International Journal of Engineering & Technology, 7 (2.24) (2018) 81-86

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

A Preference-Based Supervising of Virtual Machines in Cloud Environment

B. Ezhilarasi [1], P. Padmakumari [2], A. Umamakeswari [3]

[1,2,3] *School of Computing, SASTRA Deemed to be University, Thanjavur.*
**Corresponding Author E-mail: ezhilarasirani22@gmail.com*

Abstract

Cloud Computing is a well-known technology in today's world. A large number of users are benefited from the cloud services. The cloud computing must provide efficient service on time for customer satisfaction. So, prominent resource monitoring and scheduling techniques are needed. To achieve the customer satisfaction and to reduce the communication overhead, a method called Resource Supervisor (RS) is proposed. The proposed algorithm assigns the preference for the tasks having highest length, monitor the status of the resource and schedule the tasks to various resources quickly. The proposed method is implemented using Cloud Simulator, and experimental results are validated by comparing RS method with existing algorithms, which provides better outcomes and reduces the communication overhead.

Keywords: *Resource Supervisor, Monitor, Preference factor, Scheduler.*

1. Introduction

Cloud Computing provides several services to the users using virtualized resources. It offers large-scale computing and storage. Cloud users do not need to invest the significant amount of money for getting resources. Instead, they utilize services required and are paid for storage or services they use. All the facilities in cloud depend on monitoring and scheduling of the resources. The monitoring process tracks the idle VMs, or less load VMs for reducing the communication time. The scheduling process [1] is used to split and schedule the user's job in the idle VMs based on the infrastructure, storage, software, security, platform and so on. Load Balancing (LB) of the resource is a significant concern in cloud computing. Suppose, if a VM overloads with multiple tasks, immediately the tasks are distributed to the idle VMs which in turn reduces the overloading. Effective Load Balancing technique can take care of balancing tasks among the VMs. The cloud users can use the cloud services without having high-end hardware and software. The above features make the cloud computing suitable for low-level start-ups.

The cloud uses many scheduling and monitoring algorithms. All algorithms are used to provide efficient services to the customers. The traditional scheduling [2] algorithms are not suitable for a large-scale cloud environment. The scheduling process [3] sends the job to the less loaded VMs. Meanwhile monitoring process predict the idle virtual machines based on RAM size, CPU utilization, Speed of VMs, Power consumption, Distance from the users, Bandwidth, MIPS (Million instruction per second). Before monitoring the resource, find the Preference factor [4] of the job. It reduces the waiting time for a job. Cloudwatch, Azurewatch, Cloudkick, Nimsoft, Logic monitor, Aneka are some of the open sources available for monitoring the cloud resources.

In traditional scheduling algorithms [5], scheduling can do with a limited number of tasks and leads to network traffic. The network problem is solved by implementing suitable and efficient scheduling process. And also, it analyzes the performance of VMs and decides the ideal VM for the task. Priority of the task is a significant concern during resource scheduling process. [6] It may happen to schedule lower priority tasks first. The higher priority tasks have to wait for a long time in a queue. It can reduce the computation speed. Because of the long waiting time, it is necessary to schedule the tasks prioritized based on the essential metrics. It will increase the efficiency of the resources and reduces the execution time of the tasks. The priority scheduling [7] improves the quality of resource utilization and provides the accurate throughput. Load Balancing [8] is also a significant part of task scheduling. In most of the research papers consider monitoring or scheduling. The proposed algorithm encompasses both monitoring and scheduling. It also performs load balancing to balance the load among various virtual machines. Section 2 illustrates the related works of scheduling and monitoring techniques. In section 3, proposed methodologies and techniques have been proposed. Section 4 and Section 5 explains the experiment result, conclusion and result of the proposed paper.

