NITTE EDUCATION TRUST | N.M.A.M. INSTITUTE OF TECHNOLOGY
(An Autonomous Institution affiliated to Visvesvaraya Technological University, Belagavi)
Nitte – 574 110, Karnataka, India
(ISO 9001:2015 Certified), Accredited with 'A' Grade by NAAC
☎: 08258 - 281039 – 281263, Fax: 08258 – 281265
Department of Computer Science and Engineering
B.E. CSE Program Accredited by NBA, New Delhi from 1-7-2018 to 30-6-2021

Report on Mini Project

Modified Shortest Job First Scheduling in CloudSim

Course Code: 19CSE33

Course Name: Cloud Computing

Semester: VI SEM Section: D

Submitted To:

Mr. Shashank Shetty
Assistant Professor [GD-II]
Department of Computer Science
and Engineering

Submitted By:
Sumukha Jagadeesh Bhat - 4NM19CS198
Swathi - 4NM19CS201

Date of submission:
26.05.2022

Signature of Course Instructor

ABSTRACT

This project aims to implement the improved shortest job first scheduling algorithm in cloud computing. The already existing shortest job first (SJF) algorithm doesn't utilize resources efficiently and has higher makespan and lower response time. In a scheduling algorithm, the most important parameters to be considered are makespan and response time. Therefore, we have implemented the improved SJF algorithm which maximizes the resource usage and efficiency, as well as minimize the completion time of the last task (Makespan) and the average response time.

TABLE OF CONTENTS

INTRODUCTION

The default task scheduling algorithm available in Cloudsim simulator is First Come First Serve (FCFS) scheduling algorithm. In this project, we have implemented the shortest job first (SJF) and a modified shortest job first (MSJF) algorithm and have compared the Makespan and response time among the mentioned three task scheduling algorithms.

The criterion of the Shortest Job First (SJF) scheduling algorithm is that it processes the smallest task first. In our algorithm, if the task takes more time to execute, the broker will send the task to virtual machine which has higher configuration to execute this task, and if the task has the lower execution time, the broker will send it to the virtual machine having lower configuration in order to be executed.

PROBLEM STATEMENT

The problem of task scheduling is selecting the best suitable VMs or computing resources with the consideration of maximizing resource utilization. Our aim is to reduce the total completion time (makespan) and the response time with maximizing the resources utilization. We attempt to manage the loads between the virtual machines and use the best algorithm to execute the tasks with taking into consideration maximizing the resource utilization and reducing the completion time.

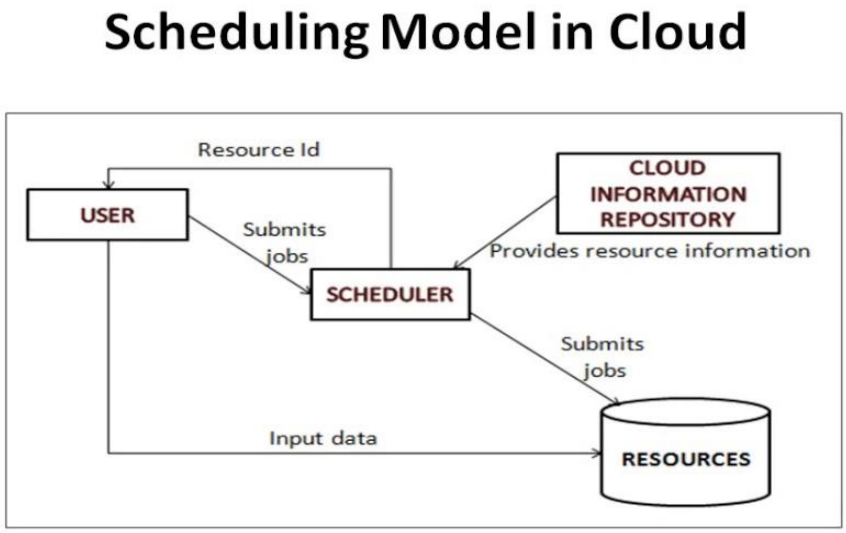


Figure 1: Scheduling in Cloud [1]

OBJECTIVES

- To get a basic understanding of how resources are scheduled in CloudSim simulator
- To improve the existing SJF scheduling algorithm
- To maximize the resource utilization and efficiency
- To minimize makespan and response time

