

Research Article

An Efficient Technique for Virtual Machine Clustering and Communications Using Task-Based Scheduling in Cloud Computing

C. Saravanakumar ¹, M. Geetha ², S. Manoj Kumar ³, S. Manikandan ⁴, C. Arun ⁵
and K. Srivatsan ⁶

¹Department of Information Technology, St. Joseph's Institute of Technology, OMR, Chennai, Tamil Nadu 600119, India

²Department of Computer Science and Engineering, School of Computing, College of Engineering and Technology, Faculty of Engineering and Technology, SRM Institute of Science and Technology, Vadapalani, Chennai, Tamil Nadu, India

³Department of Information Technology, Karpagam College of Engineering, Coimbatore, Tamil Nadu 641 032, India

⁴Department of Information Technology, K. Ramakrishnan College of Engineering, Trichy, Tamilnadu 621 112, India

⁵Department of ECE, R.M.K College of Engineering and Technology, Chennai, Tamil Nadu 601 206, India

⁶School of Electronics Engineering, VIT, Chennai, Tamil Nadu 600127, India

Correspondence should be addressed to C. Saravanakumar; saravanakumarc@stjosephstechnology.ac.in

Received 20 February 2021; Revised 6 June 2021; Accepted 1 July 2021; Published 19 July 2021

Academic Editor: Pengwei Wang

Copyright © 2021 C. Saravanakumar et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Cloud computing models use virtual machine (VM) clusters for protecting resources from failure with backup capability. Cloud user tasks are scheduled by selecting suitable resources for executing the task in the VM cluster. Existing VM clustering processes suffer from issues like preconfiguration, downtime, complex backup process, and disaster management. VM infrastructure provides the high availability resources with dynamic and on-demand configuration. The proposed methodology supports VM clustering process to place and allocate VM based on the requesting task size with bandwidth level to enhance the efficiency and availability. The proposed clustering process is classified as preclustering and postclustering based on the migration. Task and bandwidth classification process classifies tasks with adequate bandwidth for execution in a VM cluster. The mapping of bandwidth to VM is done based on the availability of the VM in the cluster. The VM clustering process uses different performance parameters like lifetime of VM, utilization of VM, bucket size, and task execution time. The main objective of the proposed VM clustering is that it maps the task with suitable VM with bandwidth for achieving high availability and reliability. It reduces task execution and allocated time when compared to existing algorithms.

1. Introduction

Cloud computing is a service-oriented architecture technique that uses virtualized resources to perform computational tasks. The cloud has a set of resources that are offered as a means of service. Cloud services are classified as Software as a Service (SaaS), Infrastructure as a Service (IaaS), Platform as a Service (PaaS), etc. The services are deployed in different models to meet customer requirements. They are classified as private cloud, public cloud, community cloud, and hybrid cloud. Private cloud resources

are shared within the organization, in which the services are shared outside the organization as a public cloud. Community cloud is a type of cloud, in which services are shared between the service providers of the same category. A combination of two or more deployment models, called a hybrid cloud, provides the customer a service. Cloud services are modeled by mapping the virtualization layer to the appropriate VM. VMs are selected from the VM list and then mapped to the respective request generated by the user of the cloud service. The cloud consists of a heterogeneous host in a data center that maintains a mobile resource-based access

environment. VM is a type of access that leads to a performance problem seen in the areas of the battery life and energy consumption. The entire performance factor in green computing is used for overcoming this problem [1]. Mobile cloud computing (MCC) is a mobile resource sharing service that allows mobile devices to have access to the appropriate cloud service. It has faced challenges in terms of scalability, such as computational storage services and other different services [2].

Mobile computing over cloud has the ability to target parameters such as traffic, quality, and customer demand. Traditional static cloud and dynamic cloud are compared to analyze the workload. The static cloud allows users to access infrastructure services with a specific configuration, while the dynamic cloud provides an agile response method to update the resource configuration. Dynamic cloud has a variety of wireless nodes with device-to-device connectivity for the achievement of a better utilization of the channel and traffic [3]. VM contention: IaaS has created problems in the area of performance. This problem is overcome through the implementation of the data center as various ranges such as single server with virtualization, single mega data center, and multiple geo-distributed data centers [4]. A researcher engaged in cloud suffers from issues that include energy consumption in data centers. The data center is a key element in the cloud that handles all kinds of resources needed in the computing environment. In the cloud, there are two types of approaches that are related to hardware and software. These approaches require reduced power consumption in cloud resources without any situation of service unavailability [5]. Cloud resources are multiplexed over VM servers designed to host cloud services in large data centers.

VM migration is the process of migrating from one location to another leading to the performance problem arising as a result of inference and the cost of the operation. iAware imposes the multiresource supply demand model for minimizing the inference during the migration [6, 7]. Intelligent transportation systems (ITS) are used in the vehicle cloud computing (VCC) architecture, which consists of two-layered components such as the central cloud server and the remote system control (RSC). RSC is a remote administration manager for monitoring and managing distributed system elements such as network communication lines. RSC uses two local server and road side unit (RSU) components. If the vehicle travels from one position to another, the VM of one RSU is moved to another RSU. It requires continuous service response over the automated vehicle control using VCC [5]. There are two main disputes seen in the intelligent transportation systems (ITS), namely, efficiency in traffic and energy, quality, and productivity. The data collected from various sensors are overcome by using the parallelized fusion technique. This technique follows the Dempster-Shafer theory with four components namely, sensor input, bootstrapping, hierarchical fusion, and state output [6, 8]. Cyber-ITS is a system that has data division, scheduling, and efficient support through the use of a generic methodological framework. There are two types of functions carried out by the system, namely, data-centric and operation-centric transformations. This model uses high-performance computer design with region-based decoupling capability [9, 10].

