Turkish Online Journal of Qualitative Inquiry (TOJQI)
Volume 12, Issue 04, April 2021: 2295-2309

Research Article

Agent based Scheduling Policies for Datacenter Environment

Sindhuja Karthikeyan[1], Shakthy Abiraami P[2], Sumuja[3], Guntamadugu Nikitha[4]

[1]Assistant Professor, Department of Computer Science and Engineering,
RMK Engineering College, Thiruvallur, India
[2,3,4]Under-Graduate Student, Department of Computer Science and Engineering, RMK Engineering College, Thiruvallur, India
Email: ksa.cse@rmkec.ac.in, shak17306.cs@rmkec.ac.in, sumu17317.cs@rmkec.ac.in, gunt17130.cs@rmkec.ac.in

Abstract

Cloud computing is a modern technology that allows the usage of IT infrastructures and services that do not seem to be installed on an area computer or server. The usage of clouds has become large in recent times. With this extensive usage, there are some issues arising because of inefficient job scheduling and resource allocation. These issues might cause dissatisfaction of users because of large waiting time to access resources and decline in cloud service providers' benefits. This might lead to high energy consumption, so as to eradicate this issue, an efficient job scheduling algorithm is proposed to cut back the delay time and energy consumption that tries to unravel the complexity involved during thescheduling phase. Main principle of this work is to make agents to act as a scheduler that dynamically assign VM resources for a user job within deadline. Experimental evaluation shows efficient execution of scheduling requests using agent in parallel manner than that of using traditional scheduling algorithm without agents.

Keywords: *Cloud computing; scheduling; Resource allocation; Agents*

Introduction

The on-demand availability of ADP machine resources, mainly information storage (cloud storage) and computing power, without direct dynamic control by the consumer is known as cloud computing. The term is commonly used to describe data centres that are accessible through the internet to multiple consumers. Functions from central servers are frequently distributed through several locations in latesthuge clouds. If the connection to the consumer is close, it may additionally be subjected to an edge server. Clouds may be restricted to one entity (enterprise clouds) or made open to several companies (public clouds).

Cloud computing, according to advocates of public and hybrid clouds, enables businesses to avoid or reduce upfront IT technologycosts.Cloud computing proponents additionally say that it helps organizations to get their applications up and running quicker, with higher manageability and much less support, and that it allows IT teams to more easily adapt resources to meet changing demand and fluctuating, delivering burst computing capability: high computing capacity during peak demand times. Cloud services usually operate on a "pay-as-you-go" basis, which can result in sudden and unexpected running costs if administrators are not familiar with cloud pricing models. Cloud computing has developed as a result of the availabilityofhigh-capacitynetworks, low-cost computers and storage devices, and also the wide availability of hardware virtualization,system architecture, and automated and utility software. Job scheduling is the method of organizing incoming requests (jobs) in a specific order to maximize the use of available resources. Resource allocation (RA) is the method of allocating available machines (resources) to required cloud applications or cloud users over the internet Resource allocation requires the type and amount of machines (resources) needed by each application or users in order to complete a user job/task.

Scheduling and resource allocation plays an important role in cloud computing. Across several datacentres, the management of these crucial roles directly impacts the cloud users and service providers in aspects like user satisfaction and profits respectively. So, these processes are enhanced to surpass the existing models. Agent Cooperation Mechanism (ACM) and Fair Emergency First (FEF) scheduling scheme is used to efficiently plan task execution, increase machines' resource utilization and productivity,minimize production delays, and efficiently handle exceptions[2].

A many-objective optimization algorithm is employed to reduce time and costs, maximize resource utilization and balancing load. Hybrid angle and one by one elimination strategies also are used[3].

A huge number of IIoT(Industrial IoT) nodes must be connected to this IP-based Internet. Existing algorithms are less efficient to figure with nodes of various priorities and also support only one channel. Deep Q Network (DQN) learning algorithm is adopted to style an adaptive resource allocation method toconfirm the precedence transmission of high-priority service data on the idea of expanding network throughput[4].

An increasing number of applications are being built in data centres as cloud technologies continue to enhance. The applications incorporate tasks that involve a group of parallel flows. Traditional techniques result in poor application-level performance.On the idea of Nash bargaining theory, a flow scheduling model with heterogeneousutility characteristics is established, and analyzed using Lagrange multiplier technique and KKT condition. Tasch, a system that allows tasks to keep up high utilities and ensures fairness ofutilitiesis presented [5].

