Accepted Manuscript

Optimistic virtual machine placement in cloud data centers using queuing approach

Anitha Ponraj

PII: S0167-739X(18)30469-2
DOI: https://doi.org/10.1016/j.future.2018.10.022
Reference: FUTURE 4527

To appear in: *Future Generation Computer Systems*

Received date : 30 March 2018
Revised date : 17 July 2018
Accepted date : 16 October 2018

Please cite this article as: A. Ponraj, Optimistic virtual machine placement in cloud data centers using queuing approach, *Future Generation Computer Systems* (2018), https://doi.org/10.1016/j.future.2018.10.022

Optimistic Virtual Machine Placement in Cloud Data Centers using Queuing Approach

Anitha Ponraj[a]

[a] Assistant Professor , Department of CSE, Sathyabama University Chennai, India

---

Abstract

Cloud computing gives many beneficial services to share large scale of information, storage resources, computing resources, and provide knowledge for research. Cloud users deploy their own applications and related data on a pay-as-you-go basis. Virtual machines (VMs) usually host these data-intensive applications. The performance of these applications often depends on workload types: I/O data-intensive or I/O computation, workload volume, CPU attributes on computing nodes, Virtual machines and the network. Therefore, the application jobs in the workload have different completion times based on the VM placement decision and large data retrieval. The main contribution of this thesis to gain high performance for the applications executed on the cloud by minimizing the completion time, minimizing the production cost and maximizing the throughput of cloud links. To provide a solution for minimizing the overall jobs' completion time (computing time as well as data transferring time) in both static and dynamic workloads, we propose VMs placement algorithm that considers computation resources, Quality of Service (QoS) metrics and virtual machine status and I/O data with priority based probability queuing model. The results obtained by the proposed methodology shows that the proposed optimal VM placement algorithm has a reduced processing cost and completion time compared with the traditional algorithms such as FCFS and priority scheduling.

Keywords: Cloud Computing, Virtual Machine, Completion Time, Processing Cost, Throughput, Scheduling.

---

1. Introduction

Cloud computing can provide resource as services via virtualization technology which provides software environment in the form of virtual machine VM. In cloud computing, applications with operating system, specific hardware, software, and library requirements can be executed in a larger amount of resources by instantiating VMs from a repository so that requirements can be supported. For that reason, to employ VMs as a computing resource can deliver various advantages such as QoS guarantee, performance isolation, easy resource management, and can deploy effective computing environment. Virtualization is one of the more important technologies to impact computing in the last few years. Virtualization has a feature to run multiple operating system instances concurrently on a single computer. The algorithm based on the throughput includes Extended Min-Min, revised ant colony optimization. Earliest deadline, FCFS, Round robin is time based approach. The advantage of this relative study is that as per the requirements of the consumers and service providers they can select the suitable class of scheduling algorithms for different types of services required. This study may further be used for optimization of different algorithms for improved resource management in cloud computing environment. [1]

Quality of Service and response time is attained by executing the high priority jobs deadline based jobs first by approximating job completion time and the priority jobs are produced from the remaining job with the help of Task Scheduler. They proposed three scheduling algorithm First come first serve, Round robin scheduling and is generalized priority algorithm. The experimental result shows that general prioritized algorithm is more competent than FCFS and Round Robin algorithm. [2]

Virtualization is the creation of a virtual version of something such as an operating system, a server, a storage device or network resources. Scheduling the basic processing units on a computing environment has always been an important issue. In cloud computing, a user may require a set of virtual machine co-operating with each other to accomplish one task. [3]

2. Related Work

This section lists out the various research works related to the problem that we have considered and the features and the drawbacks of such approaches. As already mentioned, scheduling virtual machines require computational resources and storage resources to a great extent.

[4] Proposed an enriched cost based scheduling algorithm for making competent mapping of tasks to available resources in cloud. This scheduling algorithm splits all user tasks depending on priority of each task into three different lists. This scheduling algorithm measures both resource cost and computation enactment, it also increases the ratio of communication/computation. [5] Proposed a new scheduling algorithm based on priority and admittance control scheme. In this algorithm priority is allotted to each admitted queue. Admission of each queue is decided by computing bearable delay and service cost. Advantage of this algorithm is that this policy with the proposed cloud architecture has attained improvement of 69% service rate with guaranteed QoS. As this policy provides the highest preference for highly paid user service-requests, overall servicing cost for the cloud also rises.

[6] Suggested the HBB-LB algorithm to achieve well stable load across virtual machines for maximizing the throughput. It provided a load balancing technique for cloud computing environments based on activities of honey bee strategy. Honey bee behavior motivated load balancing improves the overall throughput of processing and priority based balancing focuses on decreasing the waiting time for