In this method, a digital map of the global positioning system data is processed in a parallel manner using quad-tree-based domain decomposition technique. These data are divided into different subdomains with quad structure [11]. Multi-CPU VM scheduling and virtual CPUs (VCPUs) scheduling have been carried out due to the availability of various virtualization techniques in the cloud computing domain [10]. The existing problems have been analyzed based on the performance parameters, which improve the efficiency in the cloud service deployment. The main objective of the proposed technique is to identify the suitable VM based on the bandwidth and requesting task of allocating any issues in the performance of cloud to the task.

The VMs are configured in an isolated fashion, which suffers from repetitive booting of the respective volumes with a limited period of delivery. This problem is solved by a cluster management approach based on a docker container with a diverse configuration [12]. The traditional placement of the docker container and the VM is carried out separately, so that it is implemented using the container VM-PM model [13]. The Internet of Things plays a key role in the processing of real-time data from hardware devices that generate large quantities of data. These data are stored in a large data center with big data analysis methods. If the data are huge, a huge number of servers would be used to store the received data. It faces an expensive problem that is solved by using a cabinet-based tool called ProCon [14]. Virtualization technology offers the benefits of the physical server operating on several computers with different resources. Virtual storage eliminates read and write delays with high IO efficiency. Virtual disks are connected to VMs for processing and storing the user data via synchronization [15]. Amazon cloud provider provides various kinds of services to the end user in reliable and secure computing capacity. AWS offers the Elastic Compute Cloud (EC2) with different versions of VM and resources. The proposed method used EC2 instances as a reference for further analysis.

The rest of the paper is organized as follows. Section 2 depicts the architecture of the proposed system. Section 3 describes the phases of the proposed system. Section 4 presents the analysis of various metrics. Section 5 describes the proposed VM clustering process. Section 6 provides the experimental evaluation. Section 7 signifies a comparison of the various clustering algorithms. Section 8 presents the conclusions and indications for future work.

2. Model of the Proposed System

Existing VM management techniques configure a VM to cloud workloads based on the energy parameters, but they suffer from a resource wastage problem in the data center. The proposed model groups the tasks based on the VM types with bandwidth parameters using classification methods. The VM types are customized based on the resource availability in the cloud. The objective of the proposed model is to map a task to the correct VM by considering various processes such as resources mapping and classification [16]. The cloud requesting tasks are classified based on the metrics such as total number of queues, request count, API response

count, dispatch rate of the queue, maximum size of the task, actual and scheduled execution time, delay, and task retention time. These metrics are collected using CloudWatch monitoring service in AWS cloud. These metrics are exported and used for analyzing the proposed model.

Figure 1 shows the architecture of the proposed system. The customer makes a request for the resource from the cloud based on their current requirement. The request is considered as a task executed in a VM. There are various tasks generated by the customer identified based on various perspectives. The identified task is classified based on the reliability parameter in order to achieve a high performance. Bandwidths are selected for the classified task so that suitable VM is allocated to the requesting task. There are various bandwidths available in the requested task, which are identified and classified to enable mapping the suitable VM for service delivery. The bandwidth-VM mapping section selects the VM from the VM clusters, which, in turn, are selected by the VM selector. The VMs are clusters based on the task and bandwidth meant for providing the services without any interruption or delay. They are clustered in the respective VMs based on the type of task requested, and the hypervisor eventually provides the link between the VMs and the physical system.

Figure 2 is the sequence diagram of the proposed system. It denotes how the incidents are actually related to each other. The activity diagram shows how the process starts and terminates, the various state changes, and activities that take place between these state changes. The first phase is the login module, where the user is authenticated to access the system. The username and password are provided. These further provide access to the cloud home page, which consists of all the main functionalities. This module allows only the authorized user to log in to the system. It provides authentication and access only to the authorized user and allows the user to select any of the options that include creation of a virtual machine, viewing the existing machines, making task-based separation, and viewing the usage report. The first option in the module is helpful for the user in creation of the module by just entering the values for the new VM. Instead of typing commands in the terminal to create a virtual machine, this module helps us create a VM by simply entering the values.

The second option in the module is for viewing the existing VMs, thereby enabling the user to view the type of VM created and used. Each VM has different specifications. This module can view the existing VMs with their specification, the operating system of the VM. The third option in the module is task-based separation. This phase helps separation of the task from the cloud user with the help of this option. It separates the task in terms of CPU, memory, and IO. The fourth option in the module is VM clustering. Groups of VMs with similar features are clustered with the help of this option. The final option is the report, where the user can view a detailed usage of the VMs. In this module, the user has the ability to create new VMs for use based on requirements. The major use of this module is that, unlike in the normal VM creation where the user needs to go to the terminal and type commands,

the user just needs to type values for the VM creation. A name for the VM and the various parameters such as the number of CPU, vCPU, the disk name, NIC, and SSH are entered, where all these parameters are required for the creation of the VM template.

