EFFICIENT JOB SCHEDULING USING TASK CLASSIFIER AND FIREFLY OPTIMIZATION IN CLOUD

C.R. Durga devi*, Dr. R. Manicka Chezian**

* Ph.D. Scholar, ** Associate Professor
Department of Computer Science, Nallamuthu Gounder Mahalingam College,
Pollachi, Tamil Nadu, India

Abstract

Cloud computing is the assemblage of computing resources which are conveyed as a facility to the client or multiple tenants over the internet. Job scheduling is an indispensable and utmost vital part in any cloud environs. With growing digits of customers, scheduling becomes a determined task. Identifying the best Job scheduling method is a significant challenge to improve scheduling efficiency and minimizes the makespan in big data analytics. Task classification based on certain parameters using Ensemble classifier and firefly optimized Scheduling (FOSC) technique is introduced for scheduling large quantity of jobs to ideal virtual machine with least possible time. The classification and FOSC technique maximize the resource utilization rate across the cloud server while handling massive amount of tasks. Task classifier categorizes the tasks using Ensemble classifier based on priority. The priority level of the task is calculated based on certain parameters like task size, bandwidth and memory expectation of the task and assign to different data centers. In the Data center, the tasks are stored in one or more queue and then selection of optimal virtual machines is performed. Followed by, the tasks get scheduled using firefly optimized Round Robin Scheduling algorithm. The FOSC efficiently identify the resource optimized virtual machine and allocate the task. This helps to maximize scheduling efficiency of the cloud server.

Index Terms: Job Scheduling, Task classification, Ensemble classifier, firefly optimized scheduling, Round Robin

I. INTRODUCTION

Cloud computing and Big Data shows a very substantial role in several areas like Government sector, medical field, and IT sector and so on. Dynamic usage of resources develops a challenging mission in big data analytics. Big data analytics is achieved by machine learning algorithms to handle the data and identify significant information out of it. In big data analytics, tasks are distributed across several virtual machines to decrease job completion time as well as traffic level. Consequently, a suitable task scheduling is exploited for achieving a better service provisioning in big data analytics.

Several data mining models has been developed in the recent days to perform task scheduling with big data. Task classification and firefly optimized Scheduling (FOSC) technique is introduced in this paper. The main contribution of the technique is summarized as follows,

Grouping of incoming tasks can be performed either using clustering or classification. Ensemble classification model is used to classify the incoming tasks. Ensemble classifier is a set of classifiers in which the final decision of classification is associated with all the classifiers decisions. Resources need of the tasks are identified and then distribute to data center available with required resources. The contributions of FOSC technique are to improve task scheduling efficiency, resource utilization rate and minimize the time. This contribution is achieved by hybrid application of task classification and firefly optimization algorithm.

Firefly optimized Scheduling is applied to find the resource optimized virtual machine among the number of the virtual machines based on light intensity. The virtual machine which utilizes the minimum resource is chosen for handling high priority task. The high priority tasks are scheduled to optimal virtual machine with minimum scheduling time. This assists to improve the scheduling efficiency.

This paper is ordered as follows. Section 2 provides literature survey. Section 3 provides the sketch of RFOS technique. Trial sets of RFOS technique are described in section 4. Section 5 gives experimental results and discussions finally, the conclusion of the paper is presented in section 6.

II. LITERATURE SURVEY

Arnav Wadhonkar and Deepti Theng proposed scheduling policy that considers both the parameters Task length and task deadline for scheduling the jobs in [1]. This helps to improve the performance by reducing the makespan when compared with techniques that considers any one of these two parameters. But this policy failed to consider an important parameter, the Task Priority as input. Fakhrosadat Fanian et al. [2] suggested a combination of firefly algorithm (FA) and simulated annealing (SA) used for scheduling. The result is compared with existing algorithms like min-min, max-min, firefly, and simulated annealing in reducing make span and balancing workload on machines but it failed to consider the resource allocation.

