Journal of Neonatal Surgery
ISSN(Online): 2226-0439
Vol. 14, Issue 20s (2025)
https://www.jneonatalsurg.com

OPEN ACCESS

Optimizing Task Scheduling in Cloud Computing: A Hybrid Approach Integrating Bandwidth Awareness and Bar System Model for Enhanced Efficiency

NageswaraPrasadhu Marri[1], Dr. N R Rajalakshmi[2]

[1]Research Scholar, Dept of CSE, VelTechRangarajanDr.Sagunthala R &D Institute of Science and Technology, Chennai, India,

Email ID: vtd596@veltech.edu.in

[2]Professor, Dept of CSE, VelTechRangarajanDr.Sagunthala R &D Institute of Science and Technology, Chennai, India, Email ID: drnrrajalakshmi@veltech.edu.in

Cite this paper as: NageswaraPrasadhu Marri, Dr. N R Rajalakshmi, (2025) Optimizing Task Scheduling in Cloud Computing: A Hybrid Approach Integrating Bandwidth Awareness and Bar System Model for Enhanced Efficiency. *Journal of Neonatal Surgery*, 14 (20s), 30-38.

ABSTRACT

The emergence of the cloud computing era has led to the development of new technologies and the increasing complexity of task scheduling in the cloud. One of the biggest challenges in the management of this platform is the allocation of the appropriate resources to the tasks. In order to improve the performance of the cloud computing environment, various algorithms have been proposed.Existing algorithms suffer from their own issues. In this paper, we present a hybrid model that takes into account the various factors that affect the task's prioritization. The suggested method is based on the Bandwidth-Aware Task Scheduling model with the addition of the Bar system model.The proposed algorithm employs a minimum lease policy and an overload pre-emptively within a data center to address the issue of overrunning virtual machines. Different tests are performed on the hybrid model to evaluate its efficiency. The results of these tests proved the model's effectiveness.

Keywords: *Cloud Computing, Task scheduling, Optimization, Enhanced Efficiency*.

1. INTRODUCTION

Cloud computing has become an integral part in the modern world. It has major technical advancements in the field of information technology [1]. It allows organizations to manage their various tasks and resources efficiently. In addition to being able to provide a variety of services, cloud computing also offers security and storage. Cloud computing provides a variety of services to its users. These services are offered through the use of virtual resources, which are dynamic and elastic. They allow users to schedule their tasks and applications over the cloud. Although numerous researchers [2-5] have made significant progress in solving the scheduling problem associated with cloud computing, it is still considered as the NP-hard issue. The cloud environment is composed of various data centers that are distributed across the country. Each of these data centers houses thousands of servers [6-7]. The cloud users are provided with a group of virtual machines to carry out the tasks. Unfortunately, when it comes to scheduling the resources for the tasks, it is not easy to find the right solution. The complexity of the task assignment process can be solved by taking into account the various attributes of the task, such as its size, duration, and dependencies. When it comes to scheduling the virtual machines for the tasks, it can be very complex. Before we assign the resources to the task, it is important to consider the various properties of the task. For instance, the length of the task, its size, and its dependencies are some of the factors that need to be considered before we assign the resources to the task [8]. Although there is a lot of progress that has been made in the field of task scheduling, it is still considered an NP-hard issue. The goal of the task scheduling process is to reduce the time and cost associated with the execution of the task. It also decides which virtual machine should be used to carry out the task [9-12]. When it comes to handling the different characteristics of virtual machines in a cloud environment, it's important that the VM's attributes are taken into account. A task scheduler can also take into account the overall performance of the machines to balance the load.

The paper presents a framework for implementing the Hybrid scheduling algorithm, which takes advantage of the combination of the BAR optimization technique and the BATS optimization method. Section 2 explores the latest studies about cloud computing's provisioning and resource scheduling. Section 3 presents the hybrid model, which explains the various changes made to the cloud computing framework. Section 4 covers the analysis of the results, and the Section 5 wraps up the study.

