JOURNAL OF CRITICAL REVIEWS

ISSN- 2394-5125 VOL 08, ISSUE 01, 2021

# STUDY ON CLOUD COMPUTING FOR EFFICIENT RESOURCE ALLOCATION AND SCHEDULING APPROACHES

**ATHMAKURI NAVEEN KUMAR**

Senior software engineer, software developers industry, GLOSOFT Technologies PVT LTD, Hyderabad, India

**Email id:** shanvinaveen5@gmail.com

## ABSTRACT

The term "cloud storage" refers to a variety of online services that enable users to store and share digital media such as documents, data, photos, and videos. You may access these files from anywhere and on any device (laptop, mobile phone, tablet etc). With cloud computing, users are able to allocate their computing needs among a shared pool of powerful machines, increasing their access to resources like processing power, storage space, and software services. There is a growing population of internet users who rely on cloud-based resources. Large amounts of data are transferred from users to hosts and from hosts to users in the cloud environment, but as demand for cloud services grows, the associated cost and complexity for the cloud provider may become unsustainable. There may be times when two or more users make a request for the same item. Given these constraints, it's not easy to decide which host to use to gain access to the necessary resources and build a virtual machine (VM) in which to run the necessary applications in a way that maximises efficiency while minimising costs. Scheduling tasks in a cloud computing environment to maximise efficiency is one solution to this issue. An approach to job scheduling is provided by this project. In this research, we make an effort to suggest a host selection model based on shortest execution time to reduce overhead.

## 1. INTRODUCTION

In the realm of distributed computing, cloud computing is a fast-developing innovation. To store data, do data analytics, and power Internet of Things (IoT) applications are just a few of the many uses for cloud computing [1]. Cloud computing is a technology that has altered the methods in which businesses and individuals deliver their services. In order to save its registered users the cost of purchasing and maintaining their own computers, the service offers a variety of services as web services. IaaS (Infrastructure as a Service), PaaS (Platform as a Service), and SaaS (Software as a Service) are all offerings that can be found in the cloud [2]. For each service category, customers are required to contact providers over the Internet to make their demands. The onus of seeing to it that user requests are met rests on the service provider's shoulders. The use of scheduling algorithms is common practise among Service Providers for the purpose of managing their available computing resources and completing incoming requests (tasks).

With the help of task scheduling and resource management, service providers may maximise both income and resource utilisation. Scheduling and allocating resources are significant obstacles to the performance of cloud computing resources in practise. Because of this, investigations into cloud-based work scheduling have gained a lot of attention from academics. Scheduling entails prioritising requests (tasks) in order to make the best use of available resources. Users of cloud services must make service requests over the web, as this is the method via which these services are delivered.

ISSN- 2394-5125 VOL 08, ISSUE 01, 2021

Each service may get multiple requests (tasks) simultaneously due to the large number of people who utilise it. In the absence of scheduling, tasks in such a system can experience longer wait times, and in some cases, they might even fail to complete. Many factors must be taken into account by the scheduler at the time of scheduling, including the type of work, its size, the time it takes to complete, the resources that are available, the tasks that are waiting to be scheduled, the queue for those tasks, and the current resource load. One of the major challenges of cloud computing is the scheduling of tasks. Having tasks well scheduled can help ensure that all available resources are put to good use. One of cloud computing's primary benefits is that it encourages more efficient use of resources [3]. Task scheduling and resource allocation are hence complementary processes. In certain ways, both influence one another.

Users don't have to think about the hosting infrastructure anymore; they may simply access material whenever they want. The hosting provider's infrastructure comprises of a wide range of machines, each with their own unique set of capabilities. Any network setup that has access to the Internet can benefit from cloud computing. Profits for cloud service companies come from their customers' utilisation of their services.

When making use of a cloud service, the customer can take advantage of the full complement of available computing resources. Cloud services are offered on a "pay as you go" basis. The user of a cloud service can adjust the amount of accessible resources up or down as needed. It's a huge perk of cloud computing, but customers may have to pay more for it.