Elmougy et al. proposed a hybrid algorithm based on Shortest Job First and dynamic quantum Round Robin [3].This method split the ready queue into to sub-queues Q1 for short tasks and Q2 long tasks to minimize the waiting time and response time but it failed to reducing task starvation. Mahendra Bhatu Gawali and Subhash K. Shinde [4] uses a heuristic approach that combines the modified analytic hierarchy process, bandwidth aware divisible scheduling, longest expected processing time preemption, and divide-and-conquer methods to perform task scheduling and resource allocation Scientific workflows are taken as input tasks for the system. Comparing with existing BATS and IDEA frameworks, the resource utilization is improved but the respone time was not taken into consideration.

G. Natesan and A. Chokkalingam applied Grey wolf optimization algorithm for scheduling by modifying the hunting equation in order to improve efficiency of correct path of each wolf in searching area in [5]. It improves the performance when compared to Particle swam optimization algorithm in terms of execution time and energy but it failed to consider other parameters like reliability, security and load balancing. S. Loganathan and S. Mukherjee [6] classified all the incoming tasks are into three categories of advance reserved, immediate and best effort. Control management system is used to backfill the best effort job when the resources are available. Pre-emption is applied to increase resource utilization in data center but it failed to consider the job deadline as an input parameter.

K. C. Babu et al. [7] performed Comparative Analysis of Deadline Constrained Task Scheduling Algorithms like GAIN and IaaS-cloud partial critical paths for workflow scheduling. IC-PCP is more efficient than GAIN in terms of all parameters. The makespan value is analysed in terms of cloudlet length, VM's MIPS parameter, bandwidth and size.

Liu et al. [8] implement a Service-Aware Resource Allocation Framework, used for job scheduling by Self-learning classification algorithm which updates the features in feature mapping library and classify the tasks based on their features. Mustafa et al. [9] applied Improve Scheduling Task based Task Grouping. The tasks are classified as storage and computational tasks in order to minimum total tasks completion time. The performance of this task grouping algorithm is better when compared with other algorithm in terms of processing time and cost

Resource Optimized Traffic Aware Gradient Boosting Classification Technique is developed by C.R. Durga Devi and R.Manicka Chezian [10].The tasks are classified as immediate task and reserved task based on the priority level using ensemble classifier. H. Choi et al. [11] uses Task Classification Based Energy-Aware Consolidation Algorithm (TCEA) which classifies the tasks as data- intensive task and computation-intensive tasks. The classification data are stored in separate files for future reference. VM consolidation is done that uses a double threshold scheme for energy reduction. H. Gamal El Din Hassan Ali [12] performed Grouped tasks scheduling algorithm based on QoS, classify the tasks based on the priority which is built on the attributes like type of users, expected scheduled priority of tasks, length or load of tasks and latency of tasks. The performance of this algorithm is measured in terms of execution time, load balancing, average latency in various conditions applied, by comparing with MAX-MIN algorithm and TS algorithm.

Priority Task Scheduling Strategy for Heterogeneous Multi-Datacenters is implemented by N. Er-raji and F. Benabbou [13] by considering the task age, length and deadline as parameters for setting the task priority level. The virtual machines are also classified and then tasks are assigned to virtual machines. But the implementation is not simulated for big data analytics. C.R. Durga Devi and R.Manicka Chezian [14] use Multivariate logistic regression analysis for identifying priority level of the tasks and applied firefly optimisation algorithm

III. METHODOLOGY

The proposed technique contains two steps as follows. Task classification is the primary step which is performed by the task assigner and the secondary step is scheduling of the task to the virtual machine which is taken care by the scheduler. In cloud environment, various users from different geographical locations will submit their jobs for processing. Jobs contains a number of dependent and independent tasks to be processed to complete the job. In this proposed technique, only independent tasks without pre-emption are taken into consideration.