2. LITERATURE SURVEY

The section discusses the latest research on the scheduling of resources in cloud computing. There are numerous solutions being developed to address this issue. The algorithms suggested in these studies are based on meta-heuristic and heuristic approaches.The use of heuristic methods is used to identify the most effective solution for a given problem. These methods are usually generated by taking into account the optimization rules and the size of the problem. In contrast, the meta-heuristic approach [13] takes into account the group of solutions to select the most efficient one.

Some of the most popular meta-heuristic methods used in the development of cloud computing solutions are the Ant colony optimization [14], particle swarm optimization [15], and genetic algorithm [16]. These are combined with other factors such as the differential equation and the Honey Bee colony to perform better [17].

Although the authors of the load balancing algorithm [18-19]used in the cloud were aware of the task sizes, they decided not to consider them when developing the method. Instead, they focused on the servers' refresh time. The proposed vacation queuing method faced several disadvantages, such as the algorithm's time-intensive nature.

The authors of the study [20] considered the bandwidth parameter when it comes to resource scheduling in the cloud. They then developed a non-linear model for the process of resource scheduling. In [21], the authors proposed a task scheduling algorithm called as rolling horizon algorithm. The major drawback of this algorithm has high execution time.

The energy consumption [22] and the task length of the resource were taken into account when developing the algorithm. The authors [23] then developed a parallel model that takes into account the FCFS algorithm for scheduling jobs.

3. PROPOSED ARCHITECTURE FOR TASK SCHEDULING

The concept of task scheduling is shown in Figure 1. The various types of tasks that are sent to the cloud environment are categorized into various sizes. The proposed model takes into account the hierarchy process to perform the tasks at the data center. The most important goal of the model is to assign them to the VMs. The proposed method takes into consideration the runtime and length of the tasks and then decides which ones should be prioritized. It can be managed through the queue. The algorithm for task schedulers is based on the properties of the data center.

3.1 Decision Optimization Using Hierarchy Process

Decision optimization is performed using the hierarchy process that are made in the cloud environment. The proposed algorithm for task prioritization takes into account the various criteria and the hierarchy process to perform tasks efficiently.

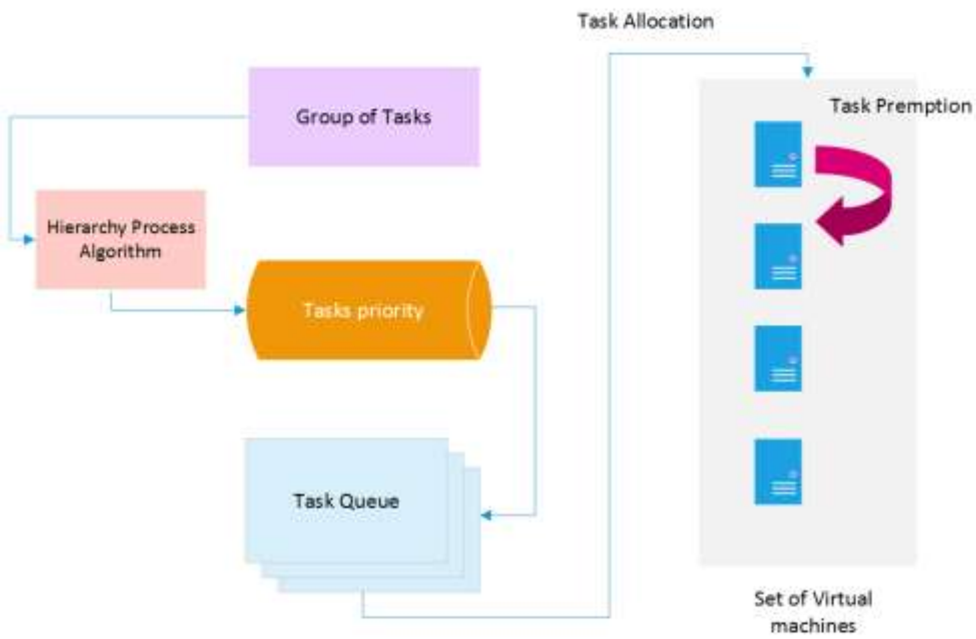

Figure 1: Proposed Architecture for Task Allocation

Algorithm 1: Hierarchy Process [24]