An end user of a cloud service has the flexibility to rent and release resources as needed. The user of a cloud service is able to choose any service to meet their specific requirements. An issue that has arisen as a result of users' ability to pick and choose which services they utilise is that it is now impossible to reliably foresee what they will want next. Therefore, study into cloud computing must incorporate work scheduling and resource allocation. When compared to randomly assigning resources, scheduling and load balancing methods significantly improve resource use efficiency. Complex problems are a common focus of cloud computing's use (user requests). It is suggested that a scheduling algorithm be used in order to resolve problems with complicated tasks.

## 2. LITERATURE REVIEW

Multi-objective methods based on the enhanced differential evolution algorithm were proposed by Tsai et al. [1]. The current approach gives a time and money paradigm for cloud computing. However, this method does not account for contextual differences between activities.

The load balancing and scheduling technique suggested by Magukuri et al. [2] does not take into account the sizes of individual jobs. When responding to requests, the authors factored in the server's refresh rates.

To accommodate vacations, Cheng et al. [3] first proposed a vacation queuing model for work scheduling. This approach does not demonstrate efficient use of resources.

Taking bandwidth into account as a resource was central to a proposal for task scheduling made by Lin et al. [4]. In order to divide up the available manpower, we devised a nonlinear programming model.

ISSN- 2394-5125 VOL 08, ISSUE 01, 2021

In [5], Ergu et al. suggested a method for prioritising tasks using AHP rankings. When it comes to scheduling jobs that need to be completed in real time, Zhu et al. [6] established the concept of rolling-horizon scheduling architecture. The authors have provided examples to show how resource allocation during task scheduling can lead to significant energy savings.

For use in the cloud, Ghanbari et al. [7] suggested a priority-based job scheduling method. A number of factors and characteristics are taken into account to reach a conclusion.

The optimised cost of energy and queuing time limitations were first presented by Polverini et al. [8].

Scheduling improvements using meta-heuristic and particle swarm optimization were proposed by Alejandra et al. [9]. Alternate ant colony optimization was proposed by Keshk et al. [10].

A job's makespan is boosted by this technique. The system does not take into account the value of available resources or the difficulty of assigned tasks. For the purpose of allocating server loads, Shamsollah et al. [11] devised a solution based on a multi-criteria algorithm. With the help of an analytical hierarchy approach, Shamsollah et al. [11] established a priority-based system for divisible load scheduling. The resource allocation challenge described by Gougarzi et al. [12] seeks to minimise the total energy cost of cloud computing systems while probabilistically achieving the stated client-level SLAs. The writers here use a backwards approach by imposing costs on the client if they fail to achieve the service level agreements. When it comes to the aforementioned job scheduling and resource allocation dilemma, some writers have proposed a heuristic algorithm.

Central load balancing decision model for cloud settings was presented by Radojevic et al. [13], which automates the scheduling process and lowers the need for human administrators.

## 3. RESEARCH METHODOLOGY

### 3.1 Introduction

To a large extent, "Research" is responsible for the development and enhancement of any field. In spite of this, research is a challenging endeavour that calls for in-depth expertise and unwavering commitment on the part of the researchers. Researchers in the field of computer science can benefit from this chapter's discussion of the many approaches to research, which will aid them in getting their work underway. This strategy is not limited to any particular area of study.

### 3.2 Problem Formulation

It is the responsibility of a cloud service provider (CSP) to manage the system's resources (cloud service provider). Researchers have created a wide variety of scheduling algorithms for optimal resource management, each taking into account factors like time, cost, energy, network, memory, and so on.

First-in, first-out (FCFS) scheduling prioritises tasks in order of arrival. The sequence in which jobs are received by the cloud servers is used to determine where they will be processed. The jobs are pulled from the queue and distributed to the available resources one by one. When using SJF (Shortest Job First), however, jobs are queued according to their size and scheduled using the first-come, first-served (FCFS) method. In the min-min scheduling strategy, the quickest work in a sorted queue is chosen and given to the resource with the shortest expected runtime. "Max-min" scheduling refers to the practise of assigning the most time-consuming work to the computational resource with the slowest throughput, i.e., the one that can complete the request in the least amount of time. But in RR

(Round Robin), tasks are queued up first-come-first-served (FCFS). In a time-sharing arrangement, virtual machines (VMs) are given certain tasks from the queue for a set amount of time (the "Time Quantum"). Other jobs are chosen after this time period, and if any tasks are still unfinished, they receive resources after everyone else's have been completed.

