International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
Article Received: *25 July 2023* ***Revised:*** *12 September 2023* ***Accepted:*** *30 October 2023*

Develop Prototypes and Scheduling Strategies for Cloud Computing

Jayant D. Sawarkar
Research Scholar
Department of Computer Science & Engineering
Dr. A.P.J. Abdul Kalam University
Indore, India
jdsawarkar@yahoo.com

Dr. Manoj Eknath Patil
Research Guide
Department of Computer Science & Engineering
Dr. A.P.J. Abdul Kalam University
Indore, India
mepatil@gmail.com

Abstract— Cloud computing usage is rapidly increasing and turning into a driving force among all modern-day industries offer it to customers worldwide. It gave a notable deal of thought to the enormous amount of energy consumed in the datacenters. Consequently, it’s far essential to decrease the loss of energy efficiency, meet battery life, meet performance requirements, decrease electricity consumption, maximize earnings, and to use resources most efficiently. Utilizing a scheduling approach to find the ideal undertaking execution sequence based on an individual’s need and with the least amount of execution time and cloud resources is the most effective way to decrease energy. The paper aims to advocate a brand-new method for cloud computing to reduce energy consumption in the environment and adhere to Service Level Agreement (SLA) and Quality of Service (QoS) norms. Utilizing energy- and energy-scheduling algorithms will assist in enhancing and limiting the procedure for mapping within incoming tasks and datacenter servers and acquiring the best use of recourses from the data centre to obtain superior computing overall performance, reducing the demand on the network and the datacenters' energy use. In order to maximise efficiency due to bandwidth usage and conserve energy used in the datacenter, Process time and system total make span should be kept to a minimum. This paper investigates energy-conscious cloud computing datacenter designs with a variety of scheduling methods and suggests a novel task scheduling method with based on file location during live migration proceeding. The evaluation and proposal of a new SLM algorithm in numerous situations in the usage of CloudSim toolkit, then outcome indicates a vastly increased energy efficiency reading levels and total make span related to the particle swarm optimization technique (PSO) and the ant colony method (ACO) demonstrates an important development in the statistics for energy use, Degrees, and Overall Makespan. The overall amount of time needed to perform all jobs is known as the span of time.

Key word: Cloud computing, SLM, Scheduling, Virtualization.

I. INTRODUCTION

The massive demand for cloud computing has resulted in the requirement for carefully controlling the resources that offer services to users. Cloud-based data centres use a significant amount of energy, which makes them an expensive system and necessitates careful resource planning. Resource planning deals with this idea [1], it refers to the process of appropriately generating the schedule and deciding which specific activities will be assigned to which resources. The energy issue can be overcome thanks to the virtualization technique.

A scheduler must schedule the same number of virtual machines that the consumer requests. Scheduling software schedules a number of virtual machines to meet demands including ensuring optimum energy efficiency, achieving a high degree of load balancing, and minimising resource use. Large-scale Cloud computing is becoming more and more common, the environment's ability to integrate needed services into energy consumption has emerged as a crucial research issue, however traditional scheduling methods are ineffective for energy-efficient planning. It is acknowledged that an application programme must optimise the transfer and processing times while scheduling some extensive data or

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 October 2023*

computing an extensive application. Effective task allocation strategies are essential to optimising a number of performance parameters in the cloud system.

There aren't many common task scheduling algorithms used in cloud environments because of how new the field of cloud computing is. Particularly in the cloud, using well-known job schedulers in large-scale distributed settings is impossible due to their high communication costs [2]. Routine-based scheduling methods are widely used in cloud computing systems. It is straightforward, simple to implement, and improves the effectiveness of the methods. The existing detection methods are either based on diversity detection or improvised detection. Researchers are now striving to create task scheduling algorithms that can be used in cloud computing settings and are compatible with them. The most crucial duty in cloud computing is job scheduling. Due to the fact that users must pay for resources they use on a time-based basis. Utilizing resources efficiently must be a priority, and scheduling is crucial to ensuring that you get the most out of them. The current scheduling methods take a lot of time because they perform several checks.

