AWRR Algorithm for Data Center to Optimize Cloud Computing

R.Ashokkumar[1], Ashok kumar.K[2], A.Gokul[3], T.Murali[4]

[1]*Assistant Professor, Department of Artificial Intelligence and Data Science, Paavai College of Engineering, Pachal, Namakkal*

[2]*Assistant Professor, Department of Information Technology, Excel Engineering College, Komarapalayam, Namakkal*

[3]*Assistant Professor, Department of Computer science and Engineering, Paavai College of Engineering, Pachal, Namakkal*

[4]*Assistant Professor, Department of Computer Science and Engineering , Paavai College of Engineering, Pachal, Namakkal*

Abstract: Weighted Round Robin (WRR) is a extensively used load-balancing algorithm that distributes requests amongst a couple of servers primarily based on predefined weights. However, regular WRR techniques go through from inefficiencies when managing dynamic workloads, various server capacities, and fluctuating community conditions. This paper provides an Advanced Weighted Round Robin (AWRR)Algorithm, which enhances ordinary WRR by way of incorporating dynamic weight adjustments, adaptive load monitoring, and predictive analytics the proposed AWRR algorithm dynamically updates server weights based totally on real-time overall performance metrics, which include CPU utilization, response time, and request queue length. Additionally, it integrates laptop studying methods to predict workload patterns and proactively modify weights, making sure best useful resource utilization and decreased latency. The implementation of honest loader distribution mechanisms prevents server overloads and enhances ordinary gadget stability.

Keywords: Cloud computing, load balancing, time span, optimization, task scheduling.

INTRODUCTION

As we shift greater toward on-line storage and services, Cloud Computing technological know- how will become an imperative phase of the business. This technological know-how affords offerings via a variety of sorts such as in software program with the aid of net browsers, in Platforms such as designing and creating cloud-based applications. In the Infrastructure, the backend is managed by means of Cloud Service Providers (CSPs) such as preserving Data Centers, servers, etc. Although there exist many different carrier shipping fashions in this technology, however, in this research, the focal point is on the Infrastructure as a Service (IaaS) model. It offers with the server-side of this science for aid allocation.

Virtualization is the spine and fundamental characteristic of cloud-based applications. This method can notably have an effect on the overall performance of the scalable and on-demand The companion editor coordinating the assessment of this manuscript and Services furnished to consumers if the migration procedure and allocation of digital computer assets are treated inefficiently. According cloud overall performance is proved to be in the pinnacle three Cloud Computing challenges. This lookup objectives to decorate useful resource allocation in the IaaS model; this idea is critical as it offers with the balancing of assets supplied to purchasers and the workload/user requests on servers. The cloud customers get entry to offerings with the aid of sending requests; these are represented in Virtual Machines (VMs)in the cloud environment. CSPs ought to supply offerings that are recommended to agencies and extend consumer satisfaction. Thus, Adjusting calculation is created primarily centering on the IaaS demonstrate out of the three benefit models in the cloud where creators bargain with the Cloud Computing technology's backend, such as server work load. There are two components in a ordinary cloud environment: the frontend is the client side, and it is available by interfacing to the models where the

Information Center store different physical machines (known as servers).

Approaching client demands are gotten from the application are powerfully planned, and through virtualization, the fundamental assets are apportioned to clients. The virtualization strategy is too mindful for adjusting the stack in the whole framework, planning, and effective assignment of resources.

CSPs and cloud clients can use the advantage of virtualization as well as energetic assignment planning procedures. In this way, productive planning can exceedingly decrease execution time and increment the proportion of asset utilization in cloud-based applications.