the task on a queue of VM. This algorithm can be protracted further by considering the QoS factors in it. Yuanjun Lailia et al., [7] proposed the amalgamation of Service Composition Optimal Selection SCOS and Optimal Allocation of Computing Resources OACR is known as dual scheduling. For addressing large-scale Cloud Services and Computing Resources DS-CSCR problem, a new Ranking Chaos Optimization RCO is recommended. In RCO algorithm, individual chaos operator was considered, and then a new adaptive ranking selection was announced for control the state of population in iteration. Moreover, dynamic heuristics were also defined and presented to guide the chaos optimization. [8] Proposed the cost effective task-scheduling algorithm using two heuristic strategies .The first strategy dynamically maps tasks to the most cost-effective VMs based on the concept of Pareto dominance. The second strategy, a supplement to the first strategy, decreases the monetary costs of non-critical tasks. This algorithm is assessed with extensive simulations on both randomly produced large DAGs and real-world applications. The further enhancements can be made using new optimization techniques and incorporating forfeits for violating consumer-provider contracts. [9] Developed a Xen-based prototype called Rainbow with and without dynamic allocation. They proposed local and global resource allocation algorithms to improve the resource utilization and CPU utilization. [10] They proposed a novel dynamic hybrid Meta heuristic algorithm to maximize the profit along with higher overall performance and lower energy cost. They have validated the proposed approach with trace-driven simulations. In [11], they have proposed a heuristic approach based improved ant colony algorithm ACA to solve the VMP Problem which produces 99.5% CPU and 44% Server utilization. But, it requires high communication cost for node communication. [12] The main advantage of this algorithm is that it utilizes all the resources in a balanced order. An equal number of VMs are allocated to all the nodes which ensure fairness. [13] Brings out a pricing gets reduced by the proposed greedy heuristics increases the revenue around 6% for the cloud provider. [14] The proposed PSO based VM Scheduling algorithm reduces the total resource wastage and the number of servers used. But, an additional computational overhead is incurred. [15] The proposed algorithm reduces the job completion time by over 65% and resource utilization by 20%. But, the scheduler does not consider the resource fragmentation problem while resource packing.

3. Proposed optimal vm placement algorithm

Virtual machine scheduling is an important process in cloud computing environment with a large number of network resources, computing resources and storage resources being scheduled and executed. In such scenarios, one of the main challenges is to provide a good quality of service Quality of Service in spite of the dynamic behavior of the network. For virtual machine scheduling, Quality of Service can be derived from a number of factors and metrics. In the proposed framework to assurance the desired level of Quality of Service to the user is discussed.

The proposed frame work consists of the following steps as follows:

Step 1: Datacenter Creation: Data center is a basis entity in the cloud. Data center which consists of cloud resources such as computing resources, network resources and storage resources. It manages the cloud resources functionality. Resources such as virtual machines, host and cloudlets tasks are deployed in cloud data center. Such resources are allocated and functionality of each entity is controlled by the data center which is allocated on that particular data center. Each data center in the proposed work has a specific set of characteristics such as architecture, host list, cost per memory, cost per storage, cost per bandwidth and virtual machine allocation policy. Data center are created based on the characteristics and the parameters.

Step 2: VCreation: Virtual machines are the computing node in the cloud computing. Virtual machines are used to process the jobs submitted by the user according to the user's given requirement. In the proposed work, the virtual machines are scheduled according to the jobs priority. The jobs submitted are prioritized based on the jobs size and allocated to the virtual machine which has the minimal completion time. Each virtual machine has a specified set of characteristics such as virtual machine identity, ram capacity, bandwidth, size and no of processing elements. Virtual machines are created and schedule based on the above mentioned parameters.

Step 3: Queuing Model: Cloudlets are created with the specified characteristics such as cloudlet id, processing element, file size of the cloudlet, output size, cloudlet length and the utilization model for the ram, bandwidth and storage. Priority is assigned for the cloudlets based on the file size of the cloudlets. These cloudlets are submitted to the cloud broker. Broker is a cloud entity which is used to allocate the host and the virtual machine based on the scheduling policy. Virtual machine scheduling policies are defined in the scheduling. Broker Computation time and data transfer time for each cloudlet are estimated and the completion time for the virtual machine is calculated. Then, the waiting time for the service and the waiting time in queue are calculated.

To schedule the tasks into the suitable virtual machine, I have added the tasks into the priority based probability queuing model. In the proposed queuing model, I have used the M/G/1 queuing model to find the waiting time and waiting time for the service the task actually taken. The tasks are added into the queue based on the priority of the tasks. The formulas used in this implementation are listed as follows:

Number of tasks in the $=L_q=(\lambda^2 \sigma^2+ \rho^2)/(2(1-\rho))$

Number of tasks waiting in the queue$=W_q=L_q/\lambda$

Number of tasks waiting in the system$=W_s=W_q+1/\mu$

Number of tasks in the system$=\lambda W_s$

Where,

?? - Mean rate of arrival

?? – Variance of service time

?? – Utilization of the data center; also the probability that the data center is busy or the probability that someone is being served

µ - Mean service rate

Lq – Mean number of customers in the queue

Wq - Mean wait in the queue

Ws – Mean wait for service

These waiting times is then added to the cloudlets completion time to schedule the cloudlets into the appropriate virtual machines. These cloudlets are submitted to the cloud broker. Broker is a cloud entity which is used to allocate the host and the virtual machine based on the scheduling policy. Virtual machine scheduling policies are defined in the scheduling. Broker Computation time and data transfer time for each cloudlets are estimated and the completion time for the virtual machine are calculated .Cloudlets are scheduled and allocated on the virtual machine which has the minimal jobs completion time for the submitted cloudlets.