A new job-scheduling policy for the time systems is developed to reduce the processor and memory power consumption. For this purpose, a genetic algorithm(GA) is employed [6].

The efficiency could be enhanced if we create two agents in a single system as job delays are reduced. An algorithm called SPT – M is suggested here wherein several solutions are generated from the initial boundary and by moving the longest processing job from one set to another set that reduces the waiting time [7]. The Green Job shop scheduling aims in reducing the cost involved in energy consumption. To do so, discreet whale optimization algorithm is used [8].

The Jaya algorithm is used to solve the problems in flexible job shop rescheduling. It carries out its execution on ten different skill cases of real life environment [9]. The efficiency is increased in mobile edging computing by employing adaptive service function. It also improves the quality of service [10].

Parallel batch scheduling is done by classifying the tasksinto groups in order to reduce scheduling time. This is achieved by using deterioration and group technology [11]. A deep reinforcement learning based method converts scheduling problems into one learning target and effective strategies that will be learnt autonomously [12].

The datacenters should be dynamic in nature so that the user jobs in execution are not interrupted by any reconfigurations in resources. To achieve this, Adaptive Dominant Resource Fairness is used, whenever a node or resource is free, it gets assigned to a user with lowest current adaptive dominant share [13]. A case study from Alibaba cloud shows the characteristics of co-allocated online services and batch jobs in internet datacenters [14]. The high energy consumption is found to be an issue in a cloud environment. The solution to this is suggested to be dynamically adjusting the available VMs; active hosts are integrated to improve energy and resource efficiency [15].

The major concerns in cloud computing are identified to be delay in scheduling process, complexity in resource allocation and heavy energy consumption. This paper stresses on the concept to resolve the scheduling delay, inefficient resource allocation and reduce cost and memory consumption in a very cloud datacenter. Several solutions are proposed for the identical made use of DQN and HDDL models [1]. These models resolve the scheduling and resource allocation concerns. the matter of increased cost and energy efficiency are surpassed by turning off idle servers and provoking more servers for brand spanking new requests. The usage of agents in situ of schedulers can automate the method of job scheduling and resource allocation at the datacenter. These agents are to be created employing a tool called JADE and it's getting ready to betested in a much simulated environment through a tool called CloudSim.

1.1 Scheduling in Cloud Computing

Scheduling is outlined as mapping a group of tasks with the available virtual machines (VMs) or allocating VMs to run on the available resources so as to satisfy consumers' needs. The main purpose of using scheduling techniques in cloud environments is to enhance system throughput and load balance, maximize the resource utilization, save energy, reduce costs, and minimize the total execution time. Therefore, the scheduler should consider the virtualized resources and consumers' required constraints to urge efficient matching between jobs and resources. Each scheduling technique should be supported by one or more strategies. The most important strategies used are time, cost, energy, QoS, and fault tolerance.

The process of scheduling in a cloud environment is explained in Figure 1. The requests from the users are passed to the scheduler, which helps in assigning the request to several VMs in a parallel manner. This in turn increases the efficiency of the machine. Then finally, it is scheduled and mapped to the nodes.

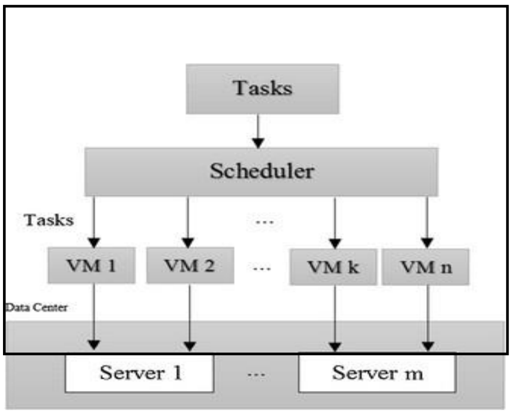

Figure 1 Process of Scheduling in cloud environment

1.2 Scheduling Levels

There are two levels of scheduling available in cloud computing.

1.3 Scheduling at Host Level (VM Scheduling)

1.4 Scheduling at VM Level (Task Scheduling)

At the VM-level, user requests are mapped for execution to the respective VMs using a task/job scheduler. It is called **Task Scheduling**. At the host-level, a VM scheduler is used to allocate the VMs into hardware resources. This type is usually called **VM Scheduling**.
The levels are given in the below Figure 2. in relation with the scheduling algorithms.