The template, which is created, is instantiated by providing the disk name and the VM name, thereby creating the VM. This module is used for helping the user in making separation on the VM based on the type of task in each request. There are three major classifications on the task that include memory-based, CPU-based, and IO-based classification. The CPU-based classification is meant for the user who requires high processing speed during the processing of the input file. In this, the input file is only processed, but not stored. The memory-based classification is for the user to have better memory space. In this, the input file is only stored in the memory and not processed. The IO-based classification is used for the user to have a responsive VM. In this option, the output is only generated for the input file and not stored in the memory.

The mixed option is used for the user in the classification of a task with more than one type. Selection of this type helps the user in the selection of either of the two types of classification method for each task. This module is used for allowing the user to view the existing VMs that are created in the system. This view option is in a tabular view, where all the existing VMs are listed along with the specifications of each VM. The module is highly useful for getting knowledge of all the VMs that have already been created in the system. All the active VMs can be viewed with the help of this module. The basic uses of this module provide a view of the existing VMs and differentiate active VMs. The module displays the information of the existing virtual machines such as their user, ID, group, and name.

3. Mathematical Analysis of VM Clustering

Normally, the VM is created from the physical machine (PM) through the use of a virtual machine manager (VMM). There are various VMs maintained and controlled by the Domain_0, which acts as a supervisor performing all mapping operations with maximum availability among other domains. Five phases of the VM states exist in the cloud, namely, VM Creation State, VM Running State, VM Suspend State, VM Resume State, and VM Destroy State. The distribution of VM is represented in the normal distribution with $[-\infty, \infty]$. This limit is unbounded, and so, this is represented with the bounded manner by changing the limit to $[-M, M]$. The VM distributions are expressed in (1). VMs are considered as VM sets called $VM = \{VM_1, VM_2, \dots, VM_n\}$ of size N . The requesting tasks are allocated into specific VM based on the task categories, which are represented as $Task = \{T_1, T_2, \dots, T_n\}$. The allocation of the VM and tasks is specified as $Allocation = \{T_1 \rightarrow VM_1, T_2 \rightarrow VM_2, \dots, T_n \rightarrow VM_n\}$ with the relevant size S . Normal distribution is done based on the clustering with a cluster size of 2, and then, it increases until stopping conditions are reached. It is represented as N_1, N_2, \dots, N_n with bounded distribution in equations (2)–(4).

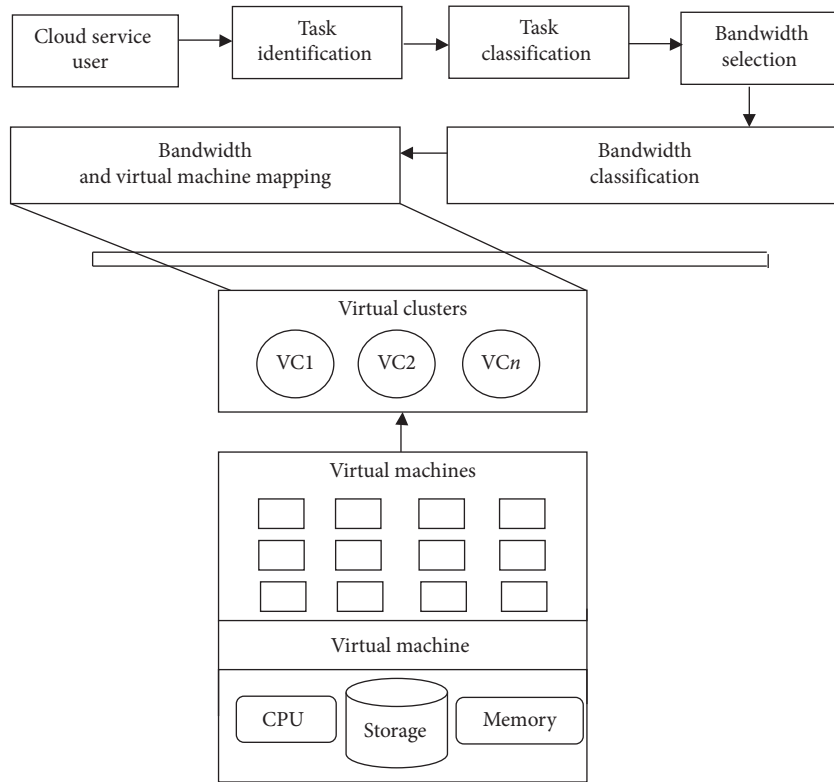


FIGURE 1: Architecture of the proposed system.