### 3.3 CloudSim Architecture

CloudSim's layered design, as depicted in fig. 3.1,

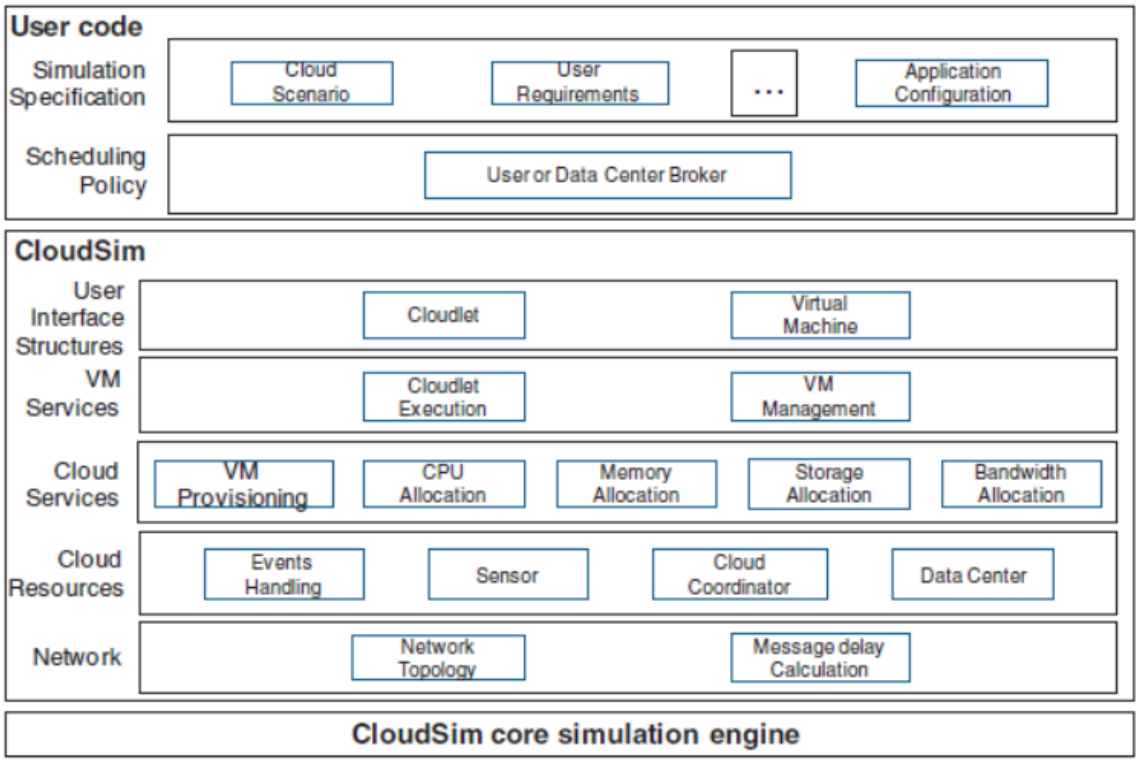


Figure 3.1: Layered CloudSim Architecture.

User Code is located at the top of the simulation stack, and it provides information about the host, such as the host configuration and the number of virtual machines (VMs), applications, tasks, users, service types, and scheduling policies for the broker.

In the following layer, true cloudSim techniques are implemented, such as the user interface structure (user tasks with their management), virtual machine (VM) services (actual execution of tasks/cloudlets with VM management), and numerous cloud services (including resource allocation and networking) (management related to network e.g. topology and message passing).