HARDWARE / SOFTWARE REQUIREMENTS

Hardware Requirements:

RAM	4 GB or more
Processor	Intel or AMD Processor
Hard Disk	100GB or more

Software Requirements:

Operating System	Windows / Linux / Mac OS
Editor	Eclipse IDE
Simulator	CloudSim version 3.0.3

LITERATURE SURVEY

Traditional SJF algorithm is less efficient because its completion time is high. So, modifications are done to existing SJF in which longer tasks are given to high configuration VMs and shorter tasks to low configuration ones [1]. Different algorithms exist for task scheduling and some of the targets are load balancing, quality of service, running time and throughput as mentioned in [2]. Various task scheduling algorithms to balance the load among different datacenters exist, and the waiting and turnaround times are tabulated for each task by varying the parameters in [3].

The scheduling algorithms can be implemented by considering space and time-shared allocation policies. After the findings given in [4], we can conclude that time shared policy is better than space shared in terms of time of completion. Largest task first (LTF) has the least performance compared to others. In [5], the authors have implemented Matching and Multi-round Allocation (MMA), Modified Cuckoo Search (MSC), Hybrid Chaotic Particle Search (HCPS), Modified Artificial Bee Colony (MABC), max-min and min-min algorithms, provided detailed explanation about each and compared the time of completion in a detailed way.

To perform task scheduling and resource allocation, paper [6] proposes a heuristic approach where it combines the modified analytic hierarchy process (MAHP), bandwidth aware divisible scheduling (BATS) + BAR optimization, longest expected processing time preemption (LEPT), and divide-and-conquer methods. Each job is processed using an MAHP procedure before being allocated to cloud resources in this technique. The resources are allocated using a combined BATS + BAR optimization method, which takes into account the cloud resources' bandwidth and load as restrictions.

In [7], the authors have presented a multi-objective approach for task scheduling optimization that minimizes task execution time, task transferring time, and task execution cost. They devise a multi-objective algorithm based on multi-objective PSO (MOPSO) method to provide an optimal solution for the suggested model. They provide a multi-objective Jswarm (MO-Jswarm) package which determines the ideal work allocation among VMs.

Earlier there was a method called non live migration to transfer the load among compute nodes. But along with that, there were many challenges such as the user would get the server downtime for infinite delay. To overcome this, a technique known as Hot Migration or Live migration is used. It allows the administrators of the system to shuffle the available load on any operating system to other Guest Operating system without intervention [8].

Soft computing is useful for resource optimization and scheduling in the cloud with greater performance. The study and implementation of resource allocation and scheduling methodologies in the cloud environment to improve the cloud environment's overall performance is done in [9].

The goal of the study in [10] is to give a complete, structured literature review on several elements of cloud computing resource allocation, such as strategic, target resources, optimization, scheduling, and power. It includes a thematic taxonomy of resource allocation dimensions, with articles discussing and examining each category. A basic analysis of the CloudSim simulator's functionality, as well as a tabulated perspective of several scheduling algorithms deployed in the cloud, together with their parameters and results are documented in [11].

A cost-based resource allocation technique that will reduce execution time, total completion time, and enhance profit by prioritizing resources based on the cost paid by the user to the service provider for the resource used. This method moves the virtual machine from one host or physical machine to another to avoid the host becoming overcrowded or underloaded.[12]

A hybrid approach for virtual machine allocation that is based on the genetic algorithm (GA) and the random forest (RF) which are basically supervised machine learning approaches is implemented in [13]. The project's goal is to reduce power consumption while improving load balancing among available resources and increasing resource utilization.

There are numerous load balancing algorithms like SJF algorithm, First Come First Serve (FCFS), equal task distribution using cloudlets binding, and uneven task distribution using cloudlets binding on the virtual machines in the study made by the authors in [14]. They also talked about static load balancing and dynamic load balancing algorithms.

Overloading and under-loading are two unwanted aspects of load unbalancing. Load balancing strategies provide a remedy for both. A comprehensive overview of load balancing techniques along with the benefits and limits of existing systems are discussed, as well as key problems that must be overcome in order to build efficient load balancing algorithms in the future are elaborated in [15].