Step 4: BCreation: The broker is created which is used for allocation of cloudlets to the virtual machines and scheduling virtual machines to the corresponding host in a data center. Broker manages the virtual machines and submission of cloudlets to the virtual machines. It also has a set of parameters such as broker id and in which data center the broker is working. Broker calculates the computation time and the processing cost for each cloudlet. Then the data transfer rate for each cloudlet is calculated and the computation time i.e., the actual execution time of task and the finishing time are estimated. After that the cloudlets are submitted to the virtual machines. Then the virtual machines which have the minimal completion time for the submitted task are calculated and scheduled accordingly. After that, the virtual machines are submitted to the broker.

Step 5: VM Allocation: The virtual machines are scheduled to the cloudlets which have the minimal completion time. First the computation time i.e., the time it takes to complete the given job for each virtual machine is calculated. Then the completion time for the jobs is calculated by summing up both the computation time and data transfer time. Computation time for the job is defined as the time needed to process the give task and data transfer time is defined as the time needed to transfer task file. Each task is associated with priority based on the file size of the task submitted by the user. The virtual machine which has the minimal job completion time is allocated to that particular task.

Thus we have obtained an optimal VM Placement Algorithm for the above mentioned cloud scenario.

Input: VmList, Required Data, CnList and JobList
Output: allocation of VMs to Jobs

```
For Vm in VmList do
MinComTime ← MAX
AllocatedHost ← NULL
For Cn in CnList do
If Load ≤ CnCap then
For job in JobList do
DTT job, Data, StorageList, Cn
CT job, Cn // method to calculate the computation time
CTime = DTT + CT
End for
ComTime ← CTime
If ComTime<MinComTime then
AllocatedHost ←CnList
MinComTime ← ComTime
End if
End if
End for
If AllocatedHost! = NULL then
Allocation.add Vm, AllocatedHost
End if
End for
DataTransferTime DTT
Input: Job Size, S, Cn
Output: returns data transferring time
For S in StorageList do
DTT = Job Size / DTRate, Cn
End for
Return DTT
ComputingTime CT
Input: job, Cn
Output: returns computing time CT
CT= ActualExecStartTime + FinsishTime;
CT =CT + WaitingTime;
Return CT
```

Code Snippet 1: Optimal VM Placement Algorithm

The main advantage of the proposed Optimal VM Placement algorithm is minimization of completion time of task submitted to virtual machines as much as the application can execute and minimization of processing cost of virtual machines to obtain a high performance. The proposed Optimal VM Placement algorithm focuses on the performance improvisation of the applications submitted on the cloud. The results obtained by the proposed methodology is not only evaluated with completion time and processing cost but it also includes the consideration of computation time and processing time required for the submitted applications.

The above mentioned optimistic VM placement algorithm takes the inputs such as computing node list, job (cloudlet) list, and required data resources information. For each vm in the VM list, it calculates the processing capability and then the processing time i.e., the number of instructions it can process per second. It checks whether the load on the computing node is less than maximum capability. Then the data transfer rate for each cloudlet is calculated and the computation time i.e., the actual execution time of task and the finishing time are estimated. Then the completion time for the jobs is calculated by summing up both the computation time and data transfer time.

The architecture of the proposed Optimal VM Scheduling framework in Figure 1.The optimistic vm placement in cloud data center architecture consist of set of users and the components such as broker, storage nodes, computing nodes and virtual machines. First, the users submit their jobs into the cloud along with constrains such as number of jobs, number of virtual machines needed, amount of storage space required. Then the submitted user information is sent to the cloud broker. Broker is created which is used for allocation of cloudlets to the virtual machines and scheduling virtual machines to the corresponding host in a data center. Broker manages the virtual machines and submission of cloudlets to the virtual machines.

It also has a set of parameters such as broker id and in which data center the broker is working. Broker calculates the computation time and the processing cost for each cloudlet. Then the data transfer rate for each cloudlet is calculated and the computation time i.e., the actual execution time of task and the finishing time are estimated. After that the cloudlets are submitted to the virtual machines. Then the virtual machines which have the minimal completion time for the submitted task are calculated and scheduled accordingly. After that, the virtual machines are submitted to the broker.

The data resources, computing and network resources information are directed to the cloud broker. Then the broker deals with the jobs (Cloudlets) submitted by the user. Then, insertion of tasks i.e. cloudlets into the priority queue and calculating the data transfer time and completion time and assignment of priority to the cloudlets. Cloudlets are created with the specified characteristics such as cloudlet id, processing element, file size of the cloudlet, output size, cloudlet length and the utilization model for the ram, bandwidth and storage. Priority is assigned for the cloudlets based on the file size of the cloudlets. These cloudlets are submitted to the cloud broker. Broker is a cloud entity which is used to allocate the host and the virtual machine based on the scheduling policy. Here, the priority queue is associated with the queuing model i.e. single channel multiple server model .So that, the mean arrival time and the service time for each cloudlets are calculated.