Task Planning is a handle that profoundly relates to work stack adjusting. As outlined in figure 1 over, as clients send demands, the assignment is submitted through a cloud broker; this is where analysts ought to center on giving a proficient calculation. The proposed calculation ought to productively yield occupations to fitting VMs taking after fundamental parameters such as due date to keep up a tall quality of administrations and guaranteeing the demands sent by clients are executed and completed inside these particular prerequisites given in the Benefit Level Understanding (SLA) archive. The client sends demands through the Web. These demands are put away in Virtual Machines (VMs) and CSP in each conveyances how must keep up the QoS by guaranteeing the users' demands can be executed and completed inside a particular due date. This prepare depends exceedingly on the planning policy's proficiency (Information Broker) which ought to be modified to result in at all procedure for adjusting workload among the machines and servers. Productive planning and utilization of assets can be accomplished by planning and creating a energetic

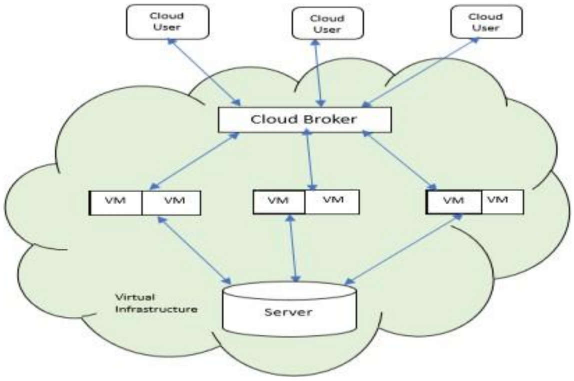


FIGURE1.Taskschedulingincloudcomputing

RESEARCH CONTRIBUTION:

This sub section highlights the commitment made by the creators in this paper.

The investigate primarily points to optimize the cloud assets by upgrading the Stack Adjusting prepare through effective Errand

Planning methods. Our commitment to the think about can be summarized as follows:

A study of existing Stack Adjusting and Err and Planning algorithms.

A proposed Stack Adjusting calculation addresses the VM infringement issue in the cloud and gives high-quality benefit in terms of work load planning and adjusting. In spite of the fact that analysts have tended to this issue in the past, most do not consider vital SoS parameters such as Due date and Completion Time.

Additionally, the proposed calculation incorporates the relocation of stack to adjust VMs, which is still not completely tended to yet.

Algorithm comes about in diminishing two primary Stack Adjusting parameters: time span and Execution time in the cloud applications and change on Asset utilization

PROBLEMSTATEMENT

This are a highlights the issue articulation which is extricated from the audit made in this inquire about. Taking after the arrangements to these issues, the unused calculation is proposed.

Dealing with approaching client requests/tasks and keeping a adjusted work load in cloud frameworks can be challenging due to unseemly allotment to Machines by considering two primary parameters: Due date and Completion Time. In Cloud Computing, Information Center (DC) can be depicted as huge capacity for cloud servers and information. DC gets demands and sends them to the dynamic stack balancer. In this layer of the show, the proposed calculation is implemented.

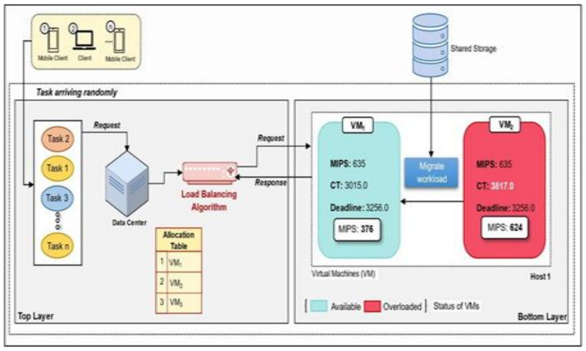


FIGURE2.Proposedframework

Bottom Layer: bargains with assignment of client demands to Virtual Machines (VMs). As the figure outlines, we have a essential group of VMs; VM2's status is set to tall need since it abuses the SLA prerequisite, which implies its Completion Time is higher than the Due date. In this way, the proposed LBA ought to apply a movement strategy to exchange the work load to another accessible Virtual Machine by reconfiguring the MIPS of both VMS

PROPOSEDLOADBALANCINGALGORITHM

- In this subsection, the proposed LB calculation is clarified to highlight the suspicions made in the execution, the algorithm's pseudo code, and at long last, the flowchart.