SUMMARY OF LITERATURE SURVEY

Author details with year of publication	Paper Title	Methodology	Advantage	Disadvantage
Mokhtar A. Alworafi, Atyaf Dhari, 2016	An Improved SJF Scheduling Algorithm in Cloud Computing Environment	Modifications are done to existing SJF where longer tasks are given to high configuration VMs and shorter tasks to low configuration ones.	Maximized resource utilization and efficiency and minimize response time.	Enhancement of makespan is not done based on QoS factors such as deadline constraint.
Hong Sun, Shi-ping Chen, Chen Jin, Kai Guo, 2013	Research and Simulation of Task Scheduling Algorithm in Cloud Computing	Compared various task scheduling algorithms.	Gives overall view of basic scheduling algorithms.	Does not give greater details about how each algorithm is implemented.
Ram Pratap, Taskeen Zaidi, 2018	Comparative Study of Task Scheduling Algorithms through Cloudsim	Load balancing through task scheduling, comparing FCFS, SJF and RR.	Compares the turnaround times for each type of scheduling.	Does not focus on drawbacks of the mentioned algorithms.
Fahd Alhaidari, Taghreed Balharith, Eyman AL-Yahyan, 2019	Comparative Analysis for Task Scheduling Algorithms on Cloud Computing	Compared FCFS, STF, LTF and RR algorithms considering space and time-shared allocation policies.	Gives details regarding what parameters to be considered for each algorithm and its findings.	Does not consider QoS metrics.
Qing-Hua Zhu, Huan Tang, Jia-Jie Huang, Yan Hou, 2021	Task Scheduling for Multi-Cloud Computing Subject to Security and Reliability Constraints	Compared the results of MMA, MCS, HCPS, MABC, max-min, min-min algorithms.	Minimizes the variance of the time of completion. Can be applied to both batch tasks and workflows.	Does not consider task features like transmission of data and deadlines.
Mahendra Bhatu Gawali, Subhash K. Shinde, 2018	Task scheduling and resource allocation in cloud computing using a heuristic approach	Gives a heuristic approach which combines analytic hierarchy process, divide and conquer methods and scientific workflows to allocate resources.	Maximize the utilization of resources such as CPU, bandwidth and physical memory.	Does not improve response or turnaround time.

Fahimeh Ramezani, Jie Lu, and Farookh Hussain, 2013	Task Scheduling Optimization in Cloud Computing Applying Multi-Objective Particle Swarm Optimization	Implemented multi objective PSO algorithm and also extended the Jswarm package to MO-Jswarm.	Achieves highest QoS and the algorithm is able to determine the optimal arrangement of tasks among VMs	Does not consider task priorities and energy consumption is more as inactive processors are not chosen for executing tasks.
Gupta, Ambika. (2021).	A modelling & Simulation via CloudSim for Live Migration in Virtual Machines.	Perform live migration by optimizing the transfer of memory and CPU state.	Provides an efficient system in which there is no downtime in maintenance of server.	Memory overhead in transferring whole memory pages and VM states.
Haitham Salman, Raniah Ali, Kawther Thabit, 2018	Study and Implementation of Resource Allocation Algorithms in Cloud Computing	Focuses on implementation of allocation of resources and scheduling approaches in the cloud environment	Integrated with soft computing which gives high performance.	Gives approximate values for result.
Muhammad Faraz Manzoor, Adnan Abid, 2020	Resource Allocation Techniques in Cloud Computing: A Review and Future Directions	Gives an overview of resource allocation and its different techniques.	Limits the overhead and costs associated with resource allocation.	Required to consider penalty limitations due to system failures.
Ekta Rani, Harpreet Kaur, 2017	Study on fundamental usage of cloudsim simulator and algorithms of resource allocation in cloud computing	1.Initialization of virtual machines and cloudlets 2. Assign the cloudlets to VMs using Soft computing 3. Calculate the performance parameters 4.Analysis	Various scheduling algorithms based on distinguishable scheduling parameters like resource utilization, response time and results have been compared and analyzed.	Although minimizes response time, it lacks in utilization of resources, increases cost and waiting time in case of load imbalance on Virtual Machines
Manish Pandey, Sachin Kumar Verma, 2017	Cost based resource allocation strategy for the cloud computing environment	Gives algorithm for VM scheduling and VM selection during the migration	Increases the provider profit and minimizes the user bill. Gives the best services to the user those are using the cloud services for a longer period of time	More recently joined users will face the absence of best services since this favors the users who use the cloud services for a longer period of time