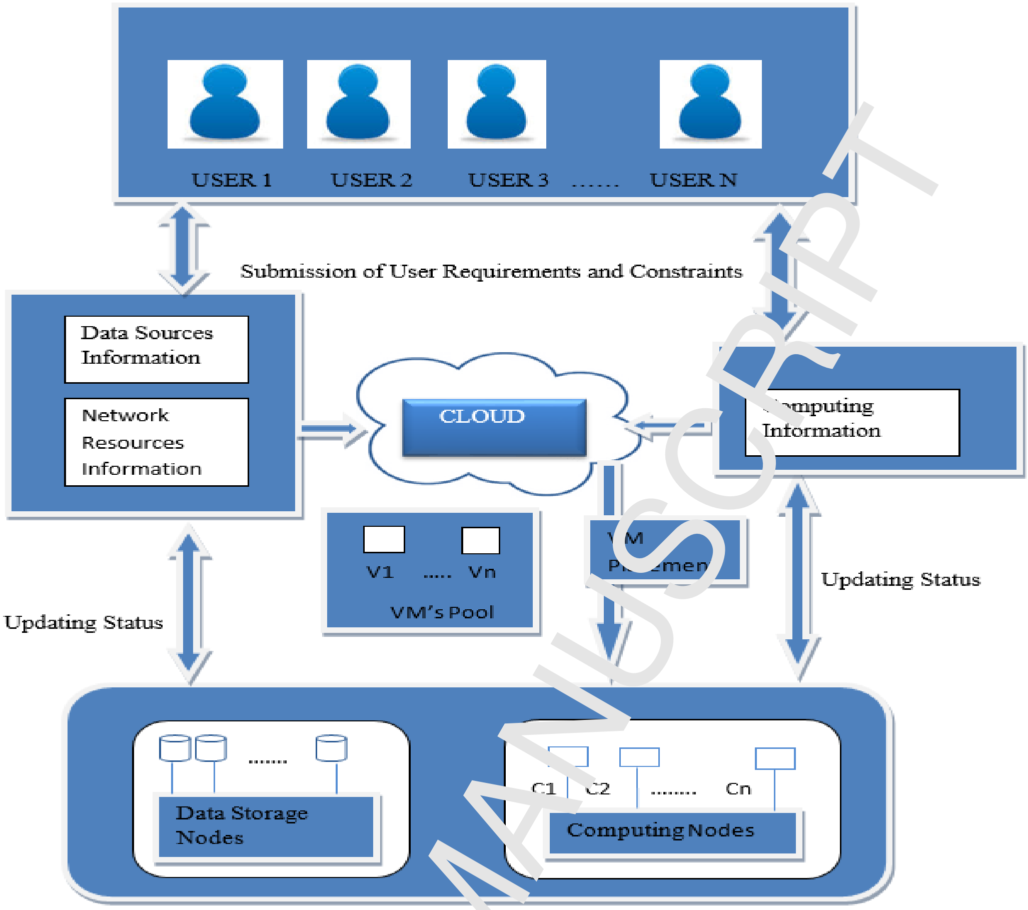


Figure 1: Optimistic vm placement in cloud data center

And then, waiting time in queue and waiting time for service is calculated using the formulas. The waiting time calculated is added to the cloudlets computation time for scheduling the virtual machine with minimal completion time.

Virtual machines are scheduled to the cloudlets which has the minimal completion time. First the computation time i.e., the time it takes to complete the given job for each virtual machine is calculated. Then the completion time for the jobs is calculated by summing up both the computation time and data transfer time. Each task is associated with priority based on the file size of the task submitted by the user. The virtual machine which has the minimal job completion time is allocated to that particular task. Virtual machine scheduling policies are defined in the scheduling policies of vm allocation. Broker computes the computation time and data transfer time for each cloudlets are estimated and the completion time for the virtual machine are calculated .Cloudlets are scheduled and allocated on the virtual machine which has the minimal jobs completion time for the submitted cloudlets. Then, Completion time and the processing cost of each cloudlet are calculated. Finally the completed jobs are delivered to the users and the status is updated to the cloud users.

4. Experiments and Results

We have implemented the proposed Optimal VM Placement algorithm and we analyses the performance pertaining to various parameters and traditional VM scheduling algorithms. We have implemented the system for the entire implementation is done on the CloudSim Simulator. The experiment is carried with different amount of cloudlets, virtual machines, data centers and hosts with the different scenarios which was mentioned below in the table.

Table 1: Experimental setup details.

Datacenters	Hosts /Datacenter	Virtual Machines/Host	Cloudlets	Workload Scenario
10	10	10	1000	Static
50	35	50	5000	Static
70	55	70	7000	Static
100	70	100	10000	Static
10	10	10	1000	Dynamic
50	35	50	5000	Dynamic
70	55	70	7000	Dynamic
100	75	100	10000	Dynamic

Virtual machines are scheduled to the cloudlets which has the minimal completion time. First the computation time i.e., the time it takes to complete the given job for each virtual machine is calculated. Then, the waiting time calculated is added to the cloudlets computation time for scheduling the virtual machine with minimal completion time Broker is created which is used for allocation of cloudlets to the virtual machines and scheduling virtual machines to the corresponding host in a data center.

Broker Computation time and data transfer time for each cloudlets are estimated and the completion time for the virtual machine are calculated .Cloudlets are scheduled and allocated on the virtual machine which has the minimal jobs completion time for the submitted cloudlets. Each task is associated with priority based on the file size of the task submitted by the user. The virtual machine which has the minimal job completion time is allocated to that particular task. The result of completion time obtained by the proposed optimal VM placement algorithm was shown in the Chart 1.