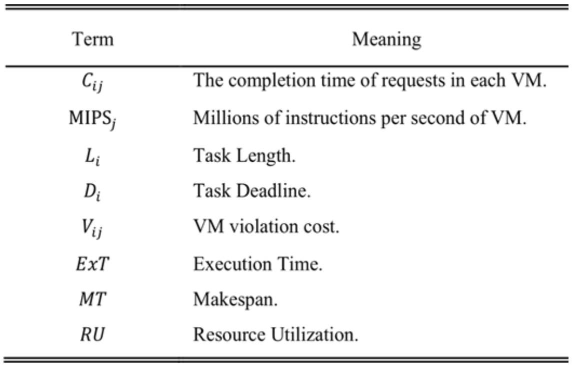
  The proposed calculation points to make strides the cloud's execution by considering both angles of Errand Planning and Stack Adjusting. It utilizes all accessible CPUs in machines and plans errands suitably to decrease time span, Execution Time, and maximize asset utilization. Underneath are the presumptions made in the proposed algorithm:
- One-to-many cloudlets (too known as errand or client ask) per Virtual Machine (VM).
- Cloudlets arrive in a irregular arrange (Entry Time)
- Each Cloudlet has a length, a time to total known as Due date (included in Benefit Level Understanding archive), a completion time, and at long last, the entry time.
- The proposed calculation checks the completion time for each workload against the Due date.
- If there is any infringement, whereby the completion time surpasses the Due date, at that point the proposed calculation will reconfigure the VM's need based on its CPU. If it is in a effective state, the cloud lets get planned else; it will relocate the VM's workload.
- Expected Completion Time is calculated by taking the cloudlet length (moreover known as Million instruction per moment (MIPS) and isolating it by Virtual Machine MIPS (too known as CPU).

  Initially, all VMs share a break even with parcel of the accessible CPU; at that point, it is reconfigured based on the infringement status. The CPU is set to its full utilization in the proposed algorithm.

Table 2 below shows the terms used in the proposed algorithm and their meanings.

The Pseudo code of the proposed LB Calculation is given underneath. The reason of giving the pseudo code is to outline the equations, the parameters, and the choices made in this Stack Adjusting algorithm.

TABLE1.Indicatetermsandmeaning.



Term	Meaning
C_{ij}	The completion time of requests in each VM.
MIPS_j	Millions of instructions per second of VM.
L_i	Task Length.
D_i	Task Deadline.
V_{ij}	VM violation cost.
ExT	Execution Time.
MT	Makespan.
RU	Resource Utilization.

of 10000 and underneath 2000 for Due date. These values act as an input to the calculation, and they make up the workload in cloud framework

Flow chart is at that point allotted an break even with share of CPU based on the add up to workload. By calculating the VM fetched, we can decide whether the VM has abused SLAPre requisites by observing if the completion time is higher than the Due date. If yes, at that point the work load is moved to another VM, and soon. This way, the calculation completely utilizes all CPU, and the framework workload is balanced.

III.IMPLEMENTATION

In this segment, the test comes about and execution points of interest of the proposed LBA calculation are described.

SIMULATIONSETUP

Cloud Sim recreation device is the most prevalent instrument utilized by analysts and designers these days for cloud-related issues in the investigate field. It can essentially dispose of the require and costs of computing offices for execution assessment and modeling the inquire about arrangement. This reenactment apparatus is an outside system that can be downloaded and imported to programming program such as Obscure, Net Beans IDE, Maven etc. To recreate the Cloud Computing environment, the Cloud Sim toolkit is coordinates into Net Beans IDE 8.2, and the Working Framework utilized is Windows 10.

PERFORMANCEMETRICS

The execution of the proposed LB calculation was analyzed based on three parameters beneath the cloud environment.