Madhusudan H S, Satish Kumar T, Punit Gupta, 2021	Hybrid Approach for Resource Allocation in Cloud Infrastructure Using Random Forest and Genetic Algorithm	The proposed virtual machine placement technique in this work is a hybrid model using a genetic algorithm and random forest technique	The algorithm used is one of the most flexible and easy-to-use algorithms	Large number of trees can make the algorithm too slow and ineffective for real-time predictions
Raushan Ranjan Kumar, Sarvjeet Kumar Jha, Deepak garg, 2018	Evaluation of Load Balancing Algorithm Using Cloudsim	Implementation of FCFS, SJF and Cloudlet binding Strategy discussed	Compared all results, found that the cloudlet binding with equal distribution have optimum results as compared to others	Not allocated the task on virtual machine in dynamic optimized way, hence the results aren't optimal
Shahbaz Afzal, G. Kavitha, 2019	Load balancing in cloud computing – A hierarchical taxonomical classification	Constructive Generic Framework (CGF) Methodology further reinforced by Systematic Literature Review (SLR) methodology	Outlines the comparative study of different load balancing algorithms and provides a hierarchy based on different QoS metrics	Some of the essential QoS metrics are not discussed in reviewed articles in full depth, like Migration time, migration cost, power consumption, service level violation, task rejection ratio and degree of balance

METHODOLOGY

- Set the parameters and note down the output for the default FCFS task scheduling algorithm
- Implement SJF algorithm and run the experiment for the same set of parameters and note down the output
- Implement the Modified SJF algorithm
    - Step 1: Create a heterogeneous cloud computing environment.
    - Step 2: Input all the required tasks and calculate the average of tasks length for all the tasks.
    - Step 3: Sort the tasks in an ascending order.
    - Step 4: For each task, if the task length is less than the average and number of tasks in VM1 less than number of tasks in VM2 then pass the task to VM1 (less computing power), else passed it to VM2 (high computing power).
    - Step 5: Calculate the average response time and makespan.

The following parameters are used in MSJF:

- Average task length (AV):

  $$\mathbf{AV}=\sum_{i=1}^{n} LT_i / n$$

  where n is the number of tasks, LT is the length of the task.

- Execution Time (ExT) is the time taken to execute the tasks on VM

  $$ExT = LT / VMP$$

  where VMP is the processing power of VM

- Completion Time (CT) is the sum of the execution time of all the previous tasks and ExTm of the present task allocated in the same VM.

  $$CT= ExT_m + \sum_{i=1}^{m-1} ExT$$

- Makespan is the amount of time needed to complete a group of tasks i.e., the completion time of last task.

  $$Makespan= CT_n$$

  where CTn is the time taken to complete the last task.

- Response Time (RT) represents the time needed to respond by the load balancing and this metric must be reduced

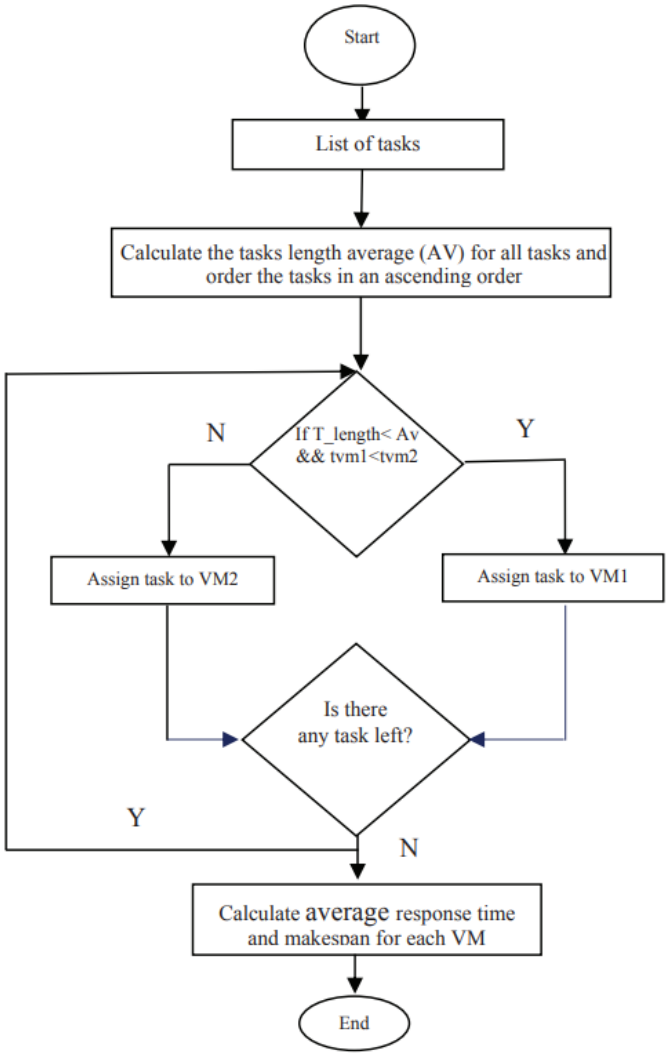
$$\mathbf{RT}= \sum_{i=1}^{n} CT_i + SB_i$$