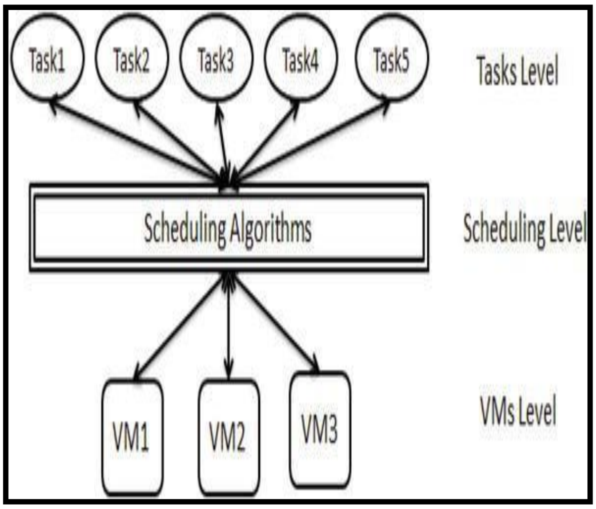

Figure 2 Scheduling System

1.3 Agents in cloud computing

Agents are independent packages used by the consumer or service provider to perform various actions continuously and autonomously. An agent may be used to deliver services or perform any kind of operations based on a schedule. Figure3 gives an overview on agent in the cloud environment.

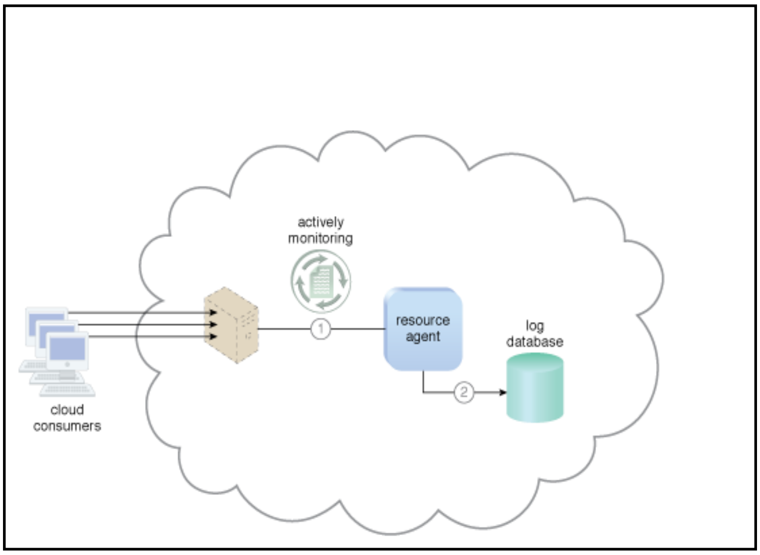

Figure 3 Agent in cloud environment

Proposed Work

The proposed work is to make use of agents to automate the process of scheduling and resource allocation in the cloud simulation environment. This is to be achieved by using an agent-based scheduling algorithm that will help us reduce the delay involved in the process.

Types of Scheduling Algorithms

The traditional scheduling algorithms are used in this paper to show their enhancement in efficiency in terms of time with the usage of agents. The algorithms used are as follows:

First Come First Serve (FCFS) Scheduling*:* First Come First Serve (FCFS) is a scheduling algorithm that automatically executes queued requests and processes so as of their arrival. It is the simplest scheduling algorithm.

Shortest Job First (SJF) Scheduling: Shortest job first, may be a scheduling policy that selects for execution the waiting process with the least execution time.

Round-Robin Scheduling: Round-robin is one among the algorithms employed in computing. Here, time slices are assigned to every task inequal amount and in circular order, handling all tasks without priority.

Priority Scheduling*:* Priority scheduling is one among the common scheduling algorithms in cloud systems. Each process is associated with a priority. Process with maximum priority is to be executed first and the process goes on. Processes with same priority are executed based on FCFS basis.

Table I shown below explains the basic differences between the various scheduling algorithms that we took for consideration in this paper. The arrival time is taken as the criteria for scheduling for FCFS since the scheduling process is said to occur based on the entry of tasks in the system. Similarly, in SJF, the burst time is considered. Burst time refers to the total time the task needs the VM and so the jobs with the least burst time gets executed first and so on. In Round-Robin, time quantum will be set and each task will use the VM for that time slice in a circular manner. Finally, in Priority scheduling, each task will be assigned with a priority number and based on this number the scheduling occurs.