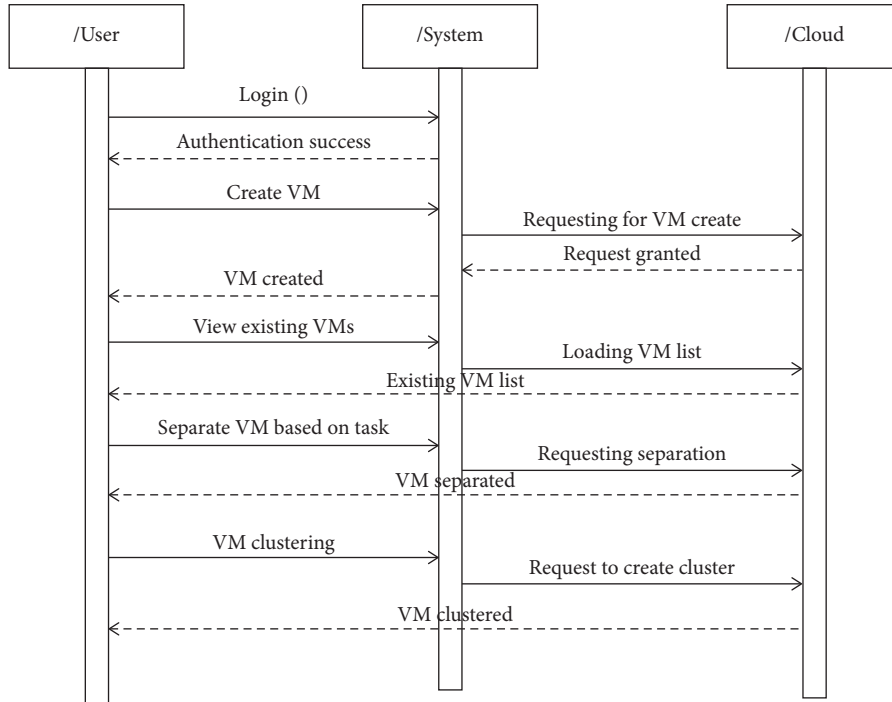


FIGURE 2: Phases of the proposed VM clustering process.

$$\text{Distribution of VM} = \left(\frac{1}{\sigma_{\text{Active VM}} \sqrt{2 \prod_{VM}}} \right) \int_{-\alpha}^{\alpha} (X_{VM} - \mu_{\text{Active VM}})^r \left(\frac{e^{-(X - \mu_{\text{Active VM}})}}{2\sigma_{\text{Active VM}}^2 \cdot S} \right), \quad (1)$$

$$= \frac{1}{\sigma_{\text{Active VM}} \sqrt{2 \prod_{VM}}} \int_{-\alpha}^M \left[(X_{VM} - \mu_{\text{Active VM}})^r \int_M^{\alpha} (X_{VM} + \mu_{\text{Active VM}})^s \right] \cdot S, \quad (2)$$

$$= \frac{1}{\sigma_{\text{Active VM}} \sqrt{2 \prod_{VM}}} \int_{-\alpha}^M [N_1 + N_2] \cdot S, \quad (3)$$

$$= \frac{1}{\sigma_{\text{Active VM}} \sqrt{2 \prod_{VM}}} \int_{-\alpha}^M [N_1 + N_2 + \dots + N_n] \cdot S. \quad (4)$$

The number of clusters is either odd or even. So, it is handled based on the conditions of the VM distribution. There are two cases to follow by performing VM clustering process.

Case 1. If the number of cluster selections is odd, i.e., $r = 2n + 1$, the VM distribution formulated is represented as

$$\mu_{2n+1} = \frac{(2^{(2n+1/2)} \sigma_{\text{Active VM}}^{2n+1})}{\sqrt{\prod_{VM}}} \int_{-M}^M [N_1 + N_2 + \dots + N_n] \cdot S. \quad (5)$$

Case 2. If the number of cluster selections is even, i.e., $r = 2n$, the VM distribution formulated is represented as

$$\mu_{2n} = \frac{(2^{2n/2} \sigma_{\text{Active VM}}^{2n})}{\sqrt{\prod_{VM}}} \int_{-M}^M [N_1 + N_2 + \dots + N_n] \cdot S. \quad (6)$$

A VM clustering process carried out is based on the categorization of VM as active and inactive VM (idle VM).

This is done by considering the processor states, which are represented in equations (7) and (8) respectively,

$$\text{VM}_{\text{Active State}} = \begin{cases} \text{VM}_{\text{Active}} \in \text{VMList}, & \text{state} == \text{active}, \\ \phi, & \text{otherwise.} \end{cases} \quad (7)$$

$$\text{VM}_{\text{Inactive State}} = \begin{cases} \text{VM}_{\text{Inactive}} \in \text{VMList} & \text{state} == \text{idle}, \\ \phi, & \text{otherwise.} \end{cases} \quad (8)$$

PM and VM mapping is done by considering the utilization of the CPU and vCPU in both the guest and host environments by finding mean value. This is shown in

$$\text{PM VM mapping ratio} = \text{mean} (\text{max PM utilization} + \text{max VM utilization}). \quad (9)$$

The lifetime of the VM is assessed based on the time taken from creation to process; the completion stage, which is always greater than one, is described in

$$\text{VM life time} = \begin{cases} \text{time (VM creation + VM execution + VM destroy)}, & \text{time} > 0, \\ \phi, & \text{otherwise.} \end{cases} \quad (10)$$

Data are stored in the cloud bucket by performing read and write operations with scalable and flexible properties by accessing in a ubiquitous manner. The average bucket count

is calculated using the bucket count of PM, as well as VM, is shown in

$$\text{average bucket count} = \left\{ \text{count} \left[\frac{\text{PM bucket count} - \text{VM bucket count}}{\text{total bucket count}} \right] \right\}. \quad (11)$$