CT: completion time of task, SB: Submission time of task

- Average of response time for each VMs

$$Avg.RT=RT/N$$

N: number of tasks in the VM



IMPLEMENTATION

The simulator used for this project is CloudSim 3.0.3.

Installation:

- Prerequisites:
  - Object oriented aspects of Java programming and its Collections interface.
  - Knowledge of cloud computing fundamental concepts.
- CloudSim is available on GitHub here.
- Download the CloudSim 3.0.3 zip file.
- Open the package in any IDE which supports Java.

Configuration:

Configure the parameters of the virtual machines and hosts as shown below

Configuration of the VMs

VM No.	MIPS	Memory	Bandwidth	Storage	Scheduler Type
VM1	1000	512 Mb	1000 Mb	10000 Mb	SpaceShared
VM2	8000	512 Mb	1000 Mb	10000 Mb	SpaceShared

Configuration of the Host

MIPS	Memory	Bandwidth	Storage
10000	16384	10000	1000000

Major built in classes in clousim:

- Cloudlet.java – creates cloudlets (user tasks)
- Vm.java – creates virtual machines
- DatacenterBroker.java – creates a broker which is the mediator between all communications taking place between the user and the datacenters.
- Datacenter.java – creates a power datacenter
- DatacenterCharacteristics.java – to set the characteristics of the datacenters such as system architecture, operating system, virtual machine manager, etc.

Implementation:

- Initialize the cloudsim package.

```
Cloudsim.init()
```

- Create datacenter

```
createDatacenter()
```

- Create broker

```
createBroker()
```

- Create VMs and cloudlets

```
createVM()
createCloudlet()
```

- Start the simulation

```
Cloudsim.startSimulation()
```

- Stop the simulation

```
Cloudsim.stopSimulation()
```

- Print the list of cloudlets

```
printCloudletList()
```

Working:

- The user first configures the parameters for cloudlets and virtual machines.
- We have used one datacenter and two virtual machines.
- User submits the list of cloudlets and VMs to Datacenter Broker.
- The Broker then submits this list to the datacenter according to modified SJF scheduling algorithm.
- The tasks are allocated to hosts in the datacenter which are then executed.
- The execution results are sent to the broker.
- The user obtains the results from the broker.
- Print the results, note the makespan and response time for individual VMs.
- Repeat the procedure by changing the number of cloudlets.
- Compare the results with FCFS and SJF algorithms.

RESULTS AND DISCUSSIONS

We implemented a modified SJF algorithm and compared the result with traditional SJF and FCFS algorithms. The results were compared by running the simulation thrice each time varying the number of cloudlets and the average response time and makespan for each VMs were tabulated as shown below.

FCFS

Task no.	**Average Response Time**		**Makespan**	
	VM-0	**VM-1**	**VM-0**	**VM-1**
Experiment 1 (20 tasks)	156.02	18.59	340.98	39.91
Experiment 2 (40 tasks)	422.03	49.84	1021.49	120.98
Experiment 3 (60 tasks)	798.03	94.84	2031.98	243.29