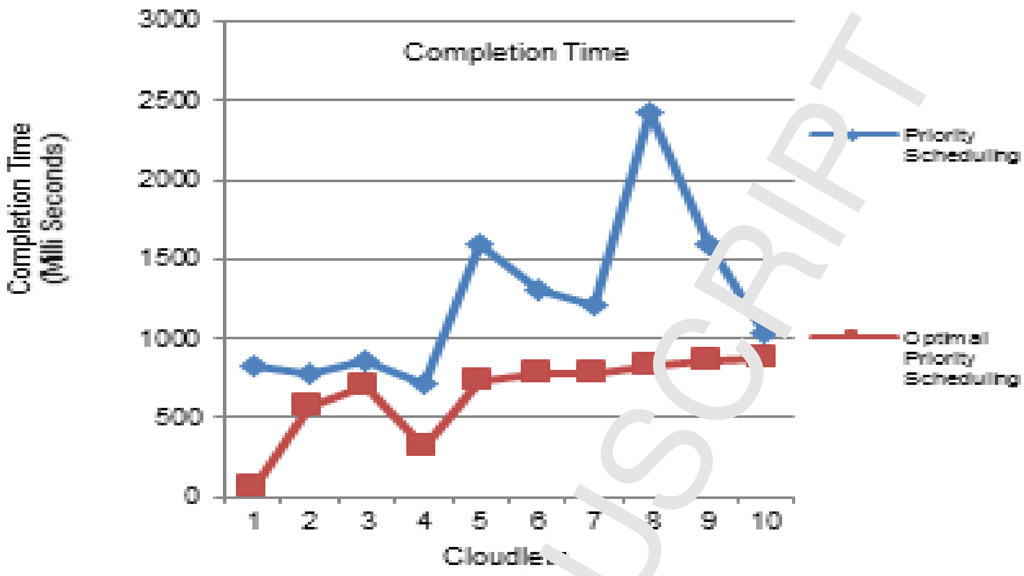


Chart 1: Completion time of cloudlets
(Optimal Priority Scheduling Vs Priority Scheduling)

It's clear that the completion time by the proposed optimal VM placement algorithm has a reduced completion time by 91.42% compared with existing priority based scheduling.

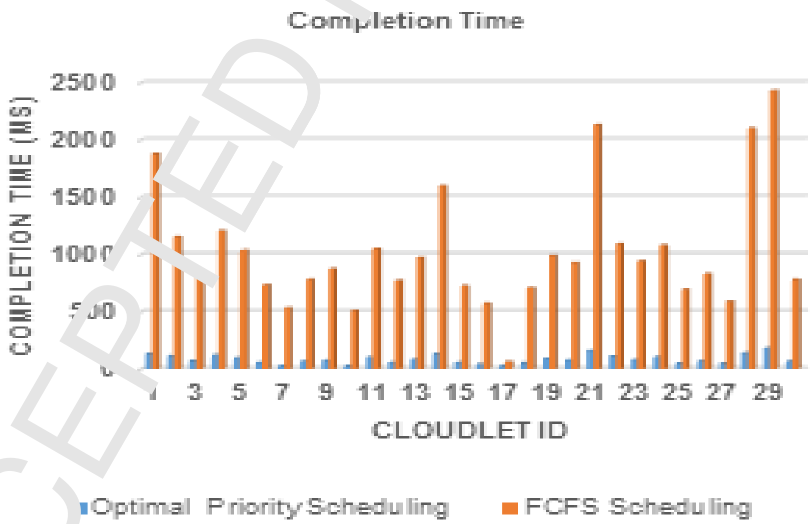


Chart 2: Completion time of cloudlets
(Optimal Priority Scheduling Vs FCFS Scheduling)

The result of completion time obtained by the proposed optimal VM placement algorithm was compared with FCFS scheduling algorithm which was shown in the Chart 2. It's clear that the completion time by the proposed optimal VM placement algorithm has a reduced completion time by 92.36% compared with existing FCFS scheduling.

And, also processing cost for each cloudlet is calculated using the execution time of each cloudlet and the cost it takes to process per second. Broker Computation time and data transfer time for each cloudlets are estimated and the completion time for the virtual machine are calculated .Cloudlets are scheduled and allocated on the virtual machine which has the minimal jobs completion time for the submitted cloudlets.

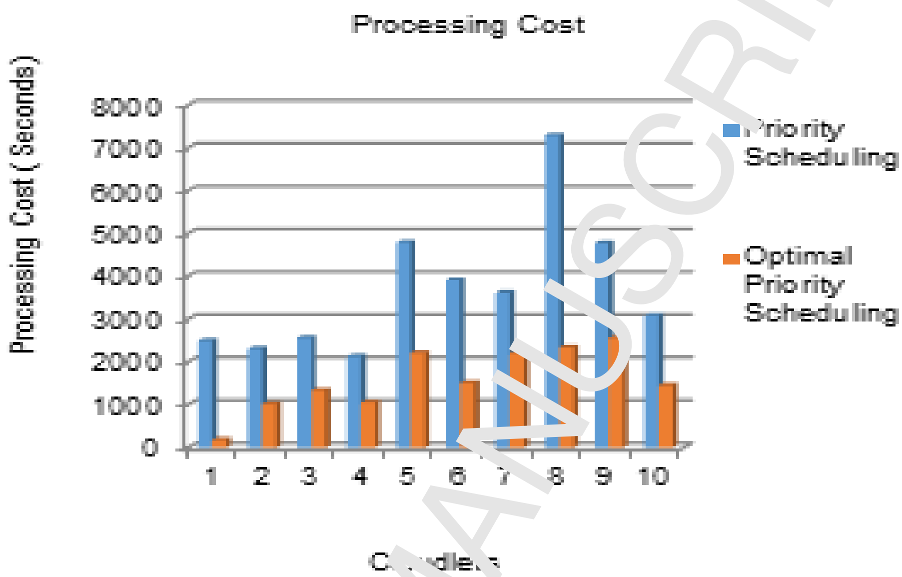


Chart 3): Processing cost of cloudlets
(Optimal Priority Scheduling Vs Priority Scheduling)