VM clusters depend on various factors, which include VM states, utilization of CPU in both VM and PM, bucket

size, lifetime of the VM, and number of cores. This process is carried out based on all clusters, which are shown in

$$\text{VM clustering}_i = \bigcup_{i=1}^n \text{Count}[\text{Active VM}_i + \text{CPU utilization}_i + \text{VM bucket}_i + \text{VM life time}_i + \text{core count}_i]. \quad (12)$$

4. Analysis of Various Performance Metrics

4.1. VM Parameter-Level Analysis. There are different features considered while clustering the VM for a good maintenance of reliability in the cloud. The capabilities available in the migration process need to keep the copy of the VM at the original source end. Two types of copying process happen in the VM, namely, precopying and post-copying process. The cluster process uses these techniques for achieving better availability. The CPU is halted during the migration process in the source machine, as well as the destination machine. Delta-based compression of VM has more number of dependent VMs with respective references for future VM consolidation at the target machine. Data-level compression is used for getting the reduction in the contents related to the VM at the source machine. The workload is classified as synthesis-based and idle-based workload. Synthesis-based workload is a preallocated load assigned before the VM migration. Idle-based workload is assigned to the task based on the demand in nature. VM size depends upon various factors, namely, the number of vCPU, memory size in GB, the bandwidth of the memory in GB/s, the frequency of the CPU in GHz, single and all core frequencies in GHz, performance of remote memory access, temporary storage in GB, number of data disks, and number of Ethernet NICs. Initially, the target machine memory is considered as dirty pages. Later, it is replaced with the respective contents after migration. Resource availability of the target machine is also addressed during the migration of the VM. Table 1 shows the VM features in multiple perspectives.

4.2. User Task Classification-Level Analysis. Cloud user tasks are classified based on the factors that include the name of the user base (UB), regions and the number of requests per single user, requested task size in bytes, duration in terms of peak hours in GMT, and the average number of peak users offline and online. Table 1 shows the classification of tasks with user parameters of different sizes. These tasks are mapped with the VM parameters for achieving high reliability. Cloud regions maintain the resources for users with the corresponding user base. The requesting size and related task size are analyzed in order to find average number of users in both peak and nonpeak hours of the particular region. The comparison of task classification is shown in Table 2. Heterogeneous VMs are available for handling the user-requesting task with different categories. Virtual machine manager schedules the task to respective VM. A large number of requesting tasks to be handled have arrived in the clouds that are performed by a large number of VMs. It is proposed that the number of virtual machines be increased on the basis of number of tasks requested.

Cloud region collects the user requests from the user base (UB) and identifies the request size. The tasks are allocated to the respective resources in the cloud region based

on the peak and nonpeak hours through autoscaling technique. It handles the maximum number of VM based on the demand. Figure 3 shows the total number of tasks on the VMs over the cloud region.

4.3. Physical Machine Analysis of Data Centers. The physical machine in the data centers has resources that include memory size in MB, storage in MB, number of processors, the speed of the processor, and VM policy such as time shared and space shared. All the resources are virtualized and also shared by multiple VMs for task execution. Table 3 shows the physical resources at the data centers with various performance parameters. The process speed is represented as instruction per second (IPS). These physical resources need VM policy for executing tasks, which are either time shared and space shared. The proposed analysis is time shared because of CPU intensive operation carried out for high availability.

4.4. Virtual Machine Analysis of Data Centers. Virtual machines are created with different parameters such as the name of the data centers, regions of the data center, the architecture of the VM, the platform of the VM, type of VMM, the cost of the resources, and physical hardware units. Data centers perform various operations, which include migration process in either the same data center with migration or different data centers with migration. Linux with X86 architecture is used for deciding the required number of physical machines at the data centers in different regions. Table 4 describes the VM attributes for task allocation and execution. Cloud customer needs higher bandwidth because of the lack in the current bandwidth level. Cloud providers provide sufficient bandwidth in order to retain the customers. Table 5 provides the bandwidth level of various domain users for effective utilization.

4.5. Delay and Bandwidth Matrix Analysis. Cloud services are deployed in various data centers as regions. Delays seen between the regions are compared for efficiency. The main objective of this delay analysis is to identify the fast response with minimum response rate. The bandwidth matrix provides an efficient route between the requests in the shortest path with maximum availability. Delay in the network during the transfer of jobs across various regions is shown in Table 6. The delay between the same region is also minimum, whereas the delay of the different regions is maximum with respect to the distance between the regions. The same region transfer rate is fast when compared to different regions, so it depends on the bandwidth level and delay, which is shown in Table 7. Figure 4 indicates the response rate of various cloud regions in ms (Milliseconds). Response time of various user bases is measured in three levels, namely, average, minimum, and maximum. The CloudAnalyst model is used for

TABLE 1: VM parameters.

VM parameters	Description
Migration algorithm (capability)	0: original precopy 1: CPU-based halting 2: delta-based compression 3: data-based compression 4: postcopy
Workload type	0: synthetic-based workload 1: idle-based workload
VM size	Size of VM at data center
Page dirty rate	Total number of modified pages in the VM
Working set size	The size of dirty pages
Page transfer rate	VM migration rate from one cluster to another cluster
CPU utilization of VM	Physical CPU utilization based on VM
Network utilization of VM	Bandwidth utilization of the VM
CPU utilization on host	Destination of VM clusters' CPU utilization
Memory utilization on host	Destination of VM clusters' memory utilization
IO utilization on host	Destination of VM clusters' IO utilization

TABLE 2: User task classification and comparison.