2. Related Works

S. Vadde and S. Ganesan [1] proposed First In First Out (FIFO) job scheduling algorithm. It selects the first job request in the queue and allocates the VM resources. And also, the first job is eliminated first from the queue. So, the FIFO algorithm does not schedule the job based on the priority wise. The task with the prolonged size waits in the queue for a long time. This paper considers the fault in both Single Instalment and Multi Instalment with First In First Out. It also discusses the effect of failure that occurs in FIFO and LIFO (Last In First Out).

M.A. Alworafi [2] presented the Shortest Job First (SJF) scheduling algorithm splits the task depending on the length of the task. So, the task with the shortest time is executed first. But the longest job should wait in the queue for a long time. This paper introduces a Modified Shortest Job First (MSJF) algorithm to reduce the

Copyright © 2018 Authors. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

completion time of the job. It reduces the response time. This MSJF performs load balancing and has calculated the length and the communication time of tasks. It is better than SJF and FIFO.

A. Alnowiser, E. Aldhahri, and A. Alahmad [3] have suggested Round Robin scheduling algorithm based on time. So, each job is executed only for a particular amount of time. But the job was not fully completed before the threshold time. Thus, the scheduler leaves the current job and moves on to the next job. This paper introduces an Enhanced Weighted Round Robin (EWRR) algorithm which increases the performance of scheduler. When compared to the regular Round Robin algorithm, EWRR improves the energy efficiency.

S. Ghanbari and M. Othman [4] proposed Priority job scheduling which performs a significant role in the job allocation techniques. The algorithm avoids the waiting time of the jobs in the queue. The scheduling is based on memory, bandwidth, size of the given task, etc. By using these metrics the user's job is ranked. So, it avoids the longer job waiting in the queue for a long time. This algorithm provides the best response to user's request.

M. Dhingra, J. Lakshmi, and S. K. Nandy [10] proposed resource monitoring architecture called Distributed monitoring framework (DOM). Each host consists of DOM () agent and virtual machine agents. The metrics collector is a cloud front-end and the customer interface module is an interface between the customers and metrics collector.

H. Chen, X. Fu, Z. Tang and X. Zhu implemented a framework [11] called Adaptive Resource Monitoring (ARM) in a cloud. Vector Auto Regression (VAR) mechanism has used for resource prediction in cloud computing. The proposed framework is designed not only for monitoring the resource and also for load balancing and job scheduling. The ARM framework uses two types of modes such as push and pull modes. The prediction method VAR predicts the current resource based on the historical data. The major drawback of this paper is time complexity.

N. Tapoglou and J. Mehnen [12] proposed a concept for job allocation in the cloud. The job allocation is based on capability, availability, and cost of the equipment. The allocation process used a Multi-Criteria decision-making tool. It makes an efficient allocation of the job to manufacture resources. Once the user allocates the job, the machine features are imported by the user. The hierarchical analytical process technique is used to rank the device. This AHP is used to rank the machine in critical decision-making time. The decision-making process not only considers the energy consumption but also increases the allocation of time to various machines.

X. Ji, F. Zeng, and M. Lin [13] proposed a method for monitoring the resource during the transmission of data using Window based Hybrid Push strategy (WHP) technique. If the threshold value is more than the resource status information, the status information is pushed. So the highest weight is assigned to the most recent job. The hybrid push strategy continuously pushes the necessary status during each time interval and also decreases the communication overhead.

3. Proposed System

Cloud computing performs a significant role in IT field. The users mainly prefer to use the cloud services as it is very efficient and cheap. Effective monitoring and scheduling algorithms play a major role in providing the right services. This proposed technique consists of three types of modules. Fig.1 shows the architecture of the proposed method.

