and D.N. Chaudhari., 2013. “Cloud Computing: Instructure as a Service”, International Journal of Inventive Engineering and Sciences, 1(3), pp. 1-7.

[13] J-L. Kim, and R. D. Ells., 2005. A Framework for Integration Model of Resource-Constrained Scheduling using Genetic Algorithms. Simulation Conference, Proceedings of the Winter, 2119–2126, Orlando, Florida,USA, (December).

[14] S. K. Bose, and S. Sundararajan., 2009. Optimizing Migration of Virtual Machines across Data-Centers. Parallel Processing Workshops, ICPPW '09. International Conference.

[15] T. Hirofuchi, H. Ogawa, H. Nakada, S. Itoh, and S. Sekiguchi., 2008 AStorage Access Mechanism for Wide-Area Live Migration of Virtual Machines. In Summer United Workshops on Parallel, Distributed and Cooperative Processing, pages 19–24.
[16] Y. Luo, B. Zhang, X. Wang, Z. Wang, Y. Sun, and H. Chen., 2008. Live and incremental whole system migration of virtual machine using blockbitmap. In IEEE International Conference on Cluster Computing.

[17] T. Maoz, A. Barak, and L. Amar., 2008. Combining Virtual Machine migration with process migration for HPC on multi-clusters and Grids. Cluster Computing. IEEE International Conference.