The development of virtualization and Internet technologies has led to the emergence of cloud computing as a new computer platform. A distributed system called cloud computing consists of a number of virtualized, networked machines that are dynamically deployed. On the basis of service-level agreements (SLAs) between service users and service providers, it offers one or more integrated computer resources. There are some difficulties with cloud computing, such as reliability, performance, resource management, and security. Task scheduling is one of the challenges with resource management. In cloud computing, task scheduling is the process of allocating users' tasks to the resources that are available in order to optimise task execution and maximise resource usage. The task execution cost is regarded as one of the performance criteria of the task scheduling algorithm because the distribution of Cloud resources is based on SLA. However, because it must fit a lot of jobs into the given resources, the task scheduling algorithm is regarded as a complex process. On the other hand, when creating a task scheduling algorithm, numerous factors need to be taken into account. Some of these factors, such as task compilation time, cost, and response time, are significant from the viewpoint of a cloud user [3]. From the standpoint of the cloud service provider, additional factors are crucial, such as resource utilisation, fault tolerance, and power usage. It is believed that the job scheduling problem is NP-Complete. Therefore, by taking into account performance criteria (such as completion time, cost, resource utilisation, etc.), optimisation methodologies could be utilised to address it. In order to increase job completion times, reduce execution costs, and maximise resource utilization [4], this study seeks to design a task scheduling algorithm for the Cloud computing environment based on a Genetic Algorithm.

1.1 ASSESSMENT OF PURPOSE AND PROBLEM

Hardware expertise isn't necessarily the most important factor influencing the energy usage in data centres; instead, the resource management system in use has a significant and powerful influence on comprehensive accomplishment. Cloud service providers have several worries, namely host consolidation, power saving policies, scalability resources, optimum task scheduling for saving energy, and so forth. Highlight their importance and the part they play in creating a productive cloud environment, but this is challenging and poorly exposed by using researchers. Because the scale of data centres that span miles of ground and contain a large number of computers and resources are growing larger every year, energy consumption stage is the primary worry in the cloud environment [5]. To create the required energy-efficient cloud environment, energy optimization task scheduling needs to be given the utmost attention and further research.

General Objective of the Study

- The overall goal of this research is to create prototypes, as well as to research and analyse the scheduling strategy for cloud computing for virtual machines in private cloud environments.
- To research and outline each significant resource scheduling strategy aimed at maximising user Quality of Service (QoS) metrics.
- Examine the available options for reducing the amount of energy used by cloud datacenters in terms of power model creation, provisioning, and execution.

1.2 AN ENERGY EFFICIENT JOB SCHEDULING TAXONOMY

Datacenters normally operate 365 days a year, with an average power density of 10 KW/m2, and this number is increasing by 12% every year. By starting to manage the power of one or more of these components, it is possible to minimise the amount of energy used in data centers in the cloud. The network, storage, server, and system for cooling all require electricity in cloud computing datacenters [6].

II. CONCEPTUAL SCHEDULING

Numerous scheduling models, such as the one shown in the Figure 1, was put forth and thoroughly evaluated, however the majority of them fall short of addressing the weaknesses of today's most recent cloud environments, or they only partially address one drawback while ignoring another. This highlights the need for new and improved scheduling models

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 10
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 October 2023*

that are both more effective and creative.

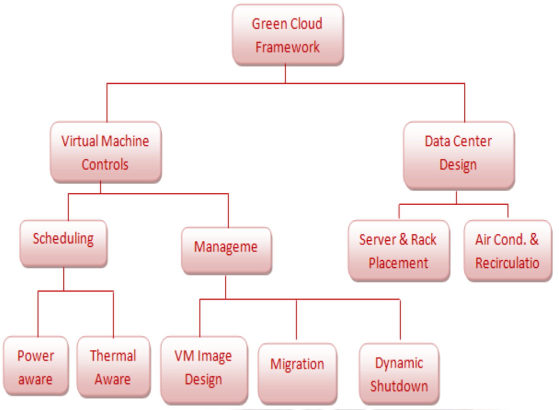

Figure 1. Models for green cloud scheduling

With a focus on all computer resources, scheduling is a broad concept and essential component of cloud computing that significantly affects the quality of cloud services. To reduce the time required to accomplish tasks, a scheduler may fairly categorise and analyse computing resources [7]. The performance of the server and the effectiveness of its associated resources can both be improved in a cloud environment by improving the scheduling mechanism's effectiveness. [8] [9].