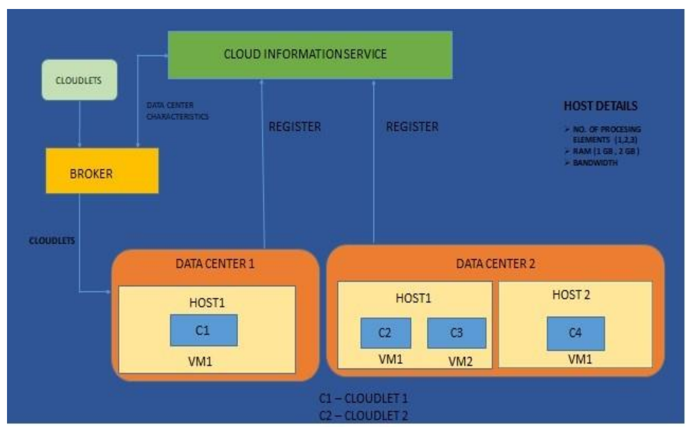

Figure.1 Cloud Architecture

Figure 1 shows the Cloud Architecture diagram which contains functional components like broker, Cloud information service, Data center and cloudlets. The jobs submitted by the users are considered as cloudlets. As soon as the cloudlets are submitted, they are registered with the broker instance. The cloud service providers have datacentres at different geographical locations. Each datacenter contains a numbers of host machines with various number of processing elements, RAM and bandwidth. These hosts are considered as servers where a number of virtual machines are deployed. The cloud information service (CIS)will maintain the resources availability details such as numbers of VM's available in each data centers with their processing capacity, speed, bandwidth availability All the data centers must register their details in this CIS and the data are updated regularly. The broker instance maintains the list of all currently available cloudlets and it gather the resource availability details about all the data centers from CIS. Based on the information gained, the broker will allocate the cloudlet to the precise data center which may handle the task effectively.

A. TASK CLASSIFICATION USING ENSEMBLE CLASSIFIER

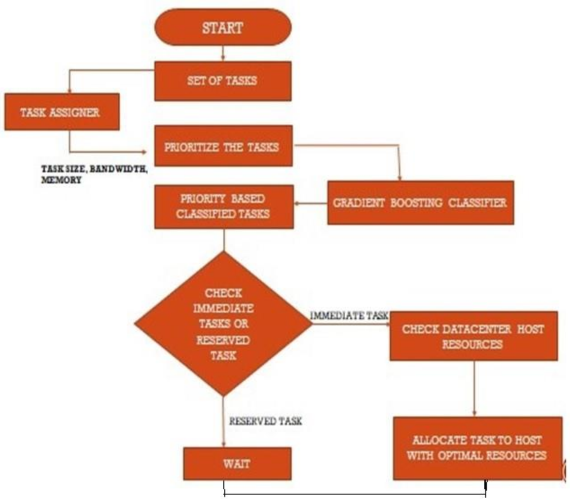

Figure.2 Flowchart for Task Classification

The above flowchart in figure 2 shows the steps performed in the proposed technique. The task assigner will assign priority for all the incoming tasks. There are certain Service Level Agreement (SLA) parameters like task size, length, starting time, finishing time, Bandwidth requirement, Memory needed, deadline associated with the tasks to be considered while handling the tasks. The execution time of the task depends on the VM on which it runs. The main parameters taken for consideration for assigning priority are task size, bandwidth and memory. Let N be the number of tasks and for each parameters, identify the range value for the set of input tasks and compare the task's parameter value with that range value (threshold value) .If the parameter value is higher than threshold value, then assign low priority for that task. Consider the following tasks with their parameter values as follows

Table 1 Assigning priority for Tasks with input parameters

Task id	Task size (MB)	Bandwidth (MB) Expected	Memory (MB) Expected	priority
1	500	4200	720	0
2	1200	5100	650	0
3	700	5600	630	0
4	1400	4700	300	1
5	950	6000	300	0
6	600	4450	450	1
7	720	5700	460	0
8	800	2500	540	1
9	2000	4900	580	0
10	1600	1000	870	0

Table 1 shows the tasks with parameter values with priority assigned for them. Let N =10, for bandwidth parameter MIN value is 1000 and MAX value is 6000. Range is the difference between MAX and MIN value. Consider task 1, Compare the Task size of 500 with the range value 1500 (500 < 1500), and then compare its bandwidth value of 4200 with range value 5000 (4200 < 5000). Finally compare the memory value of 720 with memory range value 570 (870 – 300), as its memory parameter value is greater than range value (720 > 570), low priority is set for that task. If all the parameter values are lesser than their range value, then high priority can be assigned for that task. For example consider task 4 with its task size (1400 <1500), bandwidth value (4700 < 5000) and memory value (300 < 570) all are less than their respective range values, so task 4 is assigned as high priority.