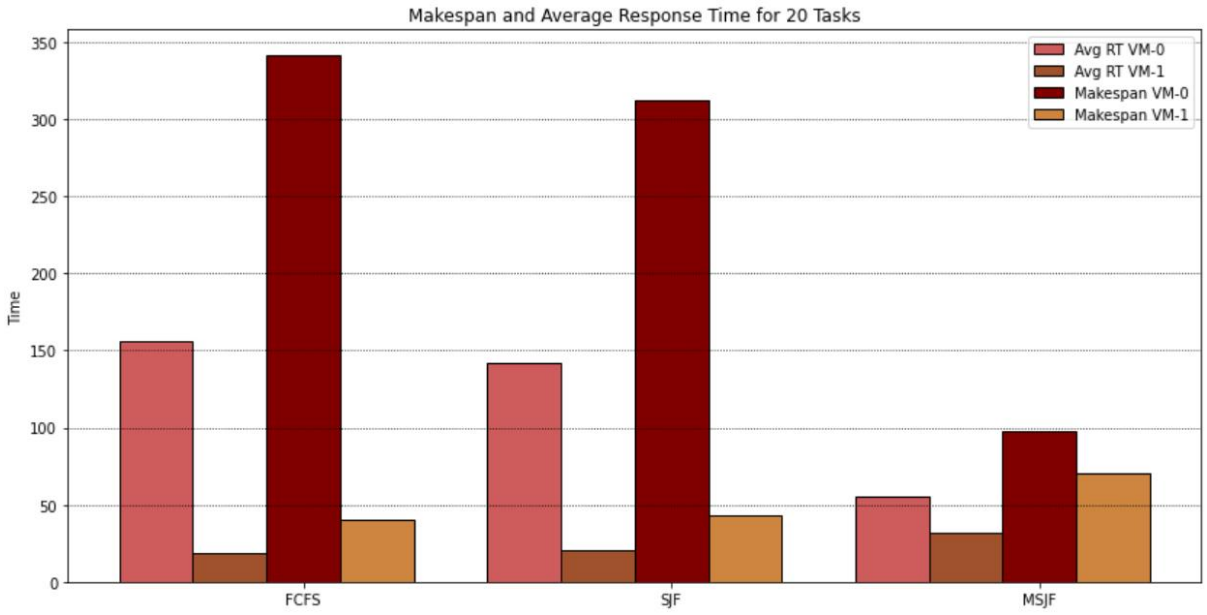
SJF

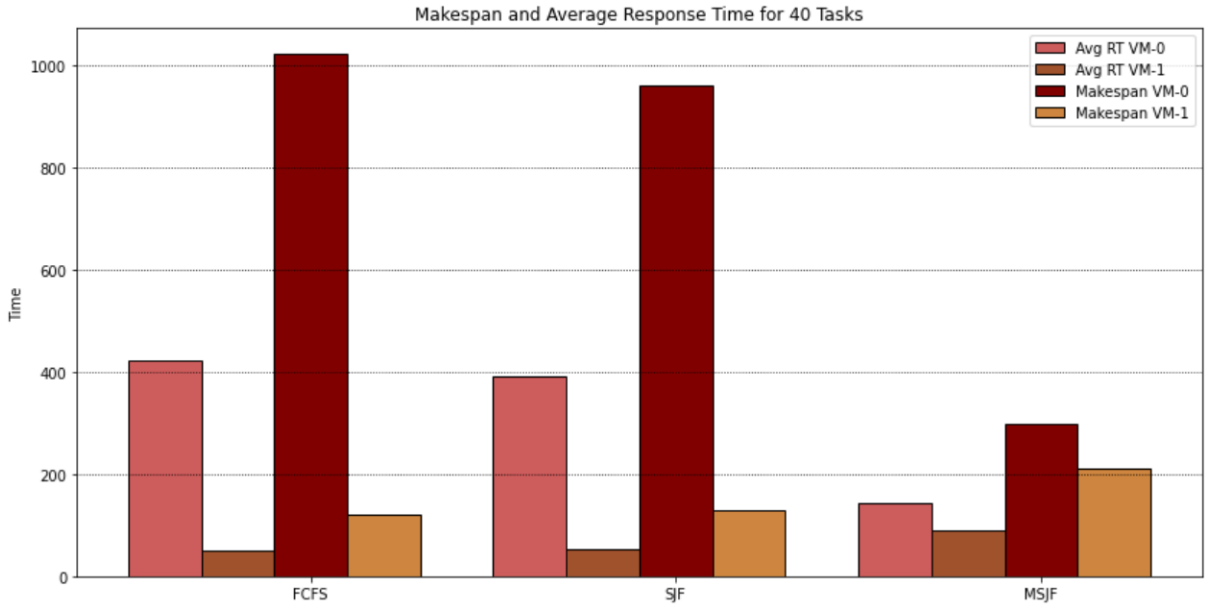
Task no.	**Average Response Time**		**Makespan**	
	VM-0	**VM-1**	**VM-0**	**VM-1**
Experiment 1 (20 tasks)	141.54	20.39	312.1	43.51
Experiment 2 (40 tasks)	391.54	53.64	960.6	128.57
Experiment 3 (60 tasks)	751.54	100.64	1939.1	254.89