Name of the user base	Regions	Requests per user per	Task size per request (bytes)	Peak hours starts (GMT)	Peak hours end (GMT)	Avg peak users	Avg off-peak users
UB1	0	60	100	3	9	1000	100
UB2	1	60	90	4	10	1000	100
UB3	2	40	75	5	8	1000	100
UB4	3	55	80	7	11	1000	100
UB5	4	30	85	8	9	1000	100
UB6	5	45	100	6	12	1000	100

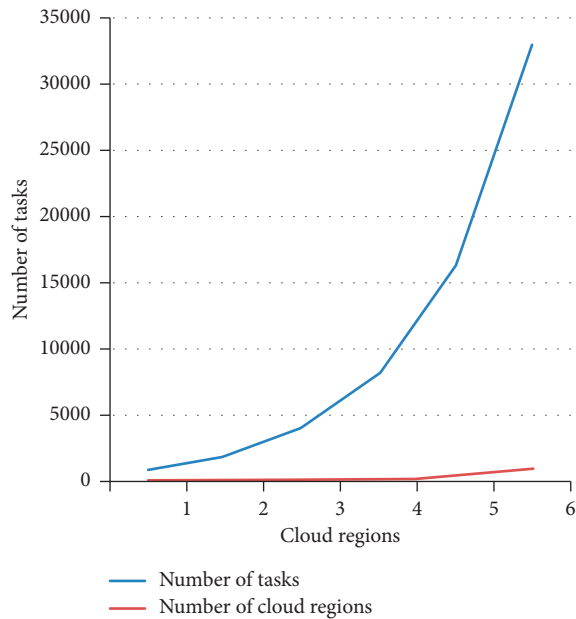


FIGURE 3: Comparison of cloud regions and user tasks.

TABLE 3: Data center analysis for the physical machine.

ID	Memory (MB)	Storage (MB)	Available bandwidth	Number of processors	Processor speed (IPS)	VM policy
0	204800	10000000	1000000	4	10000	Time shared
1	254800	20000000	1000000	4	10000	Time shared
2	304900	30000000	1000000	4	10000	Time shared
3	354500	60000000	1000000	4	10000	Time shared
4	224800	80000000	1000000	4	10000	Time shared

TABLE 4: Data center analysis for virtual machines.

Name	Region	Arch	OS	VMM	Cost per VM \$/hr	Memory cost \$/s	Storage cost \$/s	Data transfer cost \$/GB	Physical HW units
DC1	0	x86	Linux	Xen	0.1	0.05	0.1	0.1	5
DC2	1	x86	Linux	Xen	0.2	0.05	0.3	0.25	1

TABLE 5: Comparison of bandwidth utilization.

Domain of service providers	Current bandwidth utilization (%)	Expected bandwidth level (%)
Banking and insurance	61	19
Telecommunications	73	10
IT services	39	16
Cloud services	62	4
Education services	36	4
Government services	41	10

the proposed analysis with delay and bandwidth allocation of the task execution. The custom bandwidth and delay matrix are specified on internet characteristics option with various regions.

4.6. Analysis of VM Clustering. VM clustering is the process of grouping various VMs using virtual networks for achieving high availability of cloud resources. This clustering process is done in a source machine, as well as a target machine. There are two important concepts considered for good accuracy, namely, preclustering and postclustering. A preclustering process occurs at the source end, whereas postclustering occurs at the target machine end. Preclustering interconnects the VM along with the state of the processor, data, and VM-related parameter. These VMs then migrate to the target machine. Postclustering process collects the VMs in order and finds the relationships, forming a cluster by interconnecting the received VMs. This process resumes all the states of the VM and related elements into original state. Table 8 shows the various VM clusters with the required parameters. There are two types of VMs grouped, namely, active and inactive. Utilization of the CPU plays a vital role during the clustering process. VM lifetime is considered in allocating better performance during the clustering process.

5. Proposed VM Clustering

The clustered virtual machines (VMs) are created from a physical machine (PM). The mapping of VM and PM is performed by the hypervisor. The objective of clustering is to execute the requesting task, which is categorized with different parameters. The bandwidth is classified, and the corresponding VMs are mapped. The VMs are clustered in a way of similar categories of VM grouped together. The allocation of VMs is extremely simple and efficient for the execution of a task. There are large numbers of clustered VMs that function as dynamic behavior clustering. The completion time of the VM is analyzed, and clusters of VMs are shown in Figures 5–7 .

TABLE 6: Comparison of the delay matrix.

Regions/region	R0	R1	R2	R3	R4	R5
R0	25.0	100.0	150.0	250.0	250.0	100.0
R1	100.0	25.0	250.0	500.0	350.0	200.0
R2	150.0	250.0	25.0	150.0	150.0	200.0
R3	250.0	500.0	150.0	25.0	500.0	500.0
R4	250.0	350.0	150.0	500.0	25.0	500.0
R5	100.0	200.0	200.0	500.0	500.0	25.0

TABLE 7: Comparison of the bandwidth matrix.