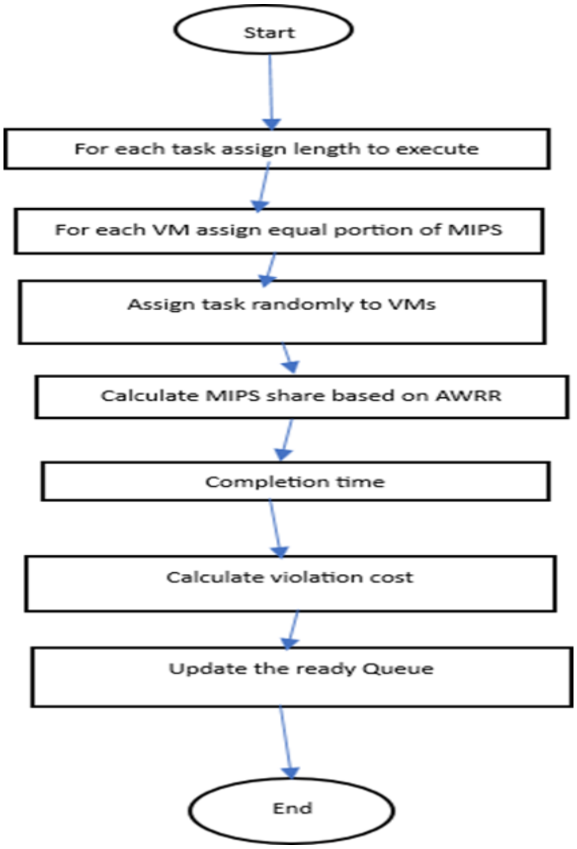

Start
For each task assign length to execute
For each VM assign equal portion of MIPS
Assign task randomly to VMs
Calculate MIPS share based on AWRR
Completion time
Calculate violation cost
Update the ready Queue
End

FIGURE3. Flowchart of the proposed algorithm. Flowchart of the proposed algorithm

The taking after execution lattice is utilized to degree and assess the performance:

1) Time span (TT): it is the add up to time taking to get a cloud let planned. This is generally utilized to degree planning algorithms' effectiveness with regard to time. It ought to be decreased to permit effective execution of assignments and to discharge the assets for other assignments. It is measured by utilizing the conditions underneath proposed by creators in, where CT signifies cloudlet completion time, and n signifies the number of Virtual Machines.

   MT = Max(CT)

   $$\text{TT avg} = \left(\frac{\sum Max(CT)}{n}\right)$$

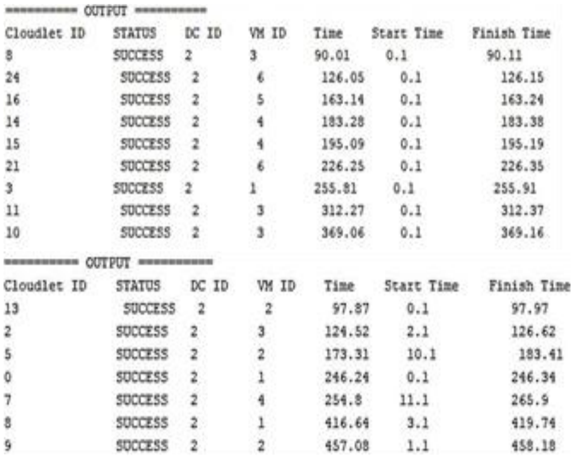
2) Execution Time(ExT): it is the correct time taken to execute the given errands(cloudlets) on a virtual machine. This metric ought to be decreased to accomplish way better execution of the calculation. where AcT indicates Cloudlet Genuine CPU Time and n signifies the number of Cloudlets.