By utilizing one or more methodologies, job scheduling aims to efficiently and effectively match user demands with the most suitable resources [10]. Scheduling algorithms can manage the cloud's data center performance, total energy usage, and processing time in general [11]. A task scheduler with low energy consumption deployment increases the number of tasks approved with the CSP. It lowers the frequency of VM server migrations, operating cost rates, and ensures dependable resource delivery, all of which contribute to maintaining the SLA [12]. The major cloud variables taken into account during the green computing scheduling phase are power and thermal awareness. In contrast to the original model, reduce energy consumption; the next model has a tendency to boost energy levels inside the data centre while lowering temperatures in the hardware structure.

2.1 ASSESSMENT FACTORS FOR ENERGY EFFICIENT SCHEDULING

Implementing a cloud-based job scheduler datacenter ought to ultimately assist in achieving that objective since the primary goal of the cloud is offer a quality of service (QoS) to the user. Based on a number of characteristics and objective features it may be applied and guided to improve the effectiveness and quality of work scheduling in the datacenter. The effectiveness of the work schedule is assessed and prioritised in comparison to other scheduling methodologies used in the datacenter. The measurements for selecting these objectives are determined by the client and cloud service provider.

2.2 PROBLEM STATEMENT

Due to heavy workloads and low resource availability, existing cloud servers experience overload and high consumption, which lowers cloud performance and raises energy levels. In contrast, the majority of scheduling methods employed by businesses have a limited application and their implementation does not meet the demand for a greener environment. Incoming cloudlets are placed using the diagram flow shown in Figure 2 [13], which begins with the task being submitted to the data centre and continues whenever the cloudlets have finished processing. Every new job entered into the machine is assigned dynamically to the initial host and given a fresh window of time to be processed. When jobs arrive, a fresh placement allocation and time slot are reserved in the next directly available VM. This waste of resources and storage lengthens the system's overall makespan. Then, regardless of job ranking, jobs are completed one after the other. Due to the fact that several tasks of various sizes and weights live in one location as well as share resource and memory concurrently in IaaS, there is a need to create theoretical task management and scheduling tactics. In contrast, previous cloud-deployed strategies were unable to handle a variety of job types and prioritise incoming workloads.