Table 1: Comparison of scheduling algorithms based on implementation

Features	**FCFS**	**SJF**	**Round Robin**	**Priority**
Description	Executes queued requests and processes so as of their arrival	Selects for execution the waiting task with the least execution time.	Time slices are assigne d to every task inequal amount and in circular order	Process with maximu m priority is to be executed first
Model	Non pre-emptive	Non pre-emptive	Pre- emptiv e	Non pre- emptive
Criteria for scheduling	Arrival time	Burst time	Time quantu m and arrival time	Priority
Implementati on	Easy to impleme nt	Cannot be implemente d	Queue	Queue

Table 2: Comparison of scheduling algorithms based on tat and wt

Features	**FCFC**	**SJF**	**Round Robin**	**Priority**
No. of context switches	Low	Low	Depends on time quantum	Low (non pre-emptive) and High (pre- emptive)
CPU engagement	High	Medium	High	High
Throughput time	Low	High	Medium	Low
Turnaround Time(TAT)	High	Medium	High	High
Waiting time (WT)	Medium	Low	High	Low

From Table. 2, it is evident that SJF is efficient compared with the other scheduling algorithms when turnaround time and waiting time are taken into consideration. These factors are medium and low in case of SJF respectively but higher

for other algorithms. The system throughput is also high for SJF but comparatively low for the others. SJF becomes very efficient while considering the CPU Engagement feature. It is medium in SJF but high in case of others. This will in turn reduces the energy consumption of the system which could be advantageous for the cloud service providers (CSP).

2.1 System Architecture

The Architecture diagram of our proposed system is depicted in the Figure 4 and it consists of components like:

- Users.
- Agents.
- Datacenters.

User: The users are the persons who try to access the cloud datacenter to induce the resources for the completion of their task.

Agents: Agents are individual entity that performs various actions on behalf of the user. Here, scheduling is done by these created agents based on the users' request.

Datacenter: A cloud datacenter is where the actual hardware is managed and sent by the cloud company, often with the help of a third-party managed services provider.

2.2 Algorithm

The steps involved in our proposed idea are given as an algorithm. The flowchart depicting the proposed algorithm is shown in Figure 5.

ALGORITHM

1. Input n user request.
2. Nodes get created in cloudsim
3. Consumer agent processes the user request with the help of scheduling algorithm.
4. It checks for the constraints and filters the nodes based on the request.
5. Selected nodes get stored.
6. Agents map the user request to the resources (nodes).
7. Stop.