Input: Set of tasks T -> $\{T_1, T_2 \ldots T_n\}$

Output: Task Prioritization P_T

Begin

1. Submit the input T to the cloud
2. Consider the parameters {size and runtime } of each task T_n
3. Assign the priority for each task to form a priority table T_t
4. Add each value in the column $X=(X_1, X_2 \ldots X_n)$
5. By using X_T, divide the each element in the matrix
6. As next step, convert the each element value in to decimal
7. Calculate weighted sum S_w
8. Calculate the distribution time β_{max} using Eq.1.

$$\beta = \frac{sum\, of\ the\ prioeity values}{n} \quad (1)$$

9. Calculate the consistency index I from Eq. 2

$$I = \frac{\beta_{\max} - n}{n-1} \quad (2)$$

10. Rearrange the task based on the priorities

End

The initial stage of the process involves creating a workflow that is composed of five phases. Before the algorithm is implemented, the workflow is analyzed and the parameters of the task are set. The priority table for the tasks is maintained in the server. The suggested algorithm takes into account the length and execution time of the tasks.

3.2 Proposed Hybrid Model

Hybrid optimization has two modules: resource management and task scheduling. The first of these is the task scheduling, while the other is the allocation of resources. The BAT model [16] takes into account the tasks that are having independent and equal sizes, and it doesn't take into account the load on virtual machines.In some models, the tasks will wait until the virtual machines are occupied before they can execute. In the other model, the tasks will be executed once the VMs are released. The concept of the Bar system is inspired by the behavior of a staff of bar workers serving drinks to customers.The goal of the Bar system is to provide a flexible and fast environment that allows the bartenders to perform their duties. This model is very different from the other models in that it allows the agents to perform the tasks simultaneously. One of the main disadvantages of the system is that it requires agents to meet certain conditions in order to maintain their status. This issue can be solved by implementing a hybrid model, which combines the Bar system with Bandwidth-Aware scheduling. The complete procedure can be found in algorithm 2.

1. The data collected by the system is then aggregated according to the priority of the tasks.
2. The system can also collect various data points about the virtual machines, such as the task execution time and initial load.
3. The system can then generate a bipartite graph that will be used to assign the tasks to the various virtual machines.

The virtual machine $V_i$bandwidth is given in Eq. 3,

$$BW(V_i) = \frac{Used\ BW(V_i)}{Total\ BW(V_i)} * 100 \quad (3)$$

The virtual machine $V_i$initial load is given in Eq. 4,

$$iload(V_i) = \frac{Used\{BW(V_i), mem(V_i), CPU(V_i)\}_i}{Total\{BW(V_i), mem(V_i), CPU(V_i)\}} * 100 \quad (4)$$

The task T_j execution time is given in Eq. 5

$$Exec(T_j, V_i) = T_j\ (length, size) * Avail\{BW(V_i), mem(V_i), CPU(V_i)\}_i \quad (5)$$

Algorithm 2: Proposed Hybrid Optimization Model

Input: Number of Tasks T_n

Number of VMs V_m

Ouput: Task scheduling

Begin

1. Tasks T_n is given as input to the scheduler
2. Find the tasks information such as size and length
3. Find the virtual machine data $BW(V_i)$, $iload(V_i)$ and $Exec(V_j)$
4. Develop the bipartite graph, G={V,E} where V is the vertex which belongs to T_nUV and E is the edge(i.e $E \subseteq T_n \times V$)
5. Calculate the load of the VM by using the Eq.4.
6. If (load(V_i) < exec(T_j)) then
7. Allocate the task T_j to the virtual machine V_i
8. Else
9. Check the availability of the other virtual machine
10. End
11. If (VM==Overload) then
12. Perform the pre-emption operation to assign the task to the other VM
13. Else
14. Update G
15. End

End

3.3 Pre-emption Policy

The concept of minimum overload and minimum lease (MOML) pre-emption was presented by Mohsen et al [26]. In order to reduce the resource contentions and minimize the virtual machines' overload, the policy is implemented in a two-level approach. The first algorithm shows the procedure of implementing the policy.The first level selects the lowest-overhead virtual machines that are less expensive than the threshold value τ. In the second level, the lowest lease policy is selected. The pre-emption policy behaviour is determined using the τ value.