### 3.4 ABC Scheduling Using Priority Approach in Cloud Computing Environment

Distributed computing, virtualization, and grid computing have all been improved upon by cloud computing. Due to the changing nature of activities and the associated expenses when using cloud resources, the conventional scheduling methods are no longer applicable. Since each cloud resource has its own price and computational performance, this chapter is concerned with allocating tasks in

ISSN- 2394-5125 VOL 08, ISSUE 01, 2021

groups to those resources. Job grouping improves the communication to computation ratio by facilitating communication between resources and coarse-grained jobs. A method that takes this into account by arranging tasks according to their relative costs has been proposed. The suggested method utilises the fundamental principle of the ABC (Activity Based Cost) scheduling algorithm to apply cost to data centres. There are two distinct stages to the suggested process. Initial steps involve classifying three data centres, from least expensive to most expensive, so that work can be distributed among them. In the second stage, we create three distinct priority queues and distribute jobs to VMs based on their relative importance.

### 3.5 Cost Analysis and Pricing in Cloud

What a service provider receives in exchange for those services, and how many customization options are made available to end users, are both factors that can be influenced by pricing policy [96]. It is crucial that reasonable pricing models be created for cloud computing in order for it to be successful in the IT industry [97]. Here is how we arrive at our prices:

• Fixed. in which the cost of the service is always the same regardless of how often the user uses it.

• Dynamic. where prices fluctuate in real time in response to fluctuations in the market, making it entirely market driven.

### 3.6 ABC (Activity Based Cost) Approach

It is not just the cost of resources that may be calculated with the help of the activity-based cost approach, but also the cost of performance. When using an activity-based approach, it is expected that the associated costs will also vary, as each action is unique. In this method, expenses are calculated based on the amount of processing power, storage space, and time spent satisfying individual requests. Each task's priority is determined after the total cost has been determined, and the tasks are then ranked from highest to lowest. In order to determine the relative importance of each task, we can use the following formula:

$$L_k = \sum_{i=o}^{n} (R_{i,k} * C_{i,k})/P_k$$

## 4. EXPERIMENTAL ANALYSIS AND SETUP

This section compares the times at which five different heuristic scheduling algorithms (First Come First Serve, SJF, RR, Max-min, and Min-min) run in a cloud computing setting. CloudSim 3.0.3 is used throughout the investigation to simulate the environment. Scheduling may be done at the data centre, cloudlet, and VM levels in this simulator. VM-level comparisons have been carried out in this section. Table 1 uses configuration to compare the aforementioned scheduling strategies. Several virtual machines (VMs) make up a host. Three virtual machines (VMs) are used in the current studies, with the remaining host setup detailed in table 1a.

ISSN- 2394-5125 VOL 08, ISSUE 01, 2021

Table 1: Setup for experiment

**(a) Host Configuration.**

| | |
|---|---|
| No. Of VMs | 3 |
| RAM | 1024 MB |
| Storage | 100000 MB |
| BW (Bandwidth) | 10000 Kbps |

**(b) VM Configuration.**

| | |
|---|---|
| MIPS | 300 |
| Size | 1000 MB |
| RAM | 512 MB |
| BW (Bandwidth) | 1000 Kbps |

Table 1b displays the VM setup that was used throughout the experimental investigation. Algorithms were executed using the CloudSim3.0.3 simulator with these settings. The results of using this strategy are displayed in table 2.

Table 2: Comparison Among Various Scheduling Algorithms.

| | Completion Time (in seconds) | | | | |
|---|---|---|---|---|---|
| **No. of cloudlets/tasks** | **FCFS** | **SJF** | **RR** | **Max-Min** | **Min-Min** |
| 50 | 79 | 70 | 65 | 59 | 55 |
| 100 | 102 | 89 | 81 | 79 | 76 |
| 200 | 165 | 154 | 150 | 135 | 133 |
| 300 | 209 | 201 | 192 | 188 | 186 |
| 400 | 237 | 229 | 217 | 214 | 210 |