   ExT =AcT

   $$\text{ExT avg} = \left(\frac{\sum AcT}{n}\right)$$

IV. RESULTS & DISCUSSION

The reason of carrying this try is to demonstrate the diminishment in Time span, execution time, and the expanding of as set utilization in a energetic cloud environment. Amid the testing of the calculation, we have considered preemptive planning of assignments. This implies the assignment can be hindered amid execution if the workload damages SLA, it can be moved to another asset to total execution, as appeared in figure Amid the planning handle, a few SoS execution parameters of cloudlets are considered, such as:



```
========== OUTPUT ==========
Cloudlet ID    STATUS   DC ID   VM ID   Time     Start Time   Finish Time
8              SUCCESS  2       3       90.01    0.1          90.11
24              SUCCESS  2       6       126.05   0.1           126.15
16              SUCCESS  2       5       163.14   0.1           163.24
14              SUCCESS  2       4       183.28   0.1           183.38
15              SUCCESS  2       4       195.09   0.1           195.19
21              SUCCESS  2       6       226.25   0.1           226.35
3              SUCCESS  2       1       255.81   0.1          255.91
11              SUCCESS  2       3       312.27   0.1           312.37
10              SUCCESS  2       3       369.06   0.1           369.16
========== OUTPUT ==========
Cloudlet ID    STATUS    DC ID   VM ID   Time     Start Time   Finish Time
13              SUCCESS   2       2        97.87   0.1           97.97
2              SUCCESS   2       3       124.52   2.1           126.62
5              SUCCESS   2       2       173.31   10.1           183.41
0              SUCCESS   2       1       246.24   0.1           246.34
7              SUCCESS   2       4       254.8   11.1           265.9
8              SUCCESS   2       1       416.64   3.1           419.74
9              SUCCESS   2       2       457.08   1.1           458.18
```

FIGURE4.Same arrival time & random arrival time.

1) Entry Time: shows the time cloudlets arrive or when the calculation gets the client ask. This is known as the cloudlet begin time in the Cloud Sim environment. In Cloud Sim, by default, all cloudlets arrive at the broker at the same entry time. In this try, this has been altered to make changes for putting off the accommodation of cloudlets; this is known as a irregular Entry Time parameter. The broker will at that point relegate the cloudlets in a arbitrary arrange to the VMs based on the code actualized in this strategy. Utilizing this parameter, we can plan an calculation to work In a energetic environment where the entry time can be distinctive for each request.

2) Task Length: recognizes the measure of err and

sin bytes; littler assignments lead to more

TABLE2.Cloud Sim simulator requirements.

Type	Parameters	Value
Cloudlet (Task)	Length of tasks (in bytes)	Random < upper threshold (1000000)
	Total number of tasks	1-40
Virtual Machine (VM)	Number of VMs	2-6
	Processor speed	9980-15000 MIPS
	RAM in a single VM	512 Mb
	Bandwidth	1000 Mb
	Cloudlet Scheduler	Time Shared
	Number of processor elements (PEs) requirement	1
	VMM	Xen
Data Center	Number of Data Centers	2
	Number of Hosts	1
	VmScheduler	Time Shared