$$if\ \tau \to 0, then\ MOML\ chooses\ Minimum\ Overhead$$

$$if\ \tau \to \infty, then\ MOML\ chooses\ Minimum\ Lease$$

$$if\ \tau \to median, half\ of\ the\ VMs\ will\ be\ acting\ as\ minimum\ overhead$$

$$another\ half\ act\ as\ minum\ lease$$

Algorithm 3: MOML Pre-emption policy [26]

ovh → Overhead

NL → NoLeases

Input: Y_N → Candidate sets where Y={Y_1, Y2…Y_N}

Output: Selected Y_i

Begin

1. for each $Y_i \in Y_N$ do
2. Compute the total overhead of the Y_i
3. end for
4. $\min \leftarrow \infty$;

5. τ =*getmedian(ovh)*
6. for each $Y_i \in Y_N$ do
7. ovh= getovh(Y_i)
8. NL= cardinality(Y_i)
9. if $ovh \leq \tau$ then
10. if $NL \leq \min$ then
11. Select the Y_i from the Y_N
12. $\min \leftarrow NL$
13. End if
14. End if
15. End for

End

4. SIMULATION ANALYSIS

4.1 Experimental Setup

The proposed model is simulated using the CloudSim simulator [27]. Table 1 shows the simulation parameters of the Task, cloud data center and virtual machine. The Data center is configured with the 20 CPUs and 5TB Storage with RAM capacity 32GB. The Processing capacity is 10000MiPS.

Table 1: Simulation Parameters

Parameter	Value
CPUs	20
Processing Capacity	10000MiPS
Storage	5TB
RAM	32GB
No. of Tasks	[20-100]
Length of the CPU	[1000-10000]MiPS
Task Size	[50-200]MB
Output Size	[10-50]MB
CPUs in VM	1
VM RAM Capacity	1GB
VM Storage Capacity	50GB
VM Bandwidth	500MiPS

4.2 Result Analysis

The proposed hybrid model's performance is evaluated by applying it to the specified input tasks. The turnaround time, which is the amount of time it takes to finish the task, is the main measure of its efficiency.

4.2.1 TURNAROUND TIME

Figure 2 shows the proposed hybrid algorithm's turnaround time against the other known algorithms, such as ACO [14] and BAT [16]. It appears that the proposed model performed better in terms of turnaround time when compared to other algorithms. The reduction in turnaround time can be attributed to the VM attributes when task allocation is made.

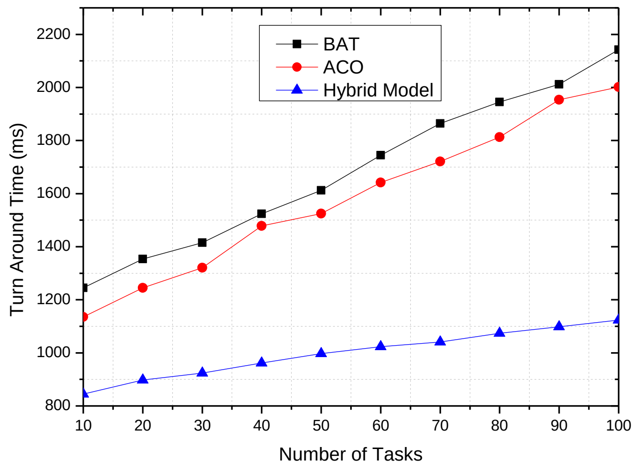

Figure 2: Number of Tasks Vs Turn Around Time

4.2.2 Response Time

The proposed algorithm is evaluated by taking into account the time it takes to execute the tasks. Figure 3 shows the hybrid model's response time along with exiting ACO and BAT algorithms. It shows that the proposed model's response time is 45% lower than that of the ACO and BAT algorithms.