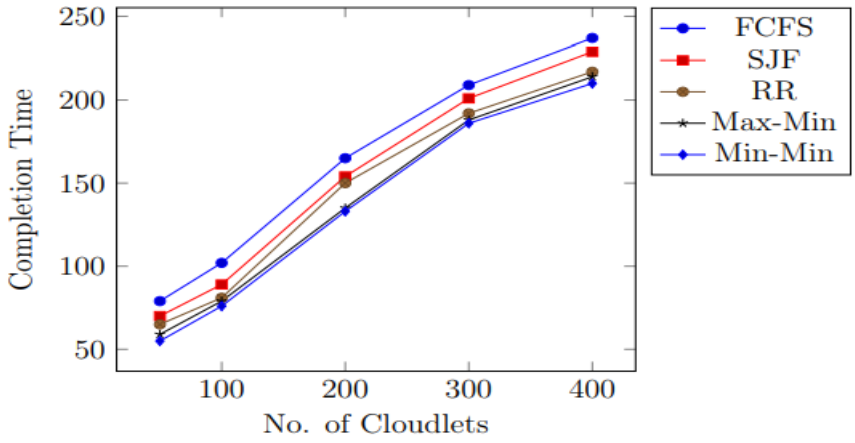


Figure 4.1: Graphical Representation of Results.

ISSN- 2394-5125 VOL 08, ISSUE 01, 2021

Figure 4.1 displays the durations of execution for the following scheduling algorithms: FCFS, SJF, RR, Max-min, and Min-Min. Time required by each varies, with FCFS being the longest and Min-min scheduling being the shortest. Because the quickest jobs are completed first under a minmin scheduling philosophy, the total time it takes to complete any given task can be kept to a minimum.

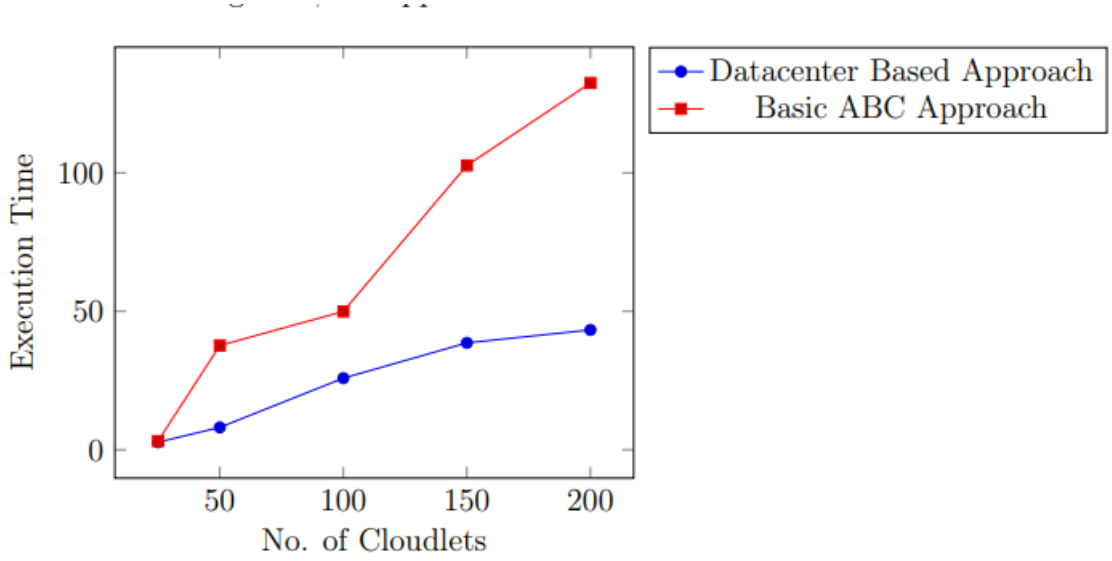


Figure 4.2: Execution of All Tasks Using High Priority Queue.

Figure 4.2 displays the outcomes when only high-priority jobs are taken into account. In order to go through the most important jobs as quickly as possible, they will all be given to the fastest processors available.