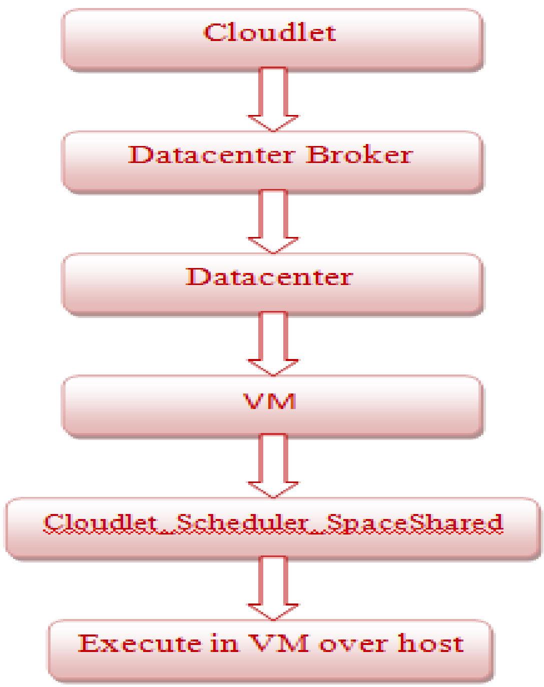

Figure 2. Flowchart for executing a job

III. ALGORITHM OF THE SLM SCHEDULER

The suggested Sharing with Live Migration scheduling technique contains four primary phases that reduce overall energy consumption: selection and placement of virtual machines, processing of many jobs at once, active live migration, and shutdown of idle hosts. The workflow employed is random, all jobs are autonomous, and user requests are not prioritised.
Step 1: Process user requests for different jobs in this step. The four task categories include reading file contents, uploading files, downloading software and updating data. Each task may fall into one of these categories. The quantity of instructions for each task and what kind of work was chosen affect the processing time.
Step 2: The queue manager collaborates with the SLM scheduler connected to the SCJ table as it processes a number of user requests.
Step 3: Resource allocation in this stage is based on the SCJ ID of each host and its accompanying resource file, which lowers the transmission load and impairs network connectivity. The previous step lowered the waiting time for the jobs by classifying them in various queues.
Step 4: Jobs in the WU queue will be handled in order. On the other hand, a selective approach utilizing the RD queue will be used to allocate jobs with identical assets with homogenous requests for processing for the same VM, and then there will be use of space-shared scheduling to manage several jobs concurrently. The number of concurrent jobs that can be processed depends on how many free PEs were previously allocated to the VM.
Step 5: When each operation is finished, the processing time is utilised to determine the host's energy consumption, and the output of the algorithm displays the energy consumption of each host.
Step 6: The energy consumption of the datacenter will be calculated using the total energy utilised by hosts running all workloads at the data centre during a specified time period.

3.1 *THE EXECUTION MODEL*

In order to create a framework that makes sense, each piece of the simulation model that is important for evaluation was explicitly described.

- Servers
- DataCentre
- Cloudlets
- Hosts
- Resources
- Backup servers
- Processing type
- Power consumption
- Processing time

The proposed SLM scheduler algorithm execution phases:

1. Users send several jobs to the data center.
2. In the event that the required server is not fully utilised, the scheduler starts processing jobs.
3. *Incoming tasks are placed in the queue if the necessary file is already in existence.*
4. *Each job in the queue, the scheduler runs a job check-up procedure, in addition, if one of the tasks calls for the resource to be used in a similar manner, they are allowed to begin the procedure right away using the prior job with no waiting in the selected host isn't overloading.*
5. *If the required main server is overloaded, the job should be moved to the duplicate servers that aren't being used.*
6. *When the under loaded replica server is no longer overrun, move the position to the main server and turn off the replica.*
7. *If there are no resources available, uses queuing for operations that need little waiting time.*

IV. *PROPOSED SLM SCHEDULERS ANALYSIS*

The results and assessment of the suggested SLM scheduler make use of energy usage levels and makespan to be able to demonstrate the performance of the suggested SLM to be tested using particle swarm optimisation (PSO) and ant colony optimisation (ACO), as both of the algorithms produced the most effective outcomes in optimising energy usage. Because PSOs may quickly identify ACO does not require a job scheduler for assigning resources under time constraints as it is the optimal option with reduced computing costs [14][15]. After integrating VM consolidation and the job scheduling algorithm, the SLM numerical evaluation is utilised to decide the level of energy utilization created by the cloud server.

4.1 MATRIX OF PERFORMANCE EVALUATION

500 cloudlets were used as the dynamic workload in a number of trials, and the results were used to assess how well the SLM scheduler model performed. The suggested SLM scheduler's performance test results were compared to those of in order to further validate and demonstrate its usefulness, ACO and PSO scheduling approaches are used in four essential Energy consumption levels, migration size, server utilisation, and makespan are components of the performance matrix.

4.2 UTILIZATION OF SERVERS

The efficiency of the SLM scheduler was assessed using six hosts and 500 cloudlets, and the highest use of each server is thought to be one of the most effective ways to optimise energy usage. In contrast to ACO and PSO, where server one to six utilisation varied between 57% and 78% according to Figure 3 and Table 1 data, the SLM boosted host utilisation. ACO and PSO maximum usage, however, varied between 11% and 89% and 25% and 75%, respectively, indicating a substantial proportion of underutilised hosts inadequate resource use within the system.