Task assigner set the priority level as high or low to all the tasks based on the above mentioned calculations. This priority assigned tasks are given as training dataset for the gradient boosting classifier. The ensemble classifier first build the base decision tree as its first step. In decision tree construction, identify the features which contain the most information regarding the target feature and then split the dataset along the values of these features. The informativeness is given by a measure called 'information gain'. Entropy is used to measure the impurity or randomness of a dataset. Gini index identify the feature with a lower index value for a split. Among these Gini index is applied as it favours large partitions.

After constructing the base tree, second step is to calculate the loss function as the squared error of difference between actual and predicted value Third step is to calculate pseudo residual and fit the base learner to pseudo residual. Ensemble Classifier combines a weak decision tree classifier outputs. Fourth step is to find the gradient decent step size value. Finally update the model. The output of this classifier identify the tasks as either immediate tasks or reserved tasks. This helps to improve the classification accuracy. The immediate tasks are high priority tasks that should be handled first before the reserved tasks with low priority. The data center and host resources availability can be obtained from CIS and then the broker assigns the immediate task first to optimized data centres and then give second preference to the reserved tasks. As the tasks are classified and the distributed to data centers, it will reduce the Traffic occurrence during task distribution over the data centers. It also reduces the workload among multiple data centers in cloud. The data centers after receiving the tasks, the cloud manager will recheck the priority level of these tasks and then place these tasks in appropriate queues. Firefly optimized scheduling algorithm is used to identify optimal virtual machine to handle the task.

B. IDENTIFICATION OF OPTIMAL VIRTUAL MACHINE BY FIREFLY OPTIMIZATION ALGORITHM

Virtual machines can be identified for selection based on the resource it provides. The main parameters taken into consideration are the time taken by the VM for completing the task, Bandwidth availability and the memory

$$TCT = S_T - E_T \tag{1}$$

In the above equation (1), TCT stands for Task completion time, S_T stands for starting time of the task and E_T stands for ending time of the task.

$$BW = A_{BW} - U_{BW} \quad (2)$$

In the above equation (2), Bandwidth of a VM is measured as the difference between available bandwidth and the unused bandwidth

$$M = T_M - U_M \quad (3)$$

In the above equation (3), The Used memory space of VM is represented as M and it is calculated as the difference between the total memory spaces T_M and the unused memory U_M. In this algorithm, the virtual machines are taken as fireflies. The light intensity INT (VM) of the firefly is calculated as follows by equation (4)

$$INT\ (VM) = (TCT, BW, M) \quad (4)$$

As per the firefly algorithm, the firefly with lesser brightness will move toward the firefly with brighter light. In the same way, the VM with more light intensity is selected as optimal VM and the high priority tasks will be allotted to optimal VM using lesser task completion time, bandwidth and memory Round Robin scheduling policy is used for allocation.

IV. EXPERIMENTAL EVALUATION

CloudSim plus is a java based open source toolkit used to test various user defined algorithms in simulated environment. Trial is conducted using Personal Cloud Datasets taken from http://cloudspaces.eu/results/datasets. Visualization of Task classification is achieved using WEKA and it is integrated with cloudsim plus using Eclipse IDE. The process of simulation contains the steps of initialization, creating cloud service layers which includes creation of datacentres, creating data brokers, creating virtual machines and cloudlets, starting the simulation, getting output results and then stop the simulation

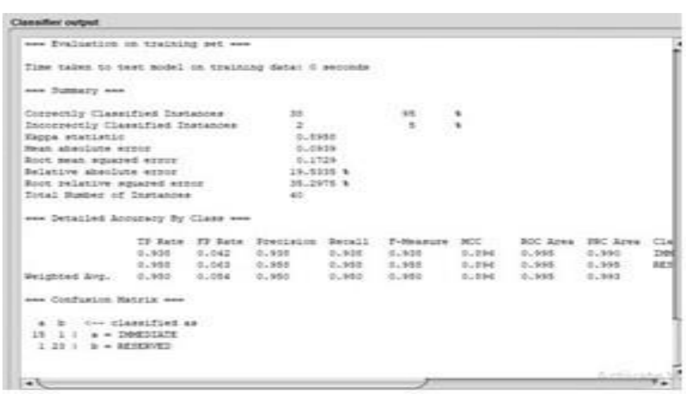

Figure. 3 Classification of Task using WEKA tool

Figure 3 shows the classification of tasks as immediate task and reserved task by applying gradient boosting classifier. The classification accuracy is measured in terms of True positive rate, false positive rate, precision, recall, F- measure and ROC values.