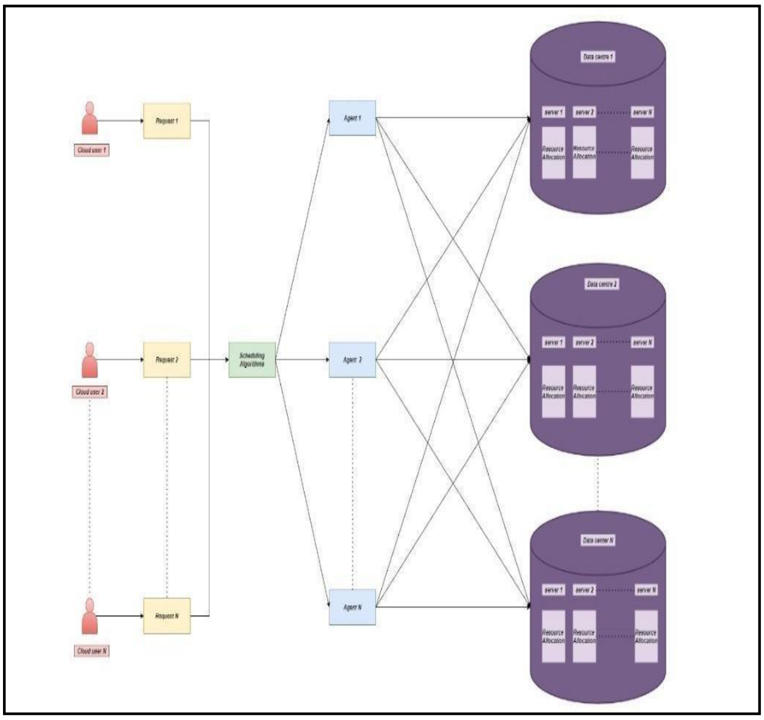

Figure 4. System Architecture

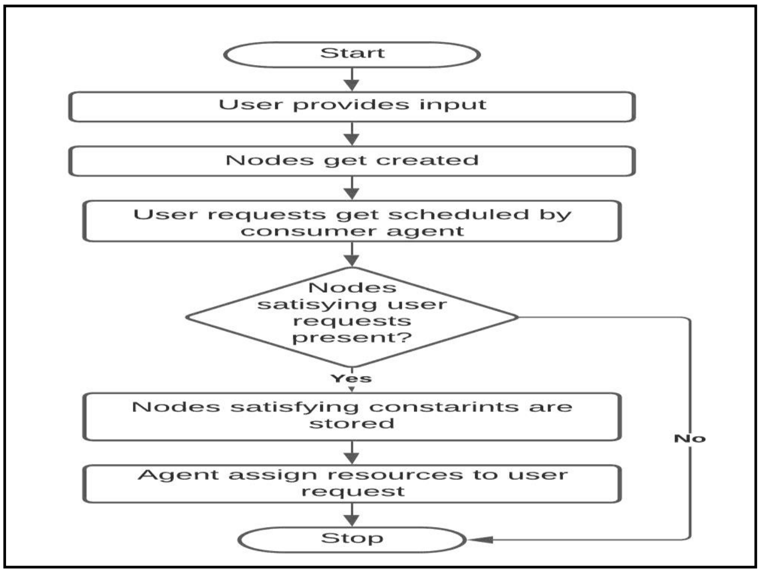

Figure 5 Flowchart of agent based scheduling algorithm

2.3 Workflow

The workflow in our proposed system is divided into two phases. They are,

2.3.1 Agent Creation.

2.3.2 Scheduling and Resource allocation

These phases are given as flowchart in the Figure 6 and Figure 7.

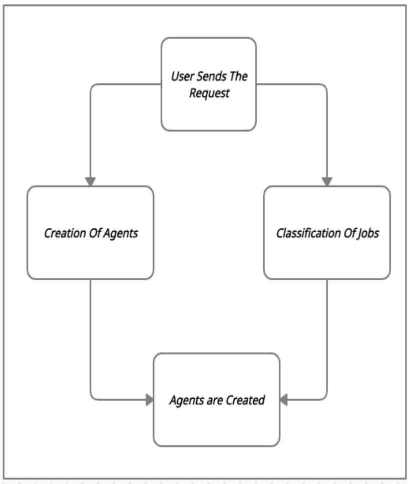

Figure 6 Agent Creation Phase

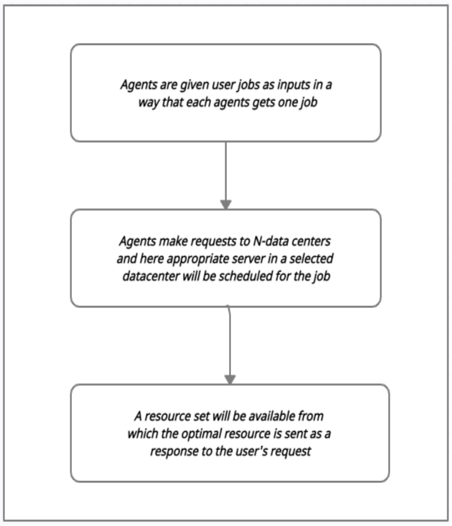

Figure 7 Scheduling and Resource Allocation Phase

The user sends the request to access the resource in a cloud. These requests are termed as Jobs/Tasks. Depending on the number of consumer requests, the respective number of agents will be created in the environment. To provide the user with the resource, two important tasks are to be done. The first phase is the Job scheduling phase, where the user jobs get mapped to an appropriate datacenter in an efficient manner. Our proposed system suggests the use of Agents, to enhance and automate the scheduling phase. Once a datacenter is identified, the task will be subjected to a server, so as the user gets his job done. At the server, the Resource allocation process takes place. The resources which are appropriate to the user's request will be allocated. Then, with the presence of resources, the user request will be executed swiftly.