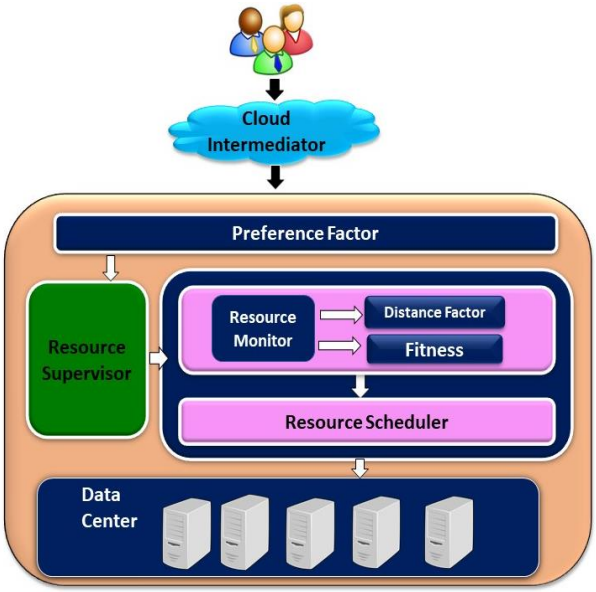

Fig. 1: Architecture diagram of proposed RS

1. Preference Scheduling
2. Resource Monitoring
3. Resource Scheduling

The implementation process starts with giving preference to each user's tasks. The tasks are sorted on the basis of the preference factor. Where in the preference factor is calculated according to the task length.

3.1 Preference Factor

Preference scheduling reduces the execution time of user jobs and balance the load during the resource monitoring. So, the longest task need not wait in the queue for long time. Without being prioritized, the least preferred task has been executed. So, the highly preferred task has been waiting in the queue for a longer time. It leads to the network congestion. In order to reduce the congestion in network, the tasks must be prioritized. Let's as assume the set of tasks

$$T = \{t00, t01, \dots \dots . tmn\} \quad (1)$$

$$resource[m][n] = \begin{matrix} [t00] & [t01] \dots \dots & [t0n] \\ [t10] & [t11] \dots . . & [t1n] \\ [tm0] & [tm1] & [tmn] \end{matrix} \quad (2)$$

Add all the rows and get the value of [n]+. The character n is defined the columns in matrix.

$$[n]+= \{tm0, tm1, tm2 \dots \dots \dots tmn\} \quad (3)$$

The next process is to find the ratio between resource[m][n] and [n]+ values.

$$resource[m][n] = \begin{matrix} [t00/\sum m0] & [t01/\sum m1] \dots & [t0n/\sum mn] \\ [t10/\sum m0] & [t11/\sum m1]. . & [t0n/\sum mn] \\ [tm0/\sum m0] & [tm1/\sum m0]. . & [t0n/\sum mn] \end{matrix} \quad (4)$$

Let us assume that the resource[m][n] is the variable k. The value of k is defined as the ratio between resource[m][n] and [n]+.

$$resource[m][n] = \begin{matrix} [k00] & [k01] \dots \dots & [k0n] \\ [k10] & [k11] \dots . . & [k1n\ ] \\ [km0] & [km1] & [kmn] \end{matrix} \quad (5)$$

It will give the preference vector value of given resource matrix. By dividing the value of pv[m]+ and n, we create the m*n matrix. So, we get the preference factor value of given resource matrix.

$$pv[m]+= \begin{matrix} [x00] \\ [x01] \\ [xm0] \end{matrix} \quad (6)$$

Then perform the above calculations for all the cloudlets. So, get the value of pv[m][n].

$$pv[m][n] = \begin{matrix}[y00] & [y01] \dots \dots & [y0n] \\ [y10] & [y11] \dots \dots & [y1n] \\ [ym0] & [ym1] & [ymn]\end{matrix} \quad (7)$$

The next process is to multiply both 6 and 7 equations. So, we get length of the tasks.

$$F[n] = pv[m][n] * pv[m] \quad (8)$$
$$f[m] = sort(F[m]) \quad (9)$$

The length of the cloudlets sort and define in the above equation. The task having highest length assign with highest preference factor. It provides the accurate result and also predict the preference factor in less time. Algorithm 1 shows the preference monitoring.