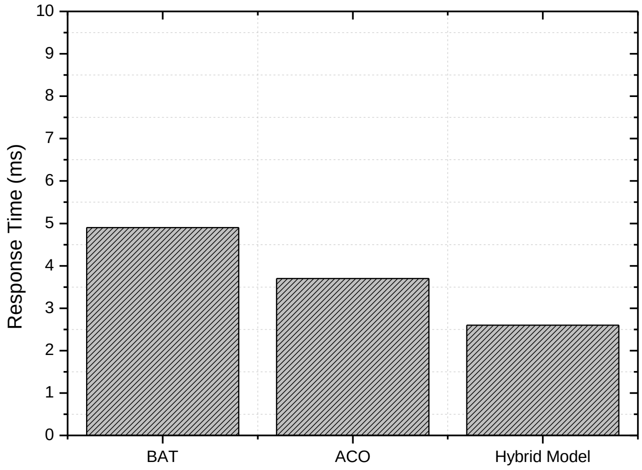

Figure 3: Response Time of Proposed and Existing Algorithms

4.2.3 CPU Utilization

The CPU utilization of the proposed hybrid model is shown in Figure 4 as compared to the existing ACO and BAT algorithms. It utilizes the CPU for each set of instructions in the tasks. The proposed model also has a pre-emption mechanism that can be used to reallocate the tasks when the VM is in an overloaded state.

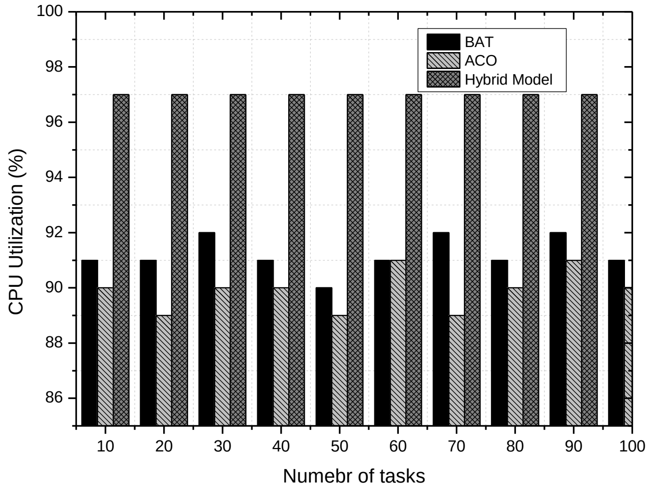

Figure 4: CPU Utilization of Proposed and Existing Algorithms

4.2.4Memory Utilization

Memory utilization of the proposed hybrid model is shown in Figure 5. The proposed model is more efficient when it comes to using the memory compared to the two algorithms such as ACO and BAT. The hybrid model performed better in memory utilization than the BAT algorithm by 30 percent.

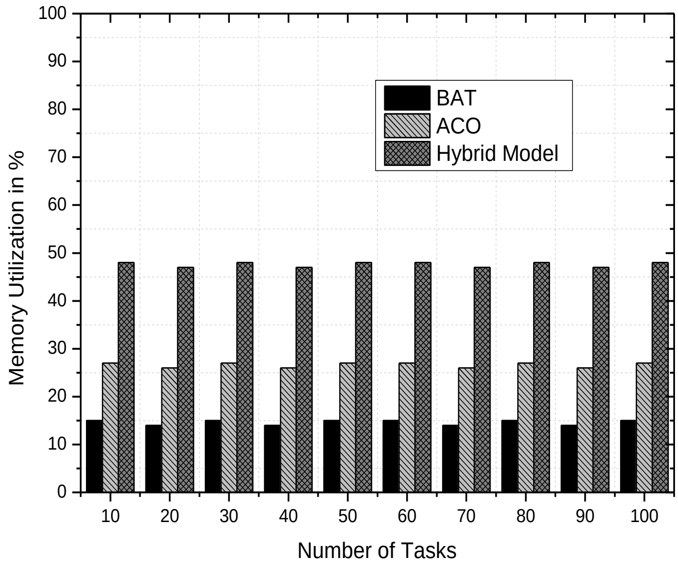

Figure 5: Memory Utilization of Proposed and Existing Algorithms

4.2.5 Bandwidth

In most algorithms, the utilization of bandwidth is not considered when it comes to assessing the performance of the cloud computing system. One of the crucial constraints that can affect the performance of a cloud computing system is the bandwidth utilization. In this framework, the hybrid model is more effective at utilizing the bandwidth compared to ACO and BAT.The graph presented in Figure 6 shows the current and hybrid model's bandwidth utilization.