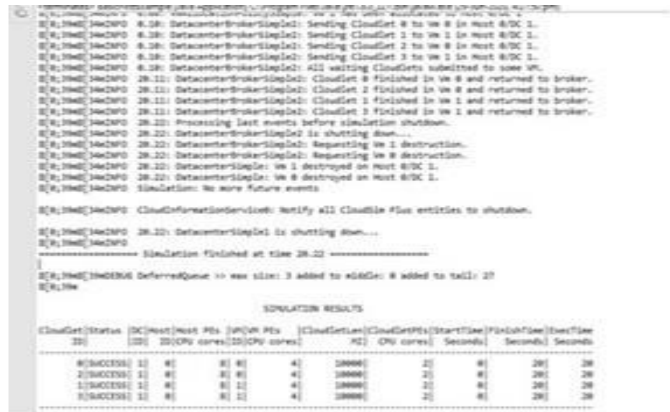

Figure 4. Assignment of cloudlets to VM

The Figure 4 shows the sample output screen from cloudsim plus in Eclipse IDE for assigning four cloudlets to VM0 and VM 1

V. RESULT AND DISCUSSION

The proposed technique of firefly optimized scheduling FOSC is compared with Particle swarm optimization algorithm PSOA and cuckoo search optimization algorithm CSOA. The performance is measured in terms of Scheduling Efficiency, makespan and resource utilization.

Scheduling is the way by which work is assigned to resources that complete the work. Scheduling Efficiency is defined as the ratio of a number of tasks that are scheduled to the virtual machine to the total number of tasks. Performance designates the complete efficiency given by the scheduling algorithm in order to provide worthy services to the consumers as per their requirements. Number of tasks taken for consideration also determines the scheduling efficiency. The proposed method should outperform other methods as the number of tasks increases.

Table.2 Tabulation for Scheduling Efficiency

No. of tasks	Scheduling efficiency (%)					
	No. of tasks correctly schedule	FOSC	No. of tasks correctly schedule	PSOA	No. of tasks correctly schedule	CSOA
25	22	88	20	80	18	72
50	47	94	42	84	35	70
75	69	92	61	81	54	72
100	94	94	88	88	77	77
125	119	95	112	90	101	81
150	142	95	124	83	112	75
175	162	93	144	82	127	73
200	191	96	168	84	149	75
225	219	97	186	83	175	78
250	238	95	219	88	207	83

Table 2 shows the scheduling efficiency with respect to number of tasks. If the number of tasks taken for input is 25, the scheduling efficiency of FOSC is 88 percentage which is higher than PSOA with 80 percent and CSOA with 72 percent. As the number of input tasks increases, the efficiency percentage is also considerably increased when compared with other methods.