Algorithm 1: Preference Scheduling

Input: Incoming user tasks
Output: Prioritized user task

```
1) BEGIN
2) Push the user tasks
3) FOR a=0 to n
4)       FOR b=0 to n
5)       ps[a]= ps[a] + resource [a][b]
6)       ps[a]=ps[a]/3
7) END FOR
8) FOR b=0 to n
9)       p[b]=p[c]* a* ps[b][c]+ p[b]
10) END FOR
11) FOR a=b+1 to n
12) IF(p[b]<p[a])
13)       temp= p[b]
14)       p[b]=p[a]
15)       P[a]=temp
16) Print (Sorted Priorities)
17) END IF
18) END FOR
19) END
```

3.2 Resource Monitoring

Resource monitoring performs a major role in all kind of cloud services. Monitoring is important for both cloud customers and provider. Monitoring [13] measures the performance of virtual machines. So, it is used to identify the current stage of virtual machines or any other resource. The proposed monitoring technique is used to monitor the RAM size of each virtual machine. The result of monitoring technique will provide only high RAM size VMs [14]. Finally, high preference task will be easily scheduled to high RAM size virtual machine. Algorithm 2 explains the proposed resource monitoring.

Algorithm 2: Resource Monitoring

```
Monitor (FV, Dist, size)
1)  BEGIN
2)  Generate the VM RAM size
3)  Find the less load VM
4)  IF(VM==idle)
5)  Calculate FV for all virtual machines
6)  THEN
7)  Compare the FV with reaming VMs
8)  Calculate the Distance value
9)  Dist = Highest individual Resource FV – FV (each Resource FV)
 //Distance calculation
10)  Print the highest RAM size VMs
11)  END IF
12)  END
```

Algorithm 3: Resource Scheduling

```
Scheduler (prior, monitor, cloudlet)
1)      BEGIN
2)      Get the cloudlets from preference factor scheduling
3)      Get the VM size from monitoring algorithm
//Scheduling process
4)      IF (VM1 RAM > VM2 RAM)
5)      Send the task to VM1
6)      ELSE
7)      Send the task to VM2
8)      END IF
9)      Print the scheduling result
10)     END
```

3.3 Resource Scheduling

Scheduling process divides the work on various resources. The major problem in cloud computing is to schedule the task to the resources. Scheduling [15] is used to find the idle resources and allocate the tasks. It also reduces the execution time of task. The scheduling algorithm finds the idle virtual machines and allocates the task to the virtual machines. It also reduces the network congestion. So, it is eligible for large cloud environment. The user allocates [16] the job to the provider. Once provider gets the task, send the job to the datacenter. The datacenter contains broker which schedules the task provided to the virtual machines with minimum execution time. Fitness Value (FV) of VMs produces the accurate result. For example, we take three virtual machines in our experiment. The task with the lowest distance is scheduled to the highest RAM size resource. The distance value calculation finds the highest RAM size virtual machine. So, the scheduler allocates the job to the virtual machine with minimum execution time. Algorithm 3 demonstrates the proposed scheduling algorithm. Fig.2 explains the flow chart of the Proposed resource supervisor method.

4. Experimental Result

The proposed system has implemented by Cloud Simulator [17] with various tasks. In our experiment simulator contains 10 Data center with 5 Host machines and 4 Vms with 3072, 2048, 1024 RAM sizes. The NetBeans IDE uses to implement the proposed Resource Supervisor algorithm.

4.1 System Setting and Resource Supervisor

The CloudSim is used for large number of calculation and process in cloud computing research. It contains default cloudlets, virtual machines, Host machine and Datacenters. The Fig.2 explains the

whole proposed system. First, the user sends the various tasks to cloud service provider.

The task is placed in the queue. Then the first module of preference scheduling is performed. The preference [18] scheduling is taken some tasks from queue and sort the task based on length of the task. The Highest length of the task is placed first in the queue. After preference scheduling, [19] the second module, Resource monitoring process is executed. It determines the less load virtual machines. It finds the highest RAM size virtual machines. So, task need not be waiting in the queue for long time.