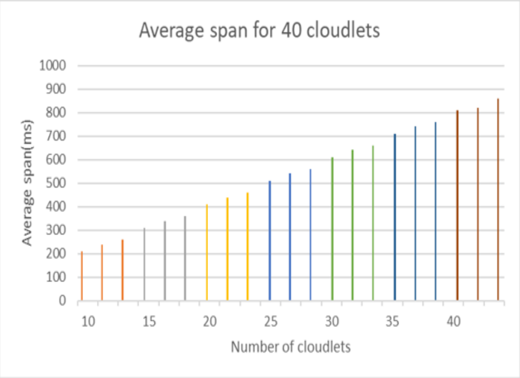


FIGURE5.Average Time span for 40 tasks

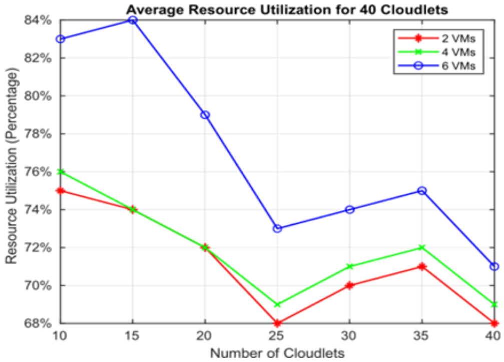
asset utilization. In Cloud Sim, each Cloudlet must have a length esteem that demonstrates the cloudlet sort, whether it is a overwhelming ask, light, or medium. In this test, length has been distinguished and doled out arbitrarily to each Cloud let. All the cloud let sought to have irregular values to separate the client demands from each other. This can be done by characterizing the length as a irregular esteem to speak to the cloud environment's add up to workload. The length parameter is an fundamental input in the try to decide the stack for each Virtual Machine. Based on this parameter, the Time to Total demands in each VM can be recognized. Based on this, we can decide if there's a infringement in SLA.

3) Parameter as it speaks to SLA; if the Time to Total demands surpasses the Due date, we can recognize that there is a infringement in the SLA.

Sample values that make up the workload

TABLE3.Resultswereobtainedfor4VMswith10to40 tasks

Number of cloudlets	Average Makespan (ms)	Average Execution Time (ms)	Resource Utilization (%)
10	303.5410115	251.6623927	83
15	430.8096146	361.2242922	84
20	512.9966944	407.6082266	79
25	577.6690028	423.5005818	73
30	754.0924138	559.1548208	74
35	869.503937	650.4350047	75
40	896.0297375	638.7887158	71

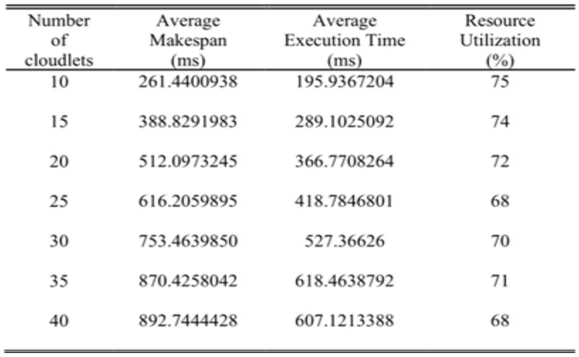


In this area, the execution of the proposed LBA calculation was recorded by taking three diverse test cases:(1)2VirtualMachines with 10to40 cloudlets; (2)4VirtualMachines with 10to40 cloudlets; (3) 6 Virtual Machines with 10 to 40 cloudlets. The increase of these factors in

V. RESULTS COMPARISON

To analyze the comes about of the later inquire about calculation and the proposed calculation. The primary parameter considered for comparison in this investigate is Time span Time. The primary objective of the proposed Stack Adjusting calculation is to upgrade the utilization and assignment of cloud assets In the above table, it contain number of cloudlets, average time span, average execution time resource utilization. The primary parameter considered for comparison in this investigate is time span time. This area gives a common comparison of the existing related work and the proposed calculation in this investigate. While it may additionally now not reflect on consideration on server fitness or authentic load, it serves as a reasonable answer for simple load distribution requirements.

Parameter as its peaks to SLA; if the Time to Total demands surpasses the Due date, we can recognize that there is a infringement in the SLA. All the cloud let sought to have irregular values to separate the client demands from each other. This can be done by characterizing the length as a irregular esteem to speak to the cloud environment’s add up to workload.



Number of cloudlets	Average Makespan (ms)	Average Execution Time (ms)	Resource Utilization (%)
10	261.4400938	195.9367204	75
15	388.8291983	289.1025092	74
20	512.0973245	366.7708264	72
25	616.2059895	418.7846801	68
30	753.4639850	527.36626	70
35	870.4258042	618.4638792	71
40	892.7444428	607.1213388	68

FIGURE 6.Average Resource Utilization for 40 tasks.