Table1. Use of hosts in SLM, ACO, and PSO

Host Number	Utilization of Servers			
	Traditional Technique	SLM	ACO	PSO
1	57%	77%	52%	48%
2	49%	58%	33%	27%
3	61%	79%	37%	52%
4	56%	68%	12%	42%
5	56%	78%	68%	71%
6	45%	68%	87%	32%

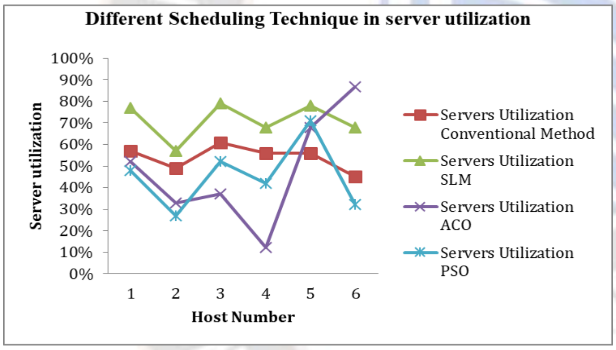

Figure 3. Comparison of server utilization in SLM, Conventional, ACO, and PSO

4.3 NUMBER OF LIVE MIGRATIONS PERFORMED

The different number of VM migrations acquired utilising various strategies is displayed in Table 2. The migration intervals and time length increase as the dynamic workload increases, and the number of migration processes decreases, showing that the high load detection method was successful in decreasing migration iteration. Hence, more migrations were carried out in the beginning of the workload arrival from 100 to 300 than when the workload increased from 400 to 500.

In terms of migration reduction, Figure 4 demonstrates that the suggested SLM scheduler performs better than the conventional, PSO, and ACO schedulers. As stated in comparing migration Table 2 to PSO, the number has dropped by 11.3%while migrations have been minimised by roughly 21% when compared to the typical amount of migrations of the ACO method.

Table 2. Number of migrations for each scheduler

No. of Cloudlet	Number of VMs Migration			
	Conventional Method	SL M	ACO	PSO
100	8	3	5	6
200	12	8	12	9
300	14	12	15	13
400	20	17	19	18
500	24	19	24	21

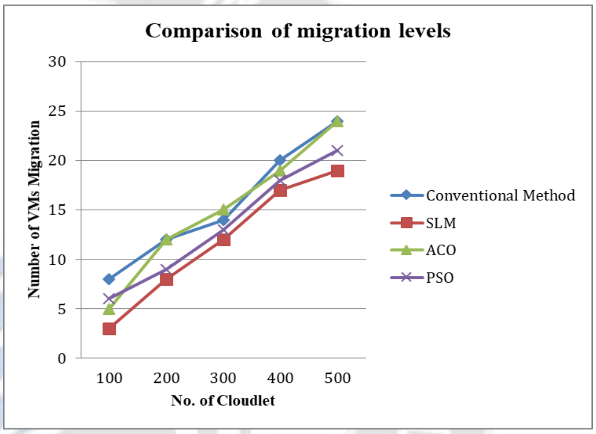

Figure 4. Comparison of migration levels

4.4 EFFECT ON ENERGY USE

When compared to the traditional approaches, PSO and ACO, the results of employing the SLM model demonstrate gains in energy consumption reduction. As an example, a set of 100 cloudlets needed 0.07 kWh in the cloud datacenter to process using the SLM algorithm, compared to 0.14 and 0.11 in ACO and PSO, similarly. The ability of this approach to schedule energy optimization has been demonstrated in Table 3 and it performed well in numerous simulated studies. With different quantities of cloudlets, the SLM clearly distinguishes between Energy Consumption categories. Compared to ACO and PSO, the traditional procedures, SLM reduces energy usage by 35.11 percent, 23 percent, and 18.1 percent, respectively.