During the resource scheduling process, the less load virtual machines are scheduled to the highest preference factor task. Scheduler is to compare all the virtual machines RAM with the highest RAM size virtual machine. Finally, sort the virtual machines based on RAM size. The user task has to be scheduled to resource with minimum execution time. When compared to the existing system, the proposed model reduces the communication problem in network. The Fitness calculation [20] produce the accurate result. So, we can easily find the less load virtual machine.

4.2 Calculation for Preference Factor and Fitness Value of VMs

In the section discuss a sample example of preference scheduling. For example, take three cloudlets and four virtual machines.

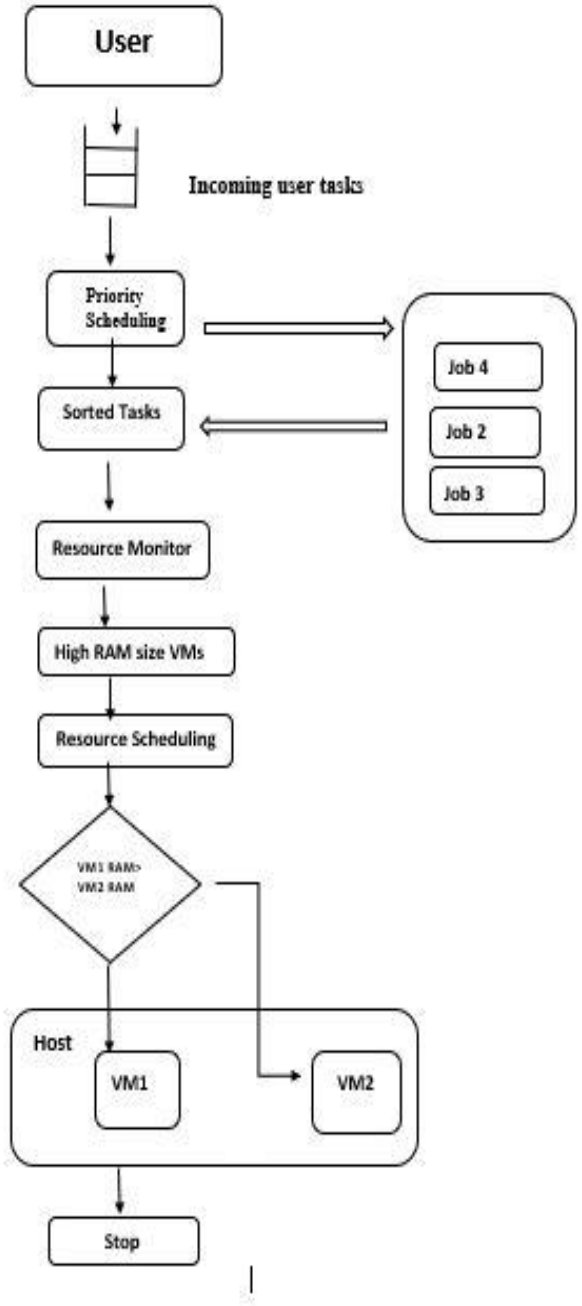

Fig .2 Flow Chart of Resource Supervisor

Table 1: Allocation of task 1

Task1	1	2	3	4
1	2	3	4	5
2	4	3	5	6
3	1	3	5	4
4	6	3	7	5

Table 2: Allocation of task 2

Task2	1	2	3	4
1	4	2	5	7
2	4	1	5	3
3	6	4	3	7
4	5	3	5	2

Table 3: Allocation of task 3

Task3	1	2	3	4
1	5	3	4	5
2	4	5	3	6
3	2	3	4	6
4	3	2	4	5

$$\text{resource}[m][n] = \begin{pmatrix} 13.0 & 19.0 & 14.0 \\ 12.0 & 10.0 & 13.0 \\ 21.0 & 18.0 & 15.0 \end{pmatrix} \quad (10)$$