Results and Discussion

3.1 Execution of scheduling Algorithms

Table 3 illustrates the number of VMs, Data centres, cloudlets considered for the experiment. This is done by simulating the process in a simulated environment called, CloudSim. The experiment is carried over with 50 virtual machines distributed across 5 datacenters. The cloudlets refer to the user jobs/tasks. Here, we considered 100 user requests. The RAM size and other specifications are also given. So, for a total of 100 user request (cloudlets) we are calculating the execution time of scheduling algorithm.

Table 3: Input specifications

Specifications	FCFS	SJF	Round Robin	Priority
No. Of Cloudlets	100	100	100	100
No. of VMs	50	50	50	50
No. of DCs	5	5	5	5
RAM Size	512	512	512	512

3.2 Experimental results without and with the use of agents

The results of our proposed work are discussed in this section. A table has been drawn to explain the execution time of traditional scheduling algorithm without Agent execution and with Agent execution.

Table 4: Experimental results

Scheduling Algorithm	Execution Time (ms)
Without Agent Execution	
FCFS	150.3
SJF	46.62
Round Robin	200.1
Priority	57

With Agent Execution	
FCFS	100.5
SJF	24.17
Round Robin	105.36
Priority	26.96

Based on the above table, we have plotted two graphs which clearly show the time taken by the traditional algorithms to execute a process without and with the use of agents respectively. Figure 8. shows the execution time utilized by the traditional scheduling algorithms like First Come First Serve (FCFS), Shortest Job First (SJF), Round Robin (RR) and Priority Scheduling respectively.

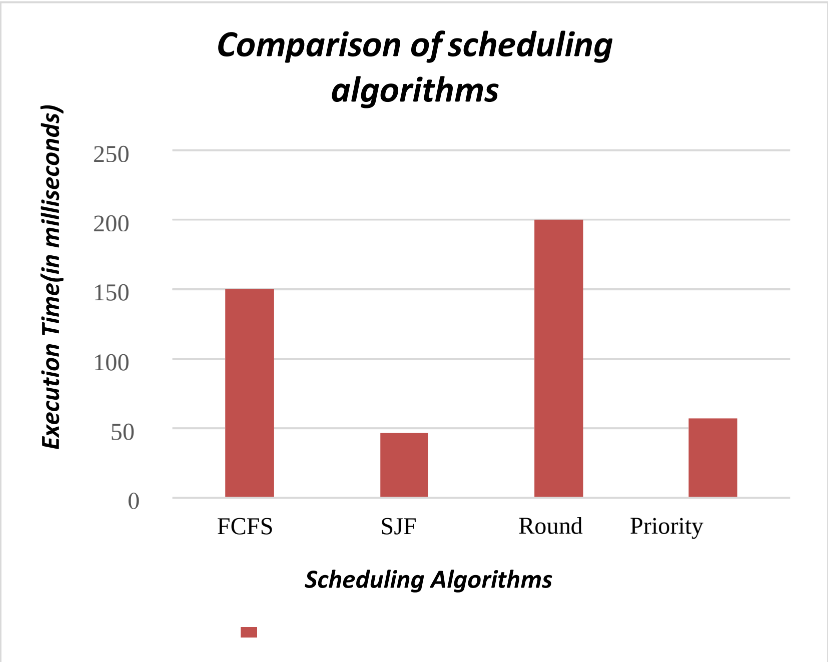

Figure 8. Comparison of various scheduling algorithms without the use of agents

We have taken the parameter as timeto plot the graphs to prove that the usage of agents reduces the time involved in completing a process. This will in turn increase the efficiency of the cloud system and also leads to profit maximization for the cloud service providers (CSP), who are employing agents in their system.

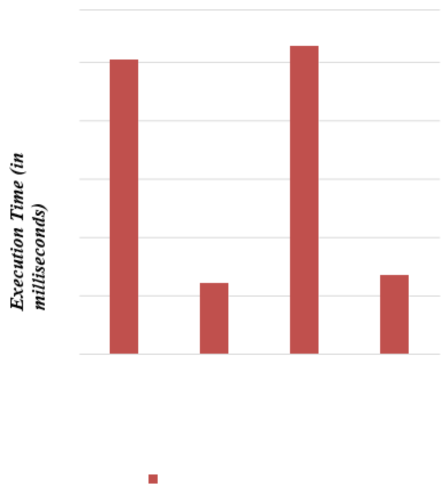

Figure 9. Comparison of various scheduling algorithms with the use of agents

Figure 9. proves that the usage of agents does reduce the time involved in the completion of a process in the cloud environment.