Table 3. SLM, ACO, and PSO energy consumption with varying number of cloudlets

Total Cloudlets	Energy Utilization (kwh)			
	Traditional Method	SLM	ACO	PSO
100	0.15	0.06	0.15	0.12
150	0.18	0.09	0.16	0.13
200	0.23	0.12	0.19	0.18

250	0.29	0.14	0.22	0.18
300	0.31	0.16	0.25	0.21
350	0.31	0.18	0.26	0.22
400	0.33	0.21	0.27	0.27
450	0.37	0.23	0.30	0.28
500	0.40	0.25	0.34	0.30

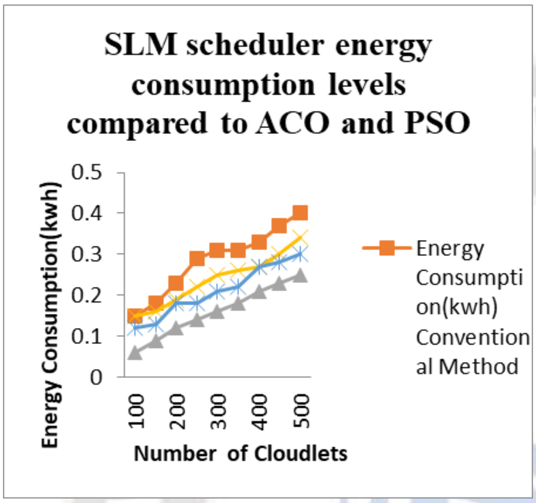

Figure 5. SLM scheduler energy consumption levels compared to ACO and PSO

4.5 MAKESPAN

Comparing to the schedulers for ACO and PSO, the SLM technique maintained continuous make span control and optimized, the SLM technique was able to maintain continuous make span control and optimise job execution time when the number of jobs increased from 100 to 500. The SLM execution strategy reduced the processing time needed for synchronised operations by running many homogenous processes concurrently, whereas other schedulers did not, so the SLM makespan is shorter than other scheduling methods.

Table 4. Makespan with different numbers of Cloudlets

Total number of cloudlets	Make Span Time			
	Traditional Technique	SLM	ACO	PSO
100	09.05	06.72	8.71	07.95
200	14.40	10.03	13.21	11.92
300	17.50	13.11	15.15	14.15
400	19.42	15.42	17.25	17.17
500	22.06	18.23	23.09	19.12

Figure 6. Improvement in the proposed SLM's Makespan versus PSO and ACO

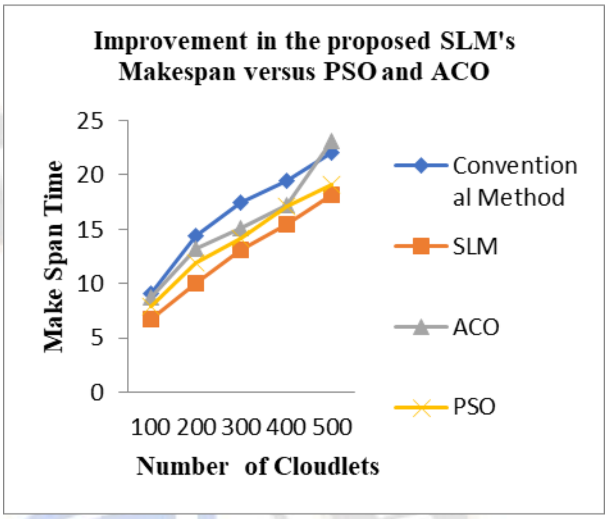

Figure 6 depicts the time span of various scheduling algorithms, with the proposed method SLM outperforming another approach. SLM outperforms traditional methodology, ACO, and PSO in terms of overall make span by around 18%, 14%, and 09%, respectively.