The result of above values is divide by each task matrix. Therefore,

$$\text{resource}[m][n] = \begin{pmatrix} 0.2739 & 0.1865 & 0.2857 \\ 0.1912 & 0.3127 & 0.2282 \\ 0.3237 & 0.2365 & 0.2155 \\ 0.2453 & 0.5647 & 0.5342 \end{pmatrix} \quad (11)$$

Then find the preference factor value of above result based on comparison of all the tasks result. Finally, sort the value according to descending order. Based on the below result the task 4 having the highest length. So, give the highest preference for task 4.

$$F[m] = \begin{pmatrix} 0.2573 \\ 0.2517 \\ 0.2487 \\ 0.2417 \end{pmatrix} \quad (12)$$

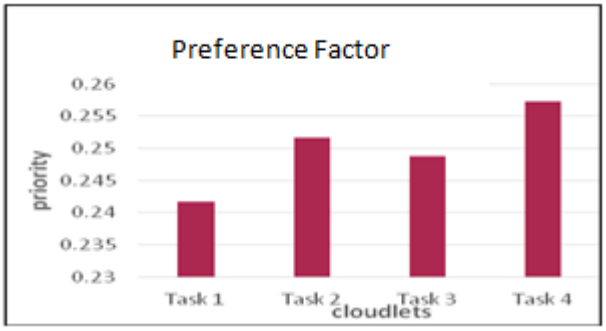

Fig. 3: Measurement of Preference factor

Fig.3 shows the comparison of preference factors measured using various tasks.

Table 4: Virtual machine RAM size

VM id	RAM
0	3072
1	2048
2	512
3	1024

Total RAM size = 3072+2048+512+1024 = 6656
Total Fitness value= 6656
Then calculate the individual fitness value for each virtual machine. So, find the ratio between total RAM size and individual RAM size.
VMid0=2.0
Distance calculation,
VM id 1 = 3.0 D = 6.0-2.0 = 4.0
VM id 2 = 2.0 D = 6.0-3.0 = 3.0
VM id 3 = 6.0

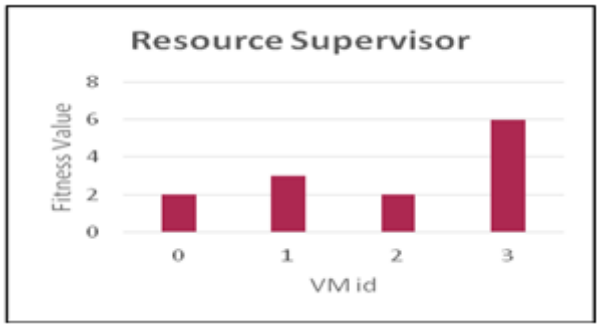

Fig. 4: Shows the allocation of task to the highest fitness VM

Table 5: VM id with fitness and distance Value

VM id	Fitness value	Distance value
0	2.0	4.0
1	3.0	3.0
2	2.0	4.0
3	6.0	0.0

Finally, based on fitness and distance value the prioritized job is sent to VMs which have the highest fitness value. Fig.4 shows the allocation of task to the highest fitness VM.

4.3 Comparison of Various Algorithms with Proposed RS Method

The proposed Resource Supervisor (RS) method is validated with various existing algorithms such as Ant Colony optimization (AC) [5], Honeybee Foraging Algorithm (HFA) [6], Inter Cloud Manager (ICM) [9] algorithm. On comparing the results, the Resource Supervisor (RS) algorithm executes the larger cloudlet with minimum execution time than existing algorithms. In Fig 5 a) shows cloudlet 4 executed within 25 seconds. b) shows the cloudlet 2 executed within 20 seconds. c) and d) shows cloudlet 1 and cloudlet 3 executed within 15 seconds and 10 seconds.