The result processing cost obtained by the proposed optimal VM placement algorithm was shown in the Chart 3. It's clear that the completion time by the proposed optimal VM placement algorithm has a reduced processing cost by 91.8% compared with existing priority based scheduling.

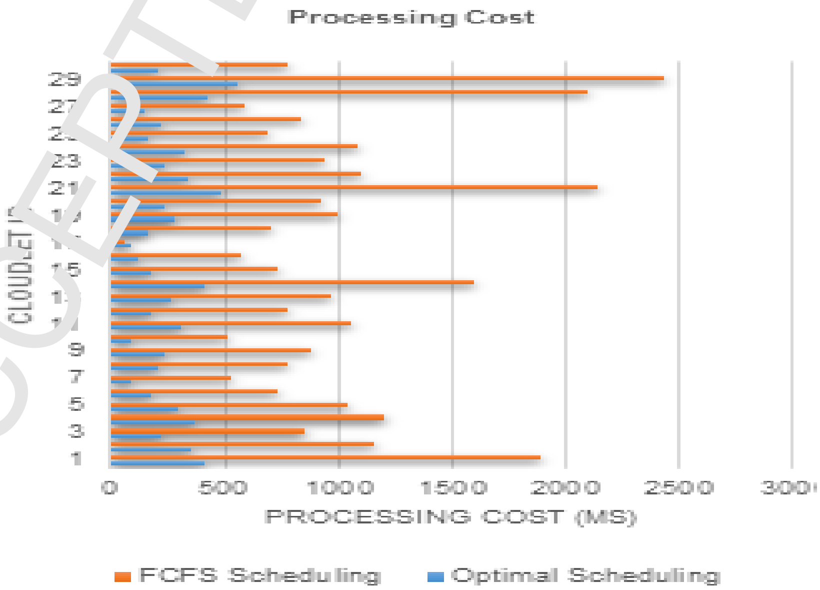


Chart 4: Processing cost of cloudlets
(Optimal Priority Scheduling Vs FCFS Scheduling)

The result processing cost obtained by the proposed optimal VM placement algorithm was shown in the Chart 4. It's clear that the completion time by the proposed optimal VM placement algorithm has a reduced processing cost by 84.62% compared with existing FCFS based scheduling.

The result obtained by the proposed optimal VM placement algorithm was shown in the table below.

Table 2: Comparison of obtained results.

Scheduling Algorithm compared with Proposed Algorithm	CompletionTime Obtained	ProcessingCost Obtained
FCFS Scheduling	92.36%	84.62%
Priority Scheduling	91.42%	91.8%

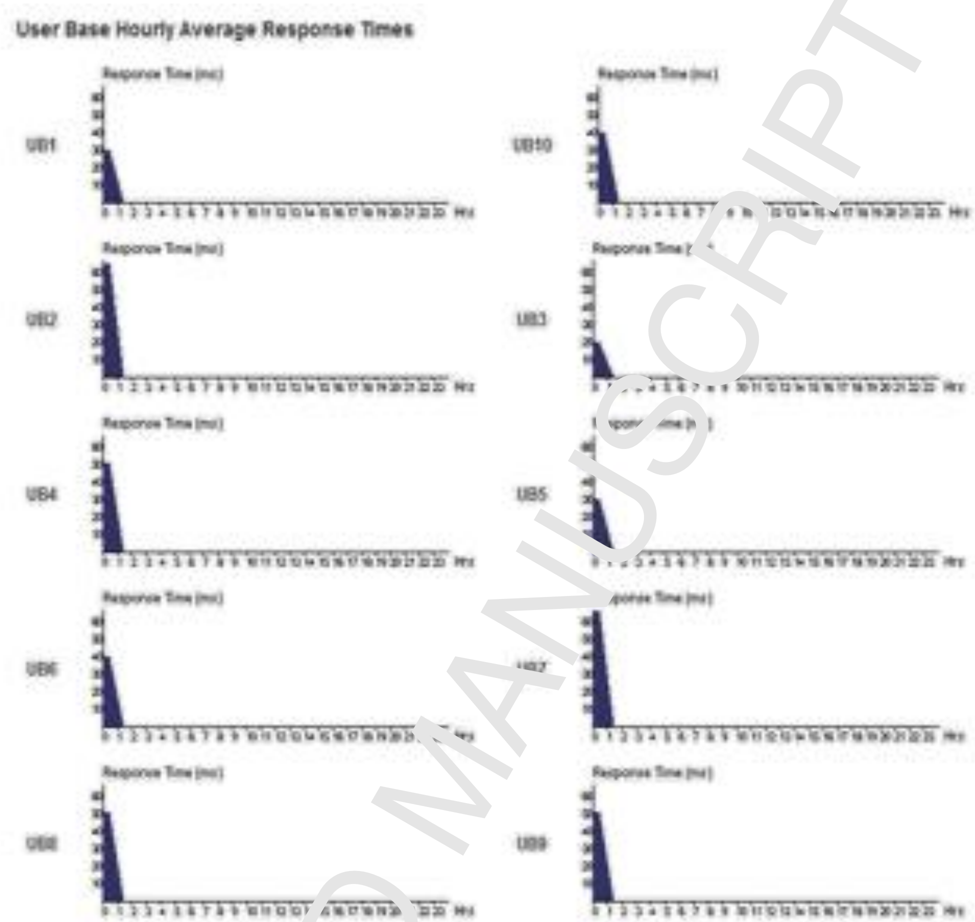
We choose the FCFS and Priority scheduling methods to compare our proposed Optimal VM Placement algorithm because we are also considering the priority of jobs (tasks) as followed in the traditional priority scheduling. And, we also consider the job which arrived first along with its computing and storage requirement. Therefore, we have to compare our results with these traditional methods, to analyze how much our proposed algorithm is working better than these traditional methods in both static and dynamic workloads.