V. CONCLUSIONS

Using forward cloud computing optimisation methods and effective cloud scheduling methods, this article suggests a dynamic workload scheduling approach that is energy-efficient. In this paper, sharing with live migration was introduced with a scheduler, and the suggested method was assessed against a number of industry benchmarks. According to the experimental findings, concurrent work allocation, execution, VM consolidation, and migration based on dynamic threshold margins all lead to significant energy savings and makespan optimization. Comparing it to the ACO and PSO scheduling algorithms, the results revealed a minimization of 15% and 10% in the system makespan, respectively. Power consumption was reduced by 23% and 18.1%, respectively, as in comparison to ACO and PSO, as a result of the adoption of the SLM scheduler. The six servers, as opposed to the other two scheduler servers, had the highest utilisation rate. This model proved effective in enabling a scheduling method and cloudlets deployment service that enhances resource efficiency and increases the system's capacity to reduce energy consumption while achieving the performance target. As the overall cost of cloud computing is directly related to energy consumption, this system will operate at a lower total cost. Reduced energy usage is more environmentally beneficial because it produces fewer CO2 emissions.

REFERENCES

[1] M. A. Rodriguez, & R. Buyya, "Deadline based resource provisioning and scheduling algorithm for scientific workflows on clouds." IEEE transactions on cloud computing, 2(2), 222-235, 2014

[2] P. Kowsik, & K. Rajakumari, "A Comparative Study on Various Scheduling Algorithms in Cloud Environment." International Journal of Innovative Research in omputer and Communication Engineering, 2(11), 6494-6500, 2014

[3] S. Sindhu, S. Mukherjee, "Efficient Task Scheduling Algorithms for Cloud Computing context." In International Conference on High Performance Architecture and Grid Computing (HPAGC-2011), vol 169, pp 79-83, 2011

[4] J. W. Ge and Y. S. Yuan, "Research of cloud computing task scheduling algorithm based on improved genetic algorithm," in Applied Mechanics and Materials, pp. 2426-2429, 2013

[5] D. Armstrong, K. Djemame, R. Kavanagh, "Towards energy aware cloudcomputingapplicationconstruction"JournalofCloudComputingAdvances,Systems and Applications,Vol.6,pp.1-13,2017.

[6] M. Avgerinou, P. Bertoldi, L. Castellazzi, "Trends in Data Centre Energy Consumption under the European Code of Conduct for Data Centre Energy Efficiency," Energies MDPI, Vol.10, 2017.

[7] T. Ma, M. Tang, W. Shen, Y. Jin, "Improved FIFO Scheduling Algorithm Based on Fuzzy Clustering in Cloud Computing," Information2017,Vol.5,pp.25,2015.

[8] S. Sharma, N. Kumar, "Task Scheduling Algorithm in Cloud Computing based on Power Factor," International Journal of Innovative Research in Science, Engineering and Technology, Vol. 6, 2017.

[9] A. K. Yadav, S. B. Rathod, "Study of Scheduling Techniques in Cloud Computing Environment," International Journal of Computing Trends Technology, Vol.29, pp.69–73, 2015.

[10] A. I. Awad, H. M. Abdelkader, "Enhanced Particle Swarm Optimization for Task Scheduling in Cloud Computing Environments," Proceeding of Computer Sciences, Vol.65, pp.920–929, 2015.

[11] B. H. Malik, M. Amir, B. Mazhar, S. Ali, R. Jalil, J. Khalid," Comparison of Task Scheduling Algorithms in Cloud Environment," International Journal of Advanced Computer Science and Applications, Vol.9, No.5, 2018.

[12] B. P. Mishra, H. Das, S. Dehuri and A. K. Jagadev, " Cloud Computing for Optimization: Foundations, Applications, and Challenges," Springer,Vol.39,pp.51-63,2018.

[13] Z. Cai, X. Li, R. Ruiz, Q. Li, "A delay-based dynamic scheduling algorithm for bag-of-task workflows with stochastic task execution times in clouds" Future Generation Computer Systems, Vol.71, pp.57-72, June2017.

[14] D. Wang, D. Tan, L. Liu, "Particle swarm optimization algorithm: an overview," Springer, Vol.22, pp.387-408, 2018.

[15] K. Anjaria, A. Mishra, "Thread scheduling using ant colony optimization: An intelligent scheduling approach towards minimal information leakage," Karbala International Journal of Modern Science, Vol.3, pp.241-258, 2017.