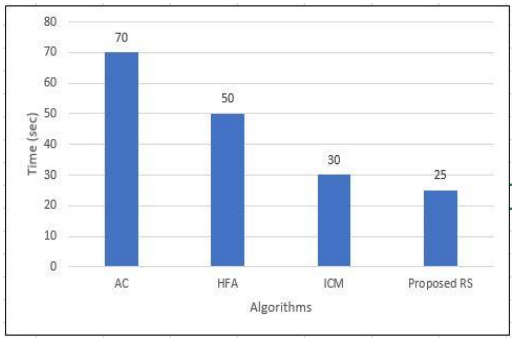

Fig. 5 (a)

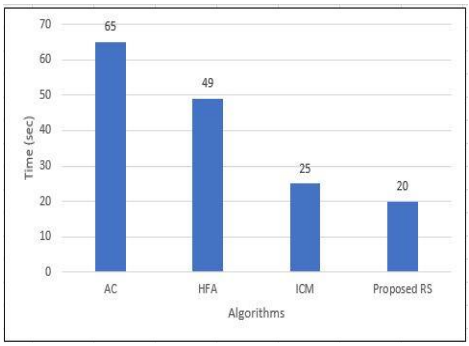

Fig. 5 (b)

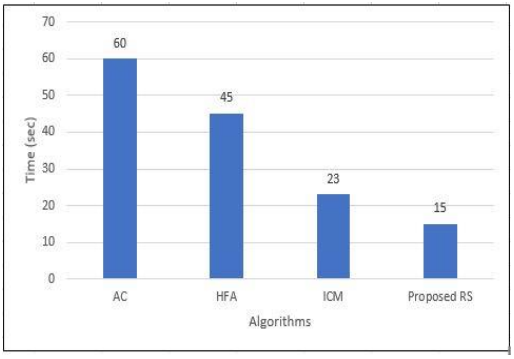

Fig. 5 (c)

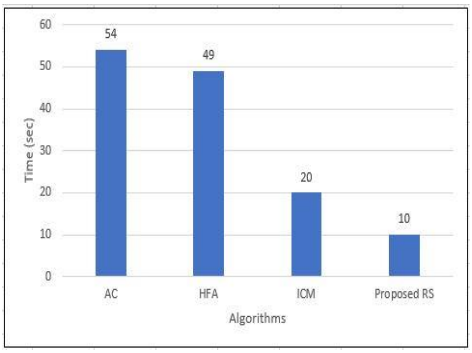

Fig. 5 (d)

Fig. 5: Comparison of various algorithm with proposed RS algorithm

5. Conclusion

In this paper, the Resource Supervisor (RS) method is monitored and schedule the resources with prioritized tasks. The preference scheduling is used to allocate the priority for task which having highest length. As a result, the waiting time of tasks get reduced. The fitness calculation finds the idle or less load virtual machines with minimum processing time. The proposed RS algorithm is increased the customer satisfaction and reduces the communication overhead. So, it is suitable for large number of user tasks, when compared to the existing system.

References

[1] S. Vadde and S. Ganesan, "Effect of fault in single load distribution with FIFO(first in, first out) back propagation of results", IEEE Int. Conf. Electro Inf. Technol, pp. 804–810, 2016.

[2] M. A. Alworafi, A. Dhari, A. A. Al-Hashmi, A. B. Darem, and Suresha, "An improved SJF scheduling algorithm in cloud computing environment", Int. Conf. Electr. Electron. Commun. Comput. Optim. Tech. ICEECCOT 2016, pp. 208–212, 2017.

[3] A. Alnowiser, E. Aldhahri, and A. Alahmadi, "Enhanced weighted round robin (EWRR) scheduling with DVFS technology in cloud", Int. Conf. Comput. Sci. Comput. Intell. CSCI 2014, vol. 1, pp. 320–326, 2014.

[4] S. Ghanbari and M. Othman, "A priority based job scheduling algorithm in cloud computing," Procedia Eng., vol. 50, pp. 778–785, 2012.

[5] Q. Yu, L. Chen, B. Li, and J. Li, "Ant colony optimization applied to web service compositions in cloud computing", Comput. Electr. Eng., vol. 41, pp. 18–27, 2015.

[6] G. J. Tu, M. K. Hansen, P. Kryger, and P. Ahrendt, "Automatic behaviour analysis system for honeybees using computer vision", Comput. Electron. Agric, vol. 122, pp. 10–18, 2016.