The results obtained by the proposed methodology is not only evaluated with completion time and processing cost but it also includes the consideration of computation time and processing time required for the submitted applications. The completion time of job is associated with computation time needed for the applications requirement and the processing cost of task is also associated with processing time of the virtual machine which executes the applications submitted.

The results obtained by the proposed methodology shows that the proposed optimal VM placement algorithm has a reduced processing cost by 91.8% and completion time by 91.42% compared with existing priority based scheduling. Also, the proposed optimal VM placement algorithm has a reduced completion time by 92.36% and the processing cost by 84.62% compared with existing FCFS based scheduling.

The main advantage of the proposed Optimal VM Placement algorithm is minimization of completion time of task submitted to virtual machines as much as the application can execute and minimization of processing cost of virtual machines to obtain a high performance. The proposed Optimal VM Placement algorithm focused on the performance improvisation of the applications submitted on the cloud.

The result obtained by the proposed optimal VM Placement Algorithm was simulated in both the Cloud Reports and Cloud Analyst and shown in Chart 5.



Data Center Request Servicing Times

Data Center	Avg (ms)	Min (ms)	Max (ms)
DC1	1.36	0.1	2
DC2	1.447	0.2	3.4
DC3	0.724	0.067	1.233

Chart 5: Average response time for the user simulated in cloud analyst.

5. Conclusion and Future work

We have proposed an Optimal VM Placement algorithm for virtual machine scheduling in cloud computing environment. Virtual machine scheduling is very important in the field of cloud computing. Cloud computing gives beneficial services to share large scale of information, storage resources, computing resources, while applications need to retrieve its data from distributed storages, the bandwidth between Computer Networks and Social Networks could influence the overall applications performance specially, when network status is unstable. So it's important to consider data location during VMs placement to avoid unnecessary migration to gain high performance for applications. To address this problem, we proposed VMs placement algorithm that considers both computation Time and data transferring time in such a way that we can reduce the overall job's completion time. The proposed Optimal VM Placement Algorithm was implemented with static and dynamic workloads. The proposed algorithm also simulated and analyzed with Cloud Reports and Cloud Analyst.

The main advantage of the proposed Optimal VM Placement algorithm is minimization of completion time of task submitted to virtual machines as much as the application can execute and minimization of processing cost of virtual machines to obtain a high performance. The proposed Optimal VM Placement algorithm focused on the performance improvisation of the applications submitted on the cloud.

The results obtained by the proposed methodology shows that the proposed optimal VM placement algorithm has a reduced processing cost by 91.8% and completion time by 91.42% compared with existing priority based scheduling. Also, the proposed optimal VM placement algorithm has a reduced completion time by 92.36% and the processing cost by 84.62% compared with existing FCFS based scheduling. The main contribution of this thesis to gain high performance for the applications executed on the cloud by minimizing the completion time, minimizing the production cost and maximizing the throughput of cloud links.

In future, this proposed procedure can be extended to Schedule the virtual machines in a secured manner using profile based analysis. And switching can also be taken up as future work. Moreover, analyze the user profile thereby understanding the user's requirement clearly and then schedule the virtual machines will be the best solution in cloud computing environment and my future work will also focuses on the user submitted files to be stored in a secured manner on the virtual machines along with my proposed procedure. The system was designed for and implemented on a CloudSim Simulator. In future, I recommended that this proposed procedure can also be extended to public and hybrid cloud platform scenarios.

6. List of abbreviations

The following are the list of abbreviations used in this paper:

1. QOS — Quality Of Service
2. VM — Virtual Machine
3. VMM — Virtual Machine Monitor
4. VMP — Virtual Machine Placement
5. HBB-LB — Honey Bee Behavior Based Balancing
6. FCFS — First Come First Serve
7. SNs — Storage Nodes
8. CNs — Computing Nodes
9. CnList — Computing Node List
10. DTT — Data Transfer Time
11. CnCap — Computing Node Capability

12. CT — Computation Time
13. ComTime — Completion Time
14. DTRate — Date Transfer Rate.

Acknowledgments

I am highly indebted to our respectable Dean, Dr. A. RAJADURAI, MIT Campus, and Anna University for providing us with sufficient facilities. I am thankful to my Head of the Department Dr. P. ANANDHAKUMAR, Department of Computer Technology, MIT, and Anna University for his continuous encouragement throughout the course and for providing the necessary facilities to carry this work.