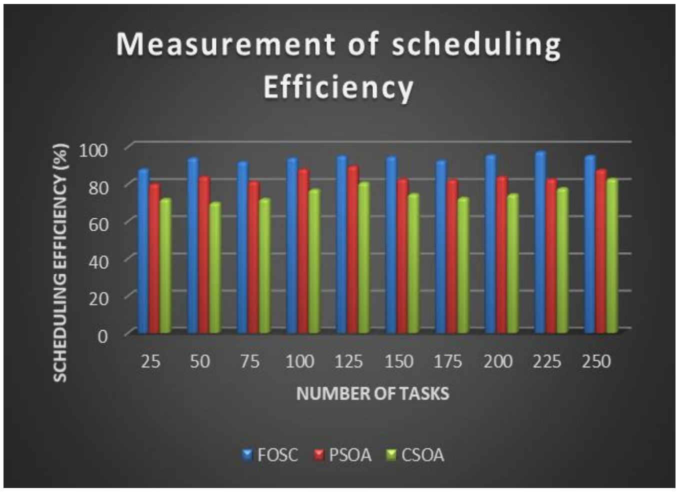

Figure 5. Performance result for scheduling efficiency

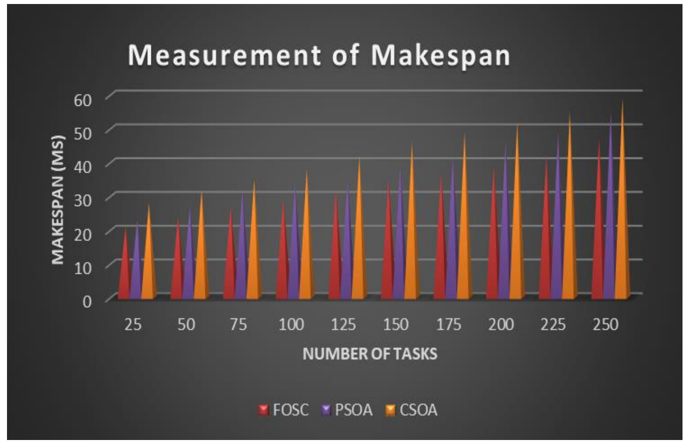

Figure 6. Performance result for Makespan

The Figure 5 shows the performance result of scheduling efficiency. The below graph clearly states that the performance of FOSC is more compared to PSOA and CSOA.

Makespan is defined as the maximum time taken to complete all received jobs per time.

$$Makespan = max\{CTJ \mid \forall J \in JQ \quad (5)$$

In the above equation (5), CT_J represents the Task Completion time for Job in Job queue JQ.

Table.3 Tabulation for makespan

No. of tasks	Makespan (ms)		
	FOSC	PSOA	CSOA
25	21	23	28
50	24	27	32
75	27	32	35
100	29	34	38
125	32	35	42
150	35	39	46
175	37	42	49
200	39	46	52
225	42	49	55
250	47	55	59

Table 3 shows the performance result of makespan for varying number of tasks. If the number of tasks is 250, the makespan for FOSC is 47 ms which is less when compared to PSOA which takes 55 ms and CSOA that takes 59 ms.

The Figure 6 shows the performance result of makespan. The above graph clearly states that the makespan time of FOSC is less compared to PSOA and CSOA

Resource utilization rate is defined as number of resource used to total number of resources available. The major resources that are taken into consideration are memory and bandwidth. The main decision on selecting the better algorithm depends on the factor of resource utilization. The proposed method should not underutilize or over utilize the resources .Resources should be used effectively while handling the task submission and execution.

Table 4 Tabulation for Resource Utilization

No. of tasks	RESOURCE UTILIZATION (%)		
	FOSC	PSOA	CSOA
25	80	71	62
50	83	74	65
75	87	78	69
100	88	81	72
125	90	83	75
150	91	85	77
175	93	89	81
200	95	91	84
225	96	93	86
250	98	94	89

Table 4 describes the resource utilization of FOSC as compared with PSOA and CSOA .If we consider the number of tasks as 150, the resource usage percentage of FOSC is 91 percentage which is higher when compared with PSOA having 85 percentage and CSOA having 77 percentage of resource usage

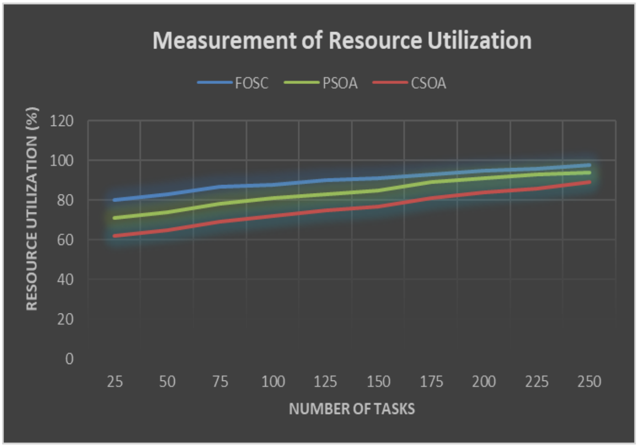

Figure 7. Performance result for Resource Utilization

The Figure 7 graphically depicts Resource Utilization Rate of the three methods taken into consideration. The above graph clearly states that the Resource Utilization Rate of FOSC is increased as compared to PSOA and CSOA