3.3 Performance Analysis

The performance of each of the considered scheduling algorithm is analyzed in this section. The analysis includes the performance of these algorithms with and without the usage of agents. The execution time for FCFS without using agents is found to be 150.3ms whereas with using agents it is found to be 100.1ms only. Similarly, all traditional algorithms are enhanced with the usage of agents.

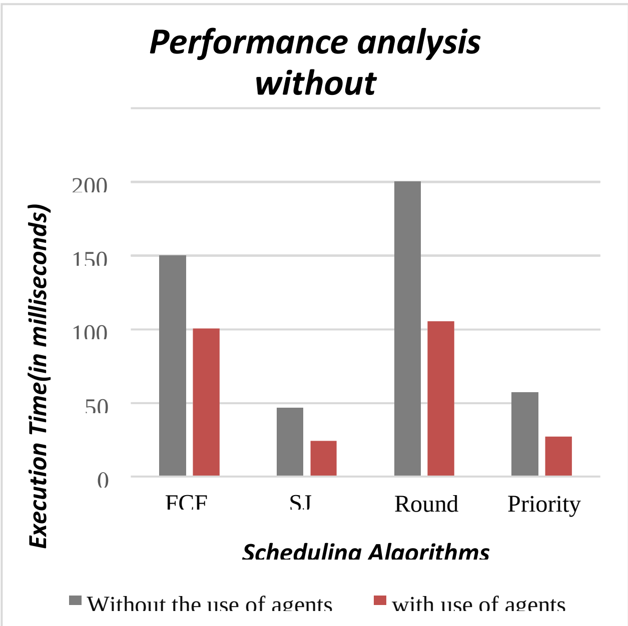

Figure 10. Performance Analysis among traditional algorithms without and with using agents

As we discussed earlier in this section, the usage of agents has an impact in the parameter “time”to a greater extent. Figure 10. helps us to come to a conclusion that the performance of the scheduling algorithms with respect to the execution time (in ms) is high when the agents are not involved and less with the involvement of agents.

Conclusion and Future Work

In this paper, we have analysed and compared scheduling using agents and scheduling without agents. The execution time for processing scheduling request using agents gives better results. Maximum machine resource utilization and productivity rate in agent-based scheduling are achieved as there is reduction in delay time and also due to parallel execution of scheduling request by creating number of VMs. We have carried out this work in a simulated environment. Our future work will include the deployment of our proposed system in a real time cloud environment.

References

[1] Jianpeng Lin, Delong Cui, Zhiping Peng, Qirui Li and Jieguang He, "A Two-Stage Framework for the Multi-User Multi-Data Center Job Scheduling and Resource Allocation", IEEE Access., vol.8, pp. 197863 - 197874, October 2020.

[2] Sehrish Malik and DoHyeun Kim, “A Hybrid Scheduling Mechanism Based on Agent Cooperation Mechanism and Fair Emergency First in Smart Factory”, IEEE Access., vol.8, pp. 227064 – 227075, December 2020.

[3] ShaojinGeng, Di Wu, Penghong Wang and Xingjuan Cai, “Many-Objective Cloud Task Scheduling”, IEEE Access., vol.8, pp. 79079 – 79088, April 2020.

[4] Xiaolong Lai, Quing Hu, Weixin Wang, Li Fei and Ying Huang, “Adaptive Resource Allocation Method Based on Deep Q Network for Industrial Internet of Things”, IEEE Access., vol.8, pp. 27426 – 27434, February 2020.

[5] Fang Dong, Xiaolin Guo, Pengcheng Zhou and Dian Shen, “Task-aware flow scheduling with heterogeneous utility characteristics for data center networks”, TUP., vol.24, pp. 400 – 411, March 2019.

[6] Hyokyung Bahn and Kyungwoon Cho, “Evolution- Based Real-Time Job Scheduling for Co-Optimizing Processor and Memory Power Savings”, IEEE Access., vol.8, pp. 152805 – 152819, August 2020.

[7] Chen-Yang Cheng, Shu-Fen Li, Kuo-Ching Ying and Yu-Hsi Liu, "Scheduling Jobs of Two Competing Agents on a Single Machine", IEEE Access., vol.7, pp. 98702 - 98714 , July 2019.