Regions/region	R0	R1	R2	R3	R4	R5
R0	2000.0	1000.0	1000.0	1000.0	1000.0	1000.0
R1	1000.0	800.0	1000.0	1000.0	1000.0	1000.0
R2	1000.0	1000.0	2500.0	1000.0	1000.0	1000.0
R3	1000.0	1000.0	1000.0	1500.0	1000.0	1000.0
R4	1000.0	1000.0	1000.0	1000.0	500.0	1000.0
R5	1000.0	1000.0	1000.0	1000.0	1000.0	2000.0

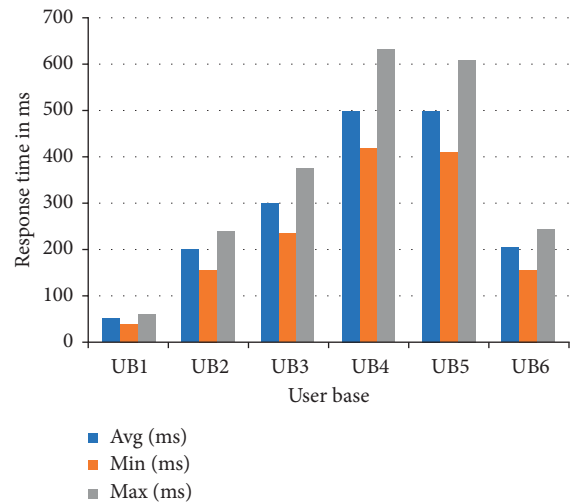


FIGURE 4: Comparison of response time of the cloud region.

TABLE 8: Parameters of VM clustering process.

Clusters	Active VMs	Inactive VMs	Maximum CPU utilization	Average CPU utilization	VM core count bucket	VM memory bucket	VM life time
0	558300	1673700	91.77688	0.72887	8	32	309.8333
1	424500	4254000	37.87926	3.32535	4	32	0.25
2	113310	1133700	0.304368	0.22055	4	32	0.16666
3	0	2591400	98.57342	3.34005	2	4	719.833
4	228300	2298000	82.58144	1.87629	2	4	0.41666

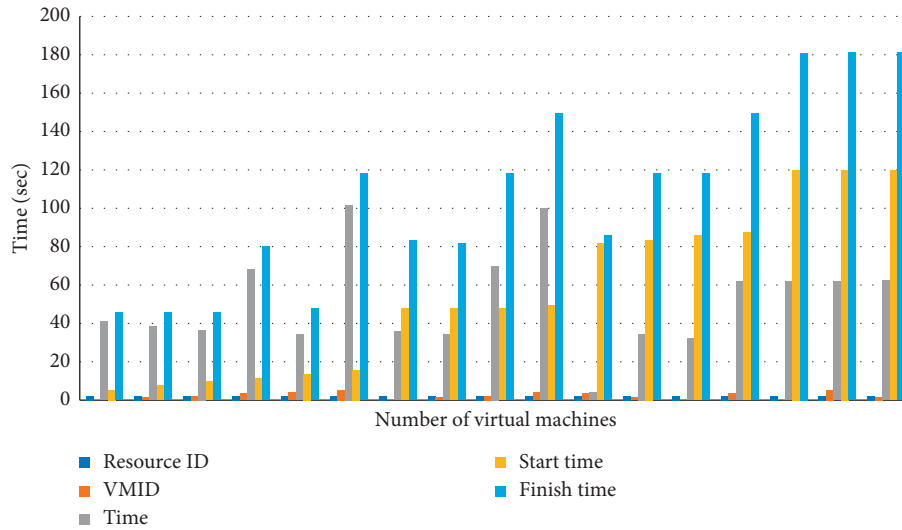


FIGURE 5: Time comparison of VM cluster 2.

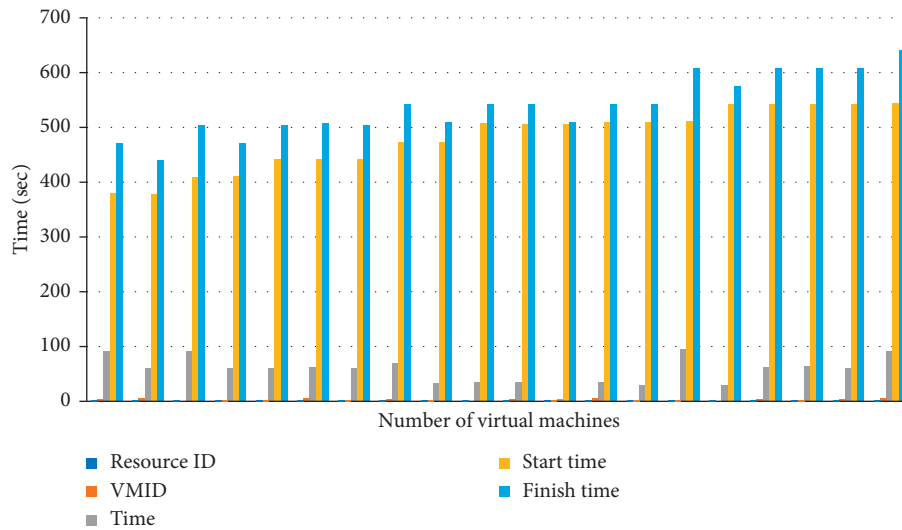


FIGURE 6: Time comparison of VM cluster 3.