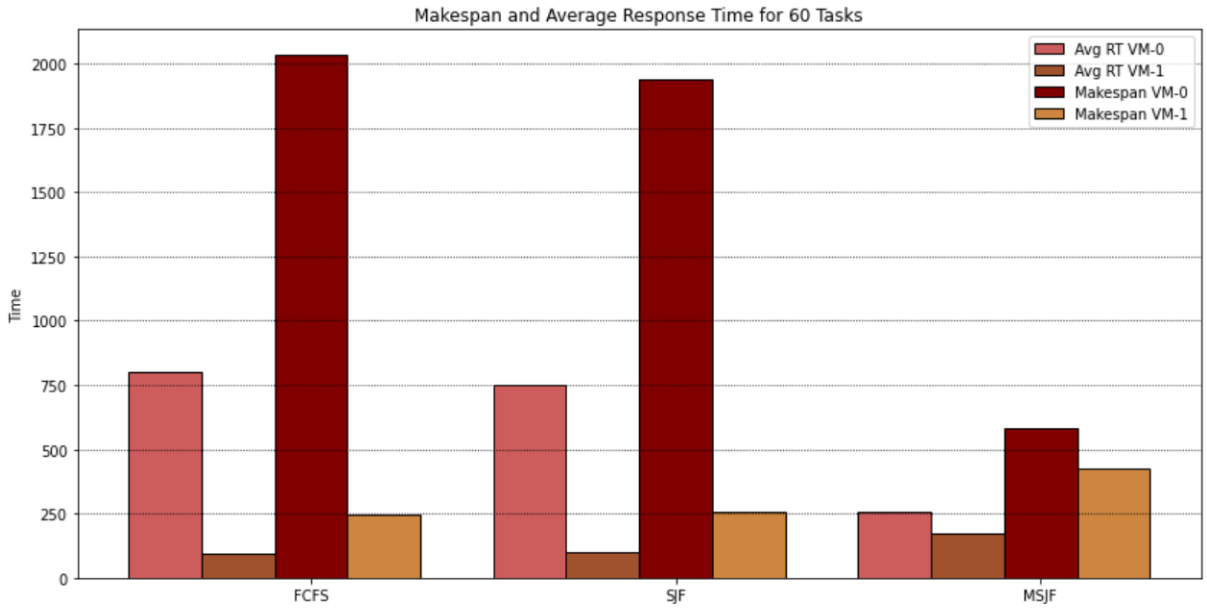
Modified SJF

Task no.	**Average Response Time**		**Makespan**	
	VM-0	**VM-1**	**VM-0**	**VM-1**
Experiment 1 (20 tasks)	55.65	31.63	97.5	70.34
Experiment 2 (40 tasks)	141.89	88.26	297.5	211.46
Experiment 3 (60 tasks)	255.64	171.24	580.0	424.78

Bar graphs are plotted which shows the variation of average response time and makespan for FCFS, SJF and MSJF scheduling algorithms as shown below.







From the above graphs, it can be observed that the Modified SJF algorithm performs more efficiently compared to FCFS and SJF algorithms i.e., the average response time and makespan is lower in MSJF than in the other two algorithms.

CONCLUSION AND FUTURE SCOPE

The improvement in completion time of last task i.e., makespan time which was necessary has been implemented in this project using CloudSim simulator. We have observed that Modified SJF has comparatively lesser response time and thus higher efficiency when compared to traditional SJF as well as FCFS algorithms. The experiment is repeated each time taking different number of cloudlets and results are tabulated.

The future enhancement that could be done is considering the Quality of Service (QoS) factors such as deadline constraints. The number of hosts in a datacenter is fixed to one in this project, which could be varied and accordingly the results can be analyzed.