I would like to express my sincere thanks and deep sense of gratitude to my supervisor Dr. P. ANANDHAKUMAR, Head of the Department and my joint Supervisor Ms. D. PREETHA EVANGELINE, Assistant Professor, Department of Computer Technology, VIT University for her valuable guidance, suggestions and constant encouragement which paved way for the successful completion of this project work.

References

[1] Amit Agarwal, SaloniJain, Efficient Optimal Algorithm of Task Scheduling in Cloud Computing Environment, International Journal of Computer Trends and Technology (2014).

[2] C.T.Lin, Comparative Based Analysis of Scheduling Algorithms for Resource Management in Cloud Computing Environment, International Journal of Computer Science and Engineering (2013).

[3] Chunge Zhu, Daochao Huang, Hong Zhang, Peng Du, Xinran Liu, Multi Resource Packing for Job Scheduling in Virtual Machine Based Cloud Environment (2015).

[4] D. G. S. Sudhasivam, M. Selvarani, Improved Cost-Based Algorithm for Task Scheduling in Cloud Computing, IEEE International Conference on Computational Intelligence and Computing Research (2010).

[5] D. Karan, Prajapati Miren Karamta, Comparison of Virtual Machine Scheduling Algorithms in Cloud Computing, International Journal of Computer Applications (2013).

[6] D. M Dakshayini, D. H. S Guruprasad, An Optimal Model for Priority based Service Scheduling Policy for Cloud Computing Environment, International Journal of Computer Applications (2011).

[7] Daniel Grosu, Lena Mashayekhy, Mahyar Movahed Nejad, Truthful Greedy Mechanisms for Dynamic Virtual Machine Provisioning and Allocation in Clouds, IEEE Transactions on Parallel and Distributed Systems (2015).

[8] Dinesh Kumar, Zahid Raza, A PSO based VM Resource Scheduling Model for Cloud Computing, IEEE International Conference on Computational Intelligence & Communication Technology (2015).

[9] Fei Tao, Lida Xu, Lin Zhang, Yuanjun LaiLi, FC-PACO-RM: A Parallel Method for Service Composition Optimal-Selection in Cloud Manufacturing System, IEEE Transactions on Industrial Informatics (2013).

[10] Hadi Salimi, Advantages, Challenges and Optimizations of Virtual Machine Scheduling in Cloud Computing Environments, International Journal of Computer Theory and Engineering (2012).

[11] Hong, Liang, Yufei Ge, GACA-VMP: Virtual Machine Placement Scheduling in Cloud Computing Based on Genetic Ant Colony Algorithm Approach, IEEE 12th International Conference on Ubiquitous Intelligence and Computing (2015).

[12] Jing Bi, Haitao Yuan, MengChu Zhou, Yushun Fan, Wei Tan, Application-Aware Dynamic Fine-Grained Resource Provisioning in a Virtualized Cloud Data Center, IEEE Transactions On Automation Science And Engineering (2013).

[13] L.D. Dhinesh Babu, P.Venkata Krishna , Honey bee behavior inspired load balancing of tasks in cloud computing environments, Science Direct, Applied Soft Computing (2013).

[14] Ying Song, Yuzhong Sun, Weisong Shi, A Two-Tiered On-Demand Resource Allocation Mechanism for VM-Based Data Centers, IEEE Transactions on Services Computing (2013).

Author biography

Anitha Ponraj was born in Tamil Nadu, India, in 1993 received the B.E, M.E. degree in computer science engineering. She received gold medal and first rank certificate for her Post Graduate degree in the year of 2018 by Anna University Chennai, India and doing the Ph.D. degree in the areas of Image processing with deep learning in Sathyabama University of Chennai, Tamil Nadu, and India.

Highlights:

This research was carried for 2 years (From 2015 to 2017) as a Post Graduate academic project in Anna University Chennai, Tamil Nadu, and India.

- Data location during Virtual machines placement was considered to avoid unnecessary migration to gain high performance for applications. The main contribution of this thesis to gain high performance for the applications executed on the cloud by minimizing the completion time, minimizing the production cost and maximizing the throughput of cloud links.
- The main advantage of the proposed Optimal VM Placement algorithm is minimization of completion time of task submitted to virtual machines as much as the application can execute and minimization of processing cost of virtual machines to obtain a high performance. The proposed Optimal VM Placement algorithm focused on the performance improvisation of the applications submitted on the cloud.
- The proposed optimal VM placement algorithm has a reduced processing cost by 91.8% and completion time by 91.42% compared with existing priority based scheduling in both static and dynamic workloads.
- Also, the proposed optimal VM placement algorithm has a reduced completion time by 92.36% and the processing cost by 84.62% compared with existing FCFS based scheduling in both static and dynamic workloads.
- The proposed optimal virtual machine placement algorithm also simulated and analyzed with Cloud Reports and Cloud Analyst.