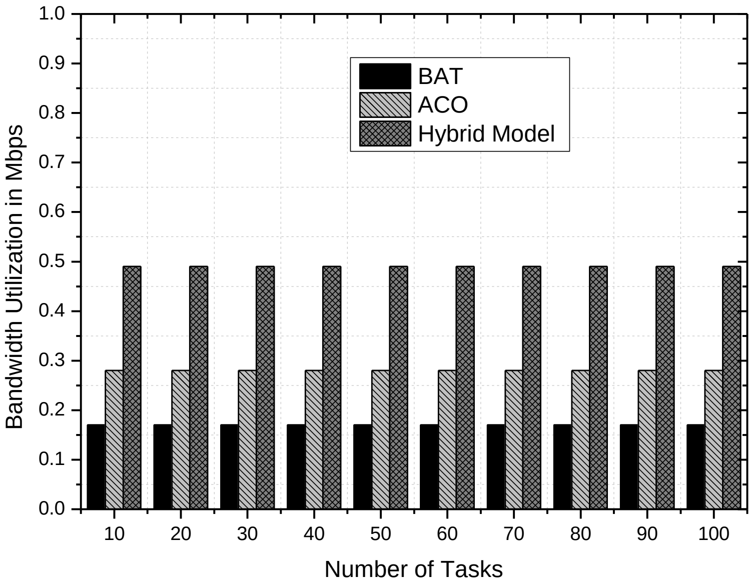

Figure 6: Bandwidth Utilization of Proposed and Existing Algorithms

5. CONCLUSION

This paper developed the hybrid optimization framework that enables efficient task allocation for virtual machines in cloud computing. The hierarchy process is used to manage the prioritized tasks. The task scheduling process is performed using the BAR and BAT models. The former takes into account the VM attributes and the latter considers the task properties. The implementation of a pre-emptive policy known as MOML in the framework minimizes the load on the VMs. It is compared with other algorithms such as the ACO and BAT.The results of the study revealed that the hybrid model is more efficient when it comes to resource utilization, memory utilization, and bandwidth utilization. In the future, it will be used in real-time workflows to demonstrate the effectiveness of the algorithm..

REFERENCES

[1] Liu, Yongkui, Xun Xu, Lin Zhang, Long Wang, and Ray Y. Zhong. "Workload-based multi-task scheduling in cloud manufacturing." Robotics and Computer-Integrated Manufacturing 45 (2017): 3-20.

[2] Abdullahi, Mohammed, and Md Asri Ngadi. "Symbiotic Organism Search optimization based task scheduling in cloud computing environment." Future Generation Computer Systems 56 (2016): 640-650.

[3] Agarwal, Dr, and Saloni Jain. "Efficient optimal algorithm of task scheduling in cloud computing environment." arXiv preprint arXiv:1404.2076 (2014).

[4] Jang, Sung Ho, Tae Young Kim, Jae Kwon Kim, and Jong Sik Lee. "The study of genetic algorithm-based task scheduling for cloud computing." International Journal of Control and Automation 5, no. 4 (2012): 157-162.

[5] Boveiri, Hamid Reza, Raouf Khayami, Mohamed Elhoseny, and M. Gunasekaran. "An efficient Swarm-Intelligence approach for task scheduling in cloud-based internet of things applications." Journal of Ambient Intelligence and Humanized Computing 10, no. 9 (2019): 3469-3479.

[6] Wu, Xiaonian, Mengqing Deng, Runlian Zhang, Bing Zeng, and Shengyuan Zhou. "A task scheduling algorithm based on QoS-driven in cloud computing." Procedia Computer Science 17 (2013): 1162-1169.

[7] Jena, R. K. "Multi objective task scheduling in cloud environment using nested PSO framework." Procedia Computer Science 57 (2015): 1219-1227.

[8] Li, Yibin, Min Chen, Wenyun Dai, and MeikangQiu. "Energy optimization with dynamic task scheduling mobile cloud computing." IEEE Systems Journal 11, no. 1 (2015): 96-105.