REFERENCES

[1] Alworafi, Mokhtar & Dhari, Atyaf & Al-Hashmi, Asma & Darem, Abdulbasit & Dr, Suresha. (2016). An Improved SJF Scheduling Algorithm in Cloud Computing Environment. 10.1109/ICEECCOT.2016.7955216.

[2] Sun, Hong & Chen, Shi-ping & Jin, Chen & Guo, Kai. (2013). Research and Simulation of Task Scheduling Algorithm in Cloud Computing. TELKOMNIKA Indonesian Journal of Electrical Engineering. 11. 10.11591/telkomnika.v11i11.3513.

[3] Pratap, Ram & Zaidi, Taskeen. (2018). Comparative Study of Task Scheduling Algorithms through Cloudsim. 397-400. 10.1109/ICRITO.2018.8748514.

[4] F. Alhaidari, T. Balharith and E. AL-Yahyan, "Comparative Analysis for Task Scheduling Algorithms on Cloud Computing," 2019 International Conference on Computer and Information Sciences (ICCIS), 2019, pp. 1-6, doi: 10.1109/ICCISci.2019.8716470.

[5] Q. -H. Zhu, H. Tang, J. -J. Huang and Y. Hou, "Task Scheduling for Multi-Cloud Computing Subject to Security and Reliability Constraints," in IEEE/CAA Journal of Automatica Sinica, vol. 8, no. 4, pp. 848-865, April 2021, doi: 10.1109/JAS.2021.1003934.

[6] Gawali, M.B., Shinde, S.K. Task scheduling and resource allocation in cloud computing using a heuristic approach. *J Cloud Comp* **7,** 4 (2018). https://doi.org/10.1186/s13677-018-0105-8

[7] Ramezani, F., Lu, J., Hussain, F. (2013). Task Scheduling Optimization in Cloud Computing Applying Multi-Objective Particle Swarm Optimization. In: Basu, S., Pautasso, C., Zhang, L., Fu, X. (eds) Service-Oriented Computing. ICSOC 2013. Lecture Notes in Computer Science, vol 8274. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-45005-1_17

[8] Gupta, Ambika. (2021). A modelling & Simulation via CloudSim for Live Migration in Virtual Machines. IOP Conference Series: Materials Science and Engineering. 1116. 012138. 10.1088/1757-899X/1116/1/012138.

[9] Salman, Haitham & Ali, Raniah & Thabit, Kawther. (2018). Study and Implementation of Resource Allocation Algorithms in Cloud Computing. 7. 591-594.

[10] Manzoor, Muhammad Faraz & Abid, Adnan & Farooq, Shoaib & Ahmed, Naeem & Farooq, Uzma. (2020). Resource Allocation Techniques in Cloud Computing: A Review and Future Directions. Elektronika ir Elektrotechnika. 26. 40-51. 10.5755/j01.eie.26.6.25865.

[11] E. Rani and H. Kaur, "Study on fundamental usage of CloudSim simulator and algorithms of resource allocation in cloud computing," 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2017, pp. 1-7, doi: 10.1109/ICCCNT.2017.8203998.

[12] M. Pandey and S. K. Verma, "Cost based resource allocation strategy for the cloud computing environment," 2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), 2017, pp. 1-7, doi: 10.1109/ICCCNT.2017.8204170.

[13] Madhusudhan H S , 1 Satish Kumar T , 2 S.M.F D Syed Mustapha , 3 Punit Gupta , 4 and Rajan Prasad Tripathi 5, “Hybrid Approach for Resource Allocation in Cloud Infrastructure Using Random Forest and Genetic Algorithm”, 2021, https://doi.org/10.1155/2021/4924708

[14] R. R. Kumar, S. K. Jha, D. Garg and S. Vaishnav, "Evaluation of Load Balancing Algorithm Using Cloudsim," 2018 3rd International Conference on Inventive Computation Technologies (ICICT), 2018, pp. 78-81, doi: 10.1109/ICICT43934.2018.9034367.

[15] Afzal, S., Kavitha, G. Load balancing in cloud computing – A hierarchical taxonomical classification. *J Cloud Comp* 8**,** 22 (2019). https://doi.org/10.1186/s13677-019-0146-7.