6. CONCLUSION

An efficient technique called, firefly optimized Scheduling (FOSC) is developed for improving the task scheduling efficiency and lessening the makespan of the cloud server. The scheduling is carried out based on the task classification. The firefly algorithm is used for finding the optimal virtual machine. From the analysis, the tasks are prioritized and are stored in different queues. Then the cloud manager finds the resource optimized virtual machine to process the high priority tasks. This helps to minimize delay while handling a large number of tasks. Experimental evaluation of FOSC technique and existing methods are carried out using Personal Cloud Dataset. The experimental result shows that the FOSC technique improves the task scheduling efficiency with minimum time and efficiently utilize the resources

REFERENCES

[1] Arnav Wadhonkar, Deepti Theng "A Task Scheduling Algorithm Based on Task Length and Deadline in Cloud Computing, International Journal of Scientific & Engineering Research, Volume 7, Issue 4, April-2016, ISSN 2229-5518.

.

[2] Fakhrosadat Fanian, Vahid Khatibi Bardsiri and Mohammad Shokouhifar, "A New Task Scheduling Algorithm using Firefly and Simulated Annealing Algorithms in Cloud Computing" International Journal of Advanced Computer Science and Applications, volume 9, issue 2, 2018.

[3] Elmougy, S., Sarhan, S. & Joundy, M. "A novel hybrid of Shortest job first and round Robin with dynamic variable quantum time task scheduling technique." Journal of Cloud Computing, volume 6, issue 12, 2017

[4] Mahendra Bhatu Gawali and Subhash K. Shinde "Task scheduling and resource allocation in Cloud computing using a heuristic approach", Journal of Cloud Computing Advances, Systems and Applications (2018), volume 7, issue 4.

[5] G. Natesan and A. Chokkalingam, "Task scheduling in heterogeneous cloud environment using mean grey wolf optimization algorithm," ICT Express, vol. 5, no. 2, pp. 110–114, 2019

[6] S. Loganathan and S. Mukherjee, "Job Scheduling with Efficient Resource Monitoring in Cloud Datacenter," Sci. World J., vol. 2015, no. ii, 2015

[7] K. C. Babu, A. S. Mahesh, A. V. Vidyapeetham, and A. V. Vidyapeetham, "Comparative Analysis of Deadline Constrained Task Scheduling Algorithms for Cloud Computing under Cloudsim," International Journal of Pure and Applied Mathematics, vol. 119, no. 10, pp. 907–920, 2018.

[8] K. Liu, R. Zheng, M. Zhang, C. Han, J. Zhu, and Q. Wu, "SRAF: A service-aware resource allocation framework for VM management in mobile data networks," Mobile Information Systems., vol. 2018, 2018

[9] L. M.Mustafa, M. K. Elmahy, and M. H. Haggag, "Improve Scheduling Task based Task Grouping in Cloud Computing System," International Journal of Computer Applications, volume. 93, issue 8, pp. 1–5, 2014

[10] C.R. Durga Devi, R.Manicka Chezian "Resource Optimized Ensemble Gradient Boosting Classifier for Traffic Aware Big Data Analytics," Journal of Advanced Research in Dynamical and Control System., volume. 10, issue. 03, pp. 839–852, 2018.

[11] H. Choi, J. Lim, H. Yu, and E. Lee, "Task classification based energy- aware consolidation in clouds," Scientific Programming, volume. 2016,

[12] H. Gamal El Din Hassan Ali, I. A. Saroit, and A. M. Kotb, "Grouped tasks scheduling algorithm based on QoS in cloud computing network," Egyptian Informatics Journal. vol. 18, no. 1, pp. 11–19, 2017,

[13] N. Er-raji and F. Benabbou, "Priority Task Scheduling Strategy for Heterogeneous Multi-Datacenters in Cloud Computing," International Journal of Advanced Computer Science and Applications, Volume 8, Issue 2, pp. 272–277, 2017

[14] C. R. Durga Devi and R. Manicka Chezian, "Multivariate logistic regression based gradient descent firefly optimized round robin scheduling with big data," Journal of Advanced Research in Dynamical and Control System, volume. 11, issue. 1, pp. 179–192, 2019.

AUTHORS

First Author – C. R. Durga devi, MCA.,M.Phil., (NET), Nallamuthu Gounder Mahalingam College,

Second Author – DR. R. Manicka chezian, M.Sc., M.S., Ph.D., Nallamuthu Gounder Mahalingam College,

Correspondence Author – C. R. Durga devi,