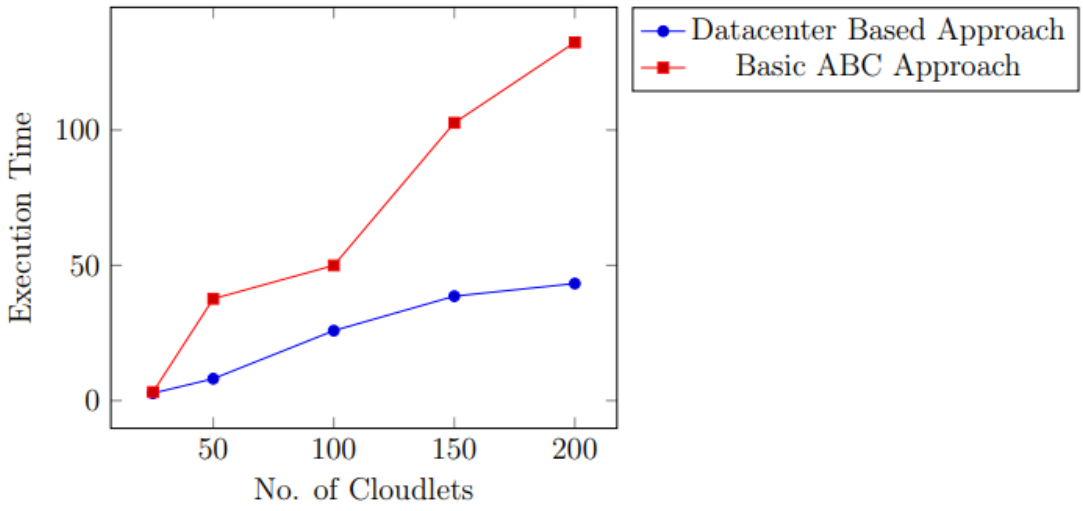


Figure 4.3: Execution of All Tasks Using Medium Priority Queue.

ISSN- 2394-5125 VOL 08, ISSUE 01, 2021

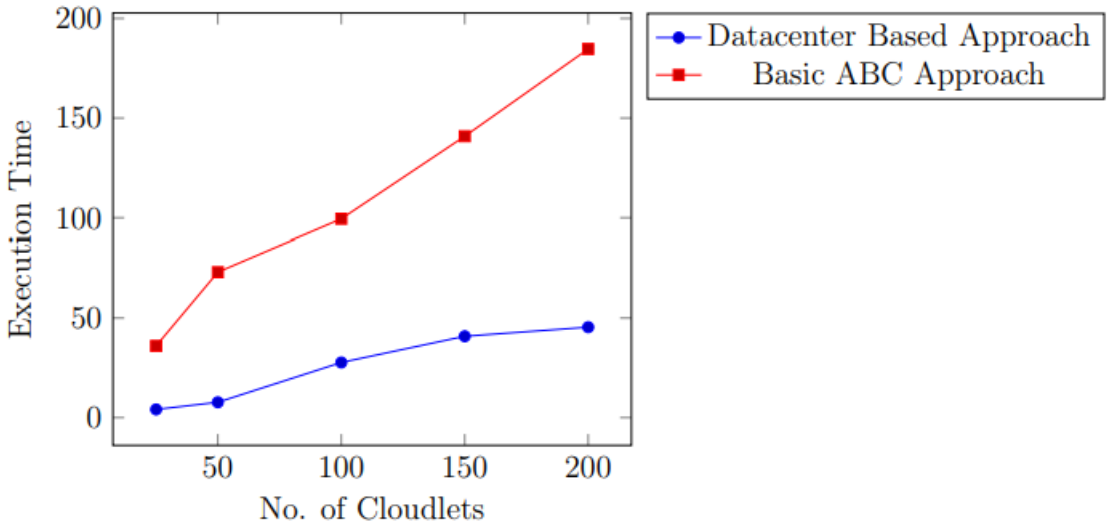


Figure 4.4: Execution of All Tasks Using Low Priority Queue.

As depicted in figures 4.3 and 4.4, jobs with medium and low priority are carried out in a similar fashion. If you look at figure 4.4, you can see the end outcome of finishing all of your top priorities.

## CONCLUSION

The proposed approach incorporated approaches for providing necessary resources and organising activities to meet strict deadlines. The sequence of approaches has been modelled to remove the obstacles to meeting deadlines for jobs running on a single host in real time. Scalability is a feature that allows for the addition or subtraction of resources based on demand. In addition, the proposed method was used in conjunction with the existing method to promptly meet the demands, with the resources being vertically scaled and optimised by being shut down when they were not needed during execution. The submitted jobs have their estimated costs determined and are then transmitted to the datacenter. The proposed method takes into account the anticipated price of tasks before they are actually performed. The proposed method is found to have significantly better outcomes than the standard ABC method, with benefits extending all the way down to the time spent on completing a single assignment.