6. Experimental Evaluation

Experimental setup is done based on the CloudSim and CloudAnalyst model with a cloudlet for task specification, VM management service, allocation of resources (CPU, storage, bandwidth, and memory), and VM provisioning service. The data center (DC), host, and VM are created with suitable cloud brokers in CloudSim and

analysis of parameters in CloudAnalyst. DC is configured with x86 architecture, Linux OS, Xen Hypervisor, hardware unit, specific region, data center name, and cost of data transfer and resources. The parameter values of the VM configuration are time-shared VM policy, 25 GB of memory, storage size of 1 TB, 1 Mbps of bandwidth, number of tasks of 100, VM size of 10000, and speed of 2.8 GHz.

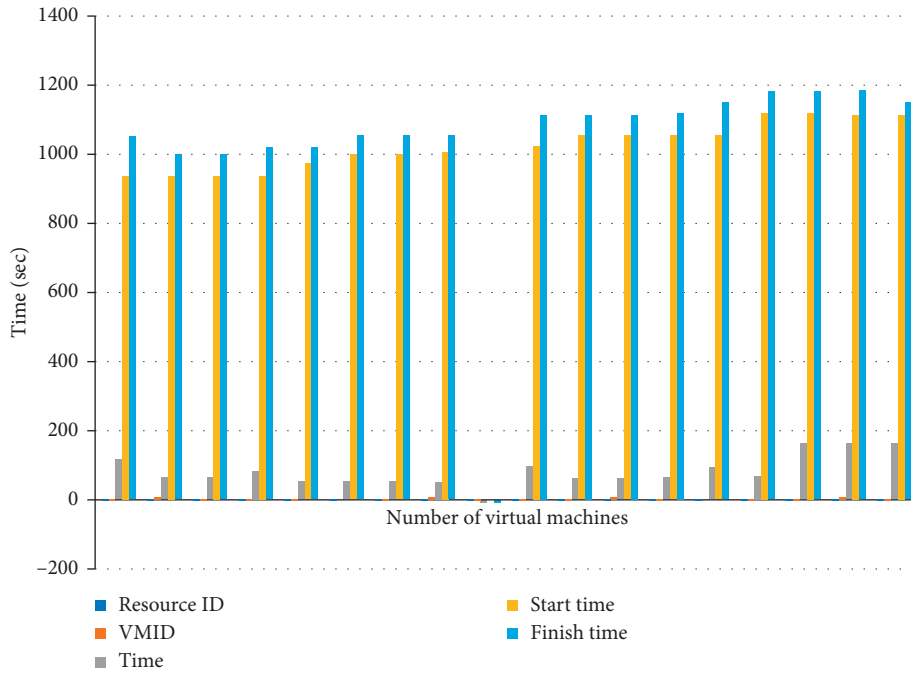
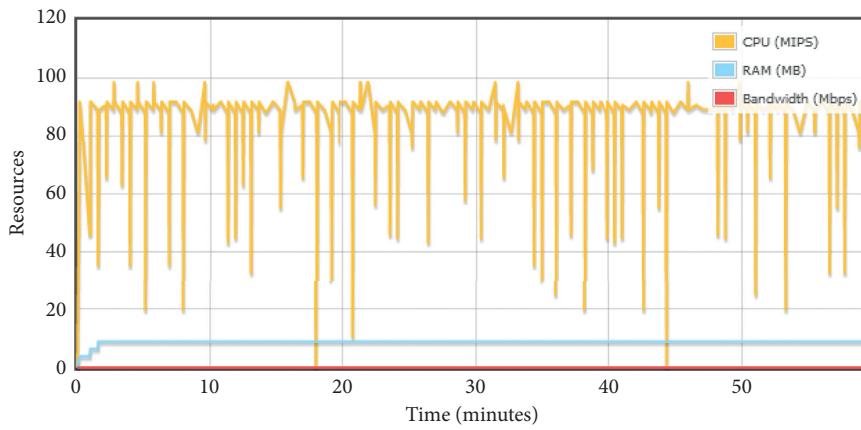
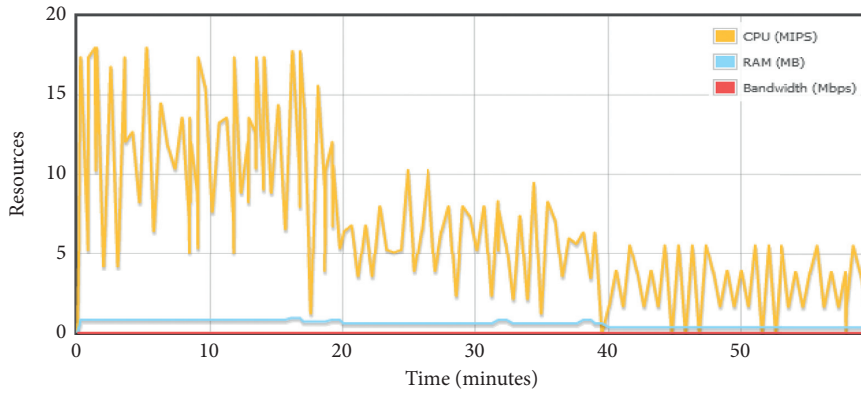


FIGURE 7: Time comparison of VM cluster 4.



(a)



(b)

FIGURE 8: Continued.