Minimize the time taken to plan a assignment for moving forward the execution of the cloud applications. This area gives a common comparison of the existing related work and the proposed calculation in this investigate

VI. CONCLUSION&FUTUREWORK

Weighted Round Robin Load balancing is a imperative method for distributing community visitors correctly through out a couple of servers. It affords a easy and easy-to-implement technique for making sure most reliable aid utilization and excessive availability. By cyclically assigning incoming requests to servers in a sequential manner, Round Robin helps forestall overload on any single server, facilitating fault tolerance and load distribution. While it may additionally now not reflect on consideration on server fitness or authentic load, it serves as a reasonable answer for simple load distribution requirements. However, for greater complicated scenarios, superior load balancing algorithms can also be preferred. Finally, Round Robin Load balancing stays a treasured device in the arsenal of community administrators.

REFERENCE

[1] H.Shukur,S.Zeebaree, R.Zebari,D.Zeebaree, O. Ahmed, and A. Salih,‘Cloud computing virtualization of resources allocation for distributed systems,’’ *J.Appl.Sci.Technol. Trends*, vol.1,no.3,pp.98–105,Jun.2020, doi:10.38094/jastt13 31.

[2] I. Odun-Ayo, M. Ananya, F. Agono, and R. Goddy-Worlu,**‘**Cloud computing architecture: A critical analysis,’’ in *Proc. 18th Int. Conf. Comput. Sci. Appl. (ICCSA)*, Jul. 2018, pp. 1–7, doi: 10.1109/ICCSA.2018.8439638.

[3] A. Jyoti, M. Shrimali, and R. Mishra, ‘‘Cloud computing and load balancing in cloud computing-survey,’’in*Proc.9thInt.Conf.Cloud Comput.,DataSci.Eng.(Confluence)*, Jan.2019, pp. 51–55, doi:10.1109/confluence.2019.8776948.

[4] S.H.H.Madni,M.S.AbdLatiff,M.Abdullahi, S. M. Abdulhamid, and M. J. Usman, ‘‘Performance comparison of heuristic algorithms for task scheduling in IaaS cloud computing environment,’’ *PLoS ONE*, vol. 12, no. 5, May 2017, Art. no. e0176321, doi: 10.1371/ journal.pone.0176321.

[5] M. Adhikari and T. Amgoth, ‘‘Heuristic-based load-balancing algorithm for IaaS cloud, **’***Future Gener.Comput.Syst.*,vol.81,pp.156–165,Apr. 2018, doi: 10.1016/j.future.2017.10.035.

[6] B.SinghandG.Singh,‘‘A study on virtualization and hypervisor in cloud computing,’’ *Int. J. Comput. Sci.MobileAppl.*, vol.6,no. 1, pp.17– 22, 2018.

[7] M. Kumar, S. C. Sharma, A. Goel, and S. P. Singh,**‘**Acomprehensive survey for scheduling techniques in cloud computing,’’ *J. Netw. Comput. Appl.*, vol. 143, pp. 1–33, Oct. 2019, doi: 10.1016/j.jnca.2019.06.006.

[8] F. Zabini, A. Bazzi, B. M. Masini, and R. Verdone,**‘**Optimal performance versus fairness tradeoff for resource allocation in wireless systems,’’*IEEE Trans. Wireless Commun.*, vol. 16,no.4,pp.2587–2600,Apr.2017,doi:10.1109/TWC.2017.2667644.

[9] M. Kumar and S. C. Sharma, ‘‘Dynamic load balancing algorithm to minimize the makespan time and utilize the resources effectively in cloud environment,’’*Int.J.Comput.Appl.*, vol.42,no. 1,pp.108–117,Jan.2020,doi:10.1080/1206212X.2017.1404823.

[10] G. Patel, R. Mehta, and U. Bhoi, ‘‘Enhanced load balanced min-min algorithm for static meta task scheduling in cloud computing,’’ *Procedia Comput. Sci.*, vol. 57, pp. 545–553, 2015, doi: 10.1016/j.procs.2015.07.385.