## REFERENCES

1. Tsai J-T, Fang J-C, Chou J-H (2013) Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm. Comput Oper Res 40(12):3045–3055

2. Maguluri ST, Srikant R (2014) Scheduling jobs with unknown duration in clouds. IEEE/ACM Trans Netw (TON) 22(6):1938–1951

3. Cheng C, Li J, Wang Y (2015) An energy-saving task scheduling strategy based on vacation queuing theory in cloud computing. Tsinghua Sci Technol 20(1):28–39

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

5. 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

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

7. Ghanbari S, Othman M, Leong WJ, Bakar MRA (2014) Multi-criteria based algorithm for scheduling divisible load. In: Proceedings of the first international conference on advanced data and information engineering (DaEng-2013), pp 547–554

8. Polverini M, Cianfrani A, Ren S, Vasilakos AV (2014) Thermal aware scheduling of batch jobs in geographically distributed data centers. IEEE Transactions on Cloud Computing 2(1):71–84

9. Rodriguez MA, Buyya R (2014) Deadline based resource provisioningand scheduling algorithm for scientific workows on clouds. IEEE Transactions on Cloud Computing 2(2):222–235

10. Keshk AE, El-Sisi AB, Tawfeek MA (2014) Cloud task scheduling for load balancing based on intelligent strategy. Int J Intell Syst Appl 6(5):25

11. Shamsollah G, Othman M (2012) Priority based job scheduling algorithm in cloud computing. Procedia Engineering 50:778–785

12. Goudarzi H, Ghasemazar M, Pedram M (2012) Sla-based optimization of power and migration cost in cloud computing. In Proceedings of the 2012 12th IEEE/ ACM International Symposium on Cluster, Cloud and Grid Computing (ccgrid 2012) (pp. 172-179). IEEE Computer Society

13. Radojevic B, Zagar M (2011) Analysis of issues with load balancing algorithms in hosted (cloud) environments. In: MIPRO, 2011 proceedings of the 34th international convention, pp 416–420

13. Min AN, Bilal QM, Saleh A, Omer FR (2019) Cost-efcient resource allocation for real-time tasks in embedded systems. Sustain Cities Soc. https://doi.org/10.1016/j.scs.2019.101523

14. Kholidy HA (2020) An intelligent swarm based prediction approach for predicting cloud computing user resource needs. Comput Commun 151:133–144. https://doi.org/10.1016/j.comcom.2019.12.028

15. Than MM, Thein T (2020) Energy-saving resource allocation in cloud data centers. In: IEEE conference on computer applications (ICCA), Yangon, Myanmar, pp 1–6. https://doi.org/10.1109/ ICCA49400.2020.9022819

16. Srimoyee B, Rituparna D, Sunirmal K, Sarbani R (2020) Energyefcient migration techniques for cloud environment: a step toward green computing. J Supercomput 76:5192–5220. https:// doi.org/10.1007/s11227-019-02801-0

17. Mansouri N, Javidi MM (2020) Cost-based job scheduling strategy in cloud computing environments. Distrib Parallel Databases 38:365–400. https://doi.org/10.1007/s10619-019-07273-y

18. Reshmi B, Poongodi P (2020) Proft and resource availabilityconstrained optimal handling of high-performance scientifc computing tasks. J Supercomput 76:4247–4261. https://doi. org/10.1007/s11227-018-2332-7

ISSN- 2394-5125 VOL 08, ISSUE 01, 2021

19. Tripathi A, Pathak I, Vidyarthi DP (2020) Modifed dragonfy algorithm for optimal virtual machine placement in cloud computing. J Netw Syst Manage. https://doi.org/10.1007/ s10922-020-09538-9

20. Dorigo Marco and Gambardella Luca Maria (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Comput 1(1):53–66. https://doi.org/10.1109/4235.585892