[8] Tianhua Jiang, Chao Zhang and Qi-Ming Sun, "Green Job Shop Scheduling Problem With Discrete Whale Optimization Algorithm", IEEE Access., vol.7,pp. 43153 - 43166 , March 2019.

[9] Kaizhou Gao, Fajun Yang, Jun-Qing Li, Hongyan Sang and Jianping Luo, "Improved Jaya Algorithm for Flexible Job Shop Rescheduling Problem", IEEE Access., vol.8, pp. 86915 - 86922, May 2020.

[10] Tianfeng Wang, Jiachen Zu, Guyu Hu and Dongyang Peng, "Adaptive Service Function Chain Scheduling in Mobile Edge Computing via Deep Reinforcement Learning", IEEE Access., vol.8, pp. 164922 - 164935, September 2020.

[11] Cuixia Miao, "Parallel-Batch Scheduling With Deterioration and Group Technology", IEEE Access, vol.7, pp. 119082 - 119086, August 2019.
[12] Huayu Zhu, Mengrong Li, Yong Tang and Yanfei Sun, "A Deep-Reinforcement-Learning-Based Optimization Approach for Real-Time Scheduling in Cloud Manufacturing", IEEE Access., vol.8, pp.9987 - 9997, January 2020.
[13] Chaoqun You, Cheng Ren and Lemin Li, "Online Multi-Resource Social Welfare Maximization for Non-Preemptive Jobs", IEEE Access., vol.8, pp. 97920 - 97934, May 2020.
[14] Congfeng Jiang, Guangjie Han, Jiangbin Lin, Gangyong Jia, Weisong Shi and Jian Wan, "Characteristics of Co-Allocated Online Services and Batch Jobs in Internet Data Centers: A Case Study From Alibaba Cloud", IEEE Access., vol.7, pp.22495 - 22508, February 2019.
[15] Chen Li, Lisu Huo and Huangke Chen, "Real-Time Workflows Oriented Hybrid Scheduling Approach with Balancing Host Weighted Square Frequencies in Clouds", IEEE Access., vol.8, pp. 40828 - 40837, November 2019.
[16] Mian Guo, Quansheng Guan, Wende Ke, "Optimal Scheduling of VMs in Queueing Cloud Computing Systems With a Heterogeneous Workload", IEEE Access., vol.6, pp. 15178 - 15191, February 2018.
[17] Belal Ali Al-Maytami, Pingzhi Fan, Abir Hussain, Thar Baker, PanosLiatsis, "A Task Scheduling Algorithm With Improved Makespan Based on Prediction of Tasks Computation Time algorithm for Cloud Computing", IEEE Access., vol.7, pp.160916 - 160926, October 2019.
[18] Fuguang Yao, Changjiu Pu, Zongyin Zhang, "Task Duplication-Based Scheduling Algorithm for Budget-Constrained Workflows in Cloud Computing", IEEE Access., vol.9, pp. 37262 - 37272, March 2021.
[19] Shudong Wang, Tianyu Zhao, Shanchen Pang, "Task Scheduling Algorithm Based on Improved Firework Algorithm in Fog Computing", IEEE Access., vol.8, pp. 32385 - 32394, February 2020.
[20] Wenbo Hu, Jiang Liu, Tao Huang, Yunjie Liu, "A Completion Time-Based Flow Scheduling for Inter- Data Center Traffic Optimization", IEEE Access., vol.6, pp. 26181 - 26193, May 2018.
[21] G Jagadeeswara Rao, G. Stalin Babu, "Analysis Energy of task scheduling algorithms in the green cloud"-2017.
[22] Yashi Goyal, Meenakshi S.Arya, Sunil Nagpal, "Energy efficiency with the help of hybrid policy in green cloud computing", International Conference on Green Computing and Internet of Things-2015.
[23] Nitesh N Naik, Kartheek Kanagala, John P Veigas, “Achieving green computing by effective utilization of cloud resources using a cloud OS", International Conference ON Emerging Trends in Computing, Communication Nanotechnology”-2013
[24] Fatima Shakeel, Seema Sharma-May, "A review on the efficiency of data centers and virtualization of servers”, International Conference on Computing, Communication, and Automation”-2017
[25] Yashwant Singh, Neetesh Mehrotra, “A review on Green IT areas for cloud computing environment”, International Conference on Futuristic Trends on Computational Analysis and Knowledge Management -2015.