[7] M. Lin, Z. Yao, and T. Huang, "A hybrid push protocol for resource monitoring in cloud computing platforms", Optik (Stuttg), vol. 127, no. 4, 2016.

[8] R. Kaur, "Load Balancing in Cloud System using Max Min and Min Min Algorithm", pp. 31–34, 2014.

[9] B. Kang and H. Choo, "A cluster-based decentralized job dispatching for the large-scale cloud", Eurasip J. Wirel. Commun. Netw, no. 1, pp. 1–8, 2016.

[10] M. Dhingra, J. Lakshmi, and S. K. Nandy, "Resource usage monitoring in clouds", IEEE/ACM Int. Work. Grid Comput., vol. 12, pp. 184–191, 2012.

[11] H. Chen, X. Fu, Z. Tang, and X. Zhu, "Resource Monitoring and Prediction in Cloud Computing Environments", 3rd Int. Conf. Appl. Comput. Inf. Technol. Int. Conf. Comput. Sci. Intell., pp. 288–292, 2015.

[12] N. Tapoglou and J. Mehnen, "Cloud-based Job Dispatching Using Multi-criteria Decision Making", Procedia CIRP, vol. 41, pp. 661–666, 2016.

[13] X. Ji, F. Zeng, and M. Lin, "Data transmission strategies for resource monitoring in cloud computing platforms", Optik (Stuttg)., vol. 127, no. 16, pp. 6726–6734, 2016.

[14] M. R. Abid, K. Kaddouri, K. Smith, M. I. El Ouadghiri, and M. Gerndt, "Virtual machines' load-balancing in inter-clouds", Int. Conf. Futur. Internet Things Cloud Work. W-FiCloud, pp. 109–116, 2016.

[15] V. C. Emeakaroha, I. Brandic, M. Maurer, and S. Dustdar, "Low Level Metrics to High Level SLAs-LoM2HiS Framework_Bridging the Gap Between Monitored Metrics and SLA Parameters in CLoud Environments.pdf", pp. 48–54, 2010.

[16] N. Srinivasu, "a Dynamic Approach To Task Scheduling in Cloud Computing Using Genetic Algorithm", vol. 85, no. 2, 2016.

[17] D. Atanasov and T. Ruskov, "Simulation of Cloud Computing Environments with CloudSim", pp. 2–6, 2014.

[18] E. Meriam and N. Tabbane, "Dynamic Scheduling Protocol Based on Cost in Cloud Computing", Glob. Summit Comput. Inf. Technol., pp. 15–20, 2016.

[19] L. Adhianto et al., "HPCTOOLKIT: Tools for performance analysis of optimized parallel programs", Concurr. Comput. Pract. Exp., vol. 22, no. 6, pp. 685–701, 2010.

[20] D. Saxena, R. K. Chauhan, and R. Kait, "Dynamic Fair Priority Optimization Task Scheduling Algorithm in Cloud Computing: Concepts and Implementations", Int. J. Comput. Netw. Inf. Secur., vol. 8, no. 2, pp. 41–48, 2016.

[21] T.Padmapriya and V.Saminadan, "Utility based Vertical Handoff Decision Model for LTE-A networks", International Journal of Computer Science and Information Security, ISSN 1947-5500, vol.14, no.11, November 2016.

[22] M. Rajesh, Manikanthan, "ANNOYED REALM OUTLOOK TAXONOMY USING TWIN TRANSFER LEARNING", International Journal of Pure and Applied Mathematics, ISSN NO: 1314-3395, Vol-116, No. 21, Oct 2017.

[23] T.Padmapriya, S.V.Manikanthan, "An enhanced distributed evolved node-b architecture in 5G tele-communications network" , International Journal of Engineering & Technology, DOI: 10.14419/ijet.v7i2.8.10419, ISSN NO:2227-524X, Vol-7, No.2.9(2018)