[9] Kumar, Pardeep, and Amandeep Verma. "Independent task scheduling in cloud computing by improved genetic algorithm." International Journal of Advanced Research in Computer Science and Software Engineering 2, no. 5 (2012).

[10] Abd Elaziz, Mohamed, ShengwuXiong, K. P. N. Jayasena, and Lin Li. "Task scheduling in cloud computing

based on hybrid moth search algorithm and differential evolution." Knowledge-Based Systems 169 (2019): 39-52.

[11] RAJU, Dasari Naga, and Vankadara SARITHA. "A Survey on Communication Issues in Mobile Cloud Computing." Walailak Journal of Science and Technology (WJST) 15, no. 1 (2018): 1-17.

[12] Nagaraju, Dasari, and Vankadara Saritha. "An evolutionary multi-objective approach for resource scheduling in mobile cloud computing." Int J IntellEngSyst 10, no. 1 (2017): 12-21.

[13] Raju, Dasari Naga, and Vankadara Saritha. "Architecture for fault tolerance in mobile cloud computing using disease resistance approach." International Journal of Communication Networks and Information Security 8, no. 2 (2016): 112.

[14] Tawfeek, Medhat A., Ashraf El-Sisi, Arabi E. Keshk, and Fawzy A. Torkey. "Cloud task scheduling based on ant colony optimization." In 2013 8th international conference on computer engineering & systems (ICCES), pp. 64-69. IEEE, 2013.

[15] Chen, Wei-Neng, and Jun Zhang. "A set-based discrete PSO for cloud workflow scheduling with user-defined QoS constraints." In 2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 773-778. IEEE, 2012.

[16] Lin W, Liang C, Wang JZ, Buyya R (2014) Bandwidth-aware divisible taskscheduling for cloud computing. Software: Practice and Experience 44(2):163–174.

[17] Li, Jian-Feng, and Jian Peng. "Task scheduling algorithm based on improved genetic algorithm in cloud computing environment." JisuanjiYingyong/ Journal of Computer Applications 31, no. 1 (2011): 184-186.

[18] Gai, Keke, MeikangQiu, and Hui Zhao. "Cost-aware multimedia data allocation for heterogeneous memory using genetic algorithm in cloud computing." IEEE transactions on cloud computing (2016).

[19] Maguluri ST, Srikant R (2014) Scheduling jobs with unknown duration inClouds. IEEE/ACM Trans Netw (TON) 22(6):1938–1951.

[20] Cheng C, Li J, Wang Y (2015) An energy-saving task scheduling strategybased on vacation queuing theory in cloud computing. Tsinghua SciTechnol 20(1):28–39.

[21] Ergu D, Kou G, Peng Y, Shi Y, Shi Y (2013) The analytic hierarchy process:task scheduling and resource allocation in cloud computing environment.The Journal of Supercomputing. 64(3):835-848.

[22] Zhu X, Yang LT, Chen H, Wang J, Yin S, Liu X (2014) Real-time tasks orientedenergy-aware scheduling in virtualized clouds. IEEE Transactions on Cloud Computing 2(2):168–180.

[23] Liu X, Zha Y, Yin Q, Peng Y, Qin L (2015) Scheduling parallel jobs withtentative runs and consolidation in the cloud. J SystSoftw 104:141–151.

[24] Handfield R, Walton SV, Sroufe R, Melnyk SA (2002) Applying environmentalcriteria to supplier assessment: a study in the application of the analyticalhierarchy process. Eur J Oper Res 141(1):70–87.

[25] Del Acebo E, de-la Rosa JL (2008) Introducing bar systems: a class of swarmintelligence optimization algorithms. In: AISB convention communication,interaction and social intelligence, pp 18–23.

[26] Salehi, Mohsen Amini, Bahman Javadi, and Rajkumar Buyya. "Resource provisioning based on preempting virtual machines in distributed systems." Concurrency and Computation: Practice and Experience 26, no. 2 (2014): 412-433.

[27] Calheiros, Rodrigo N., Rajiv Ranjan, Anton Beloglazov, César AF De Rose, and Rajkumar Buyya. "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms." Software: Practice and experience 41, no. 1 (2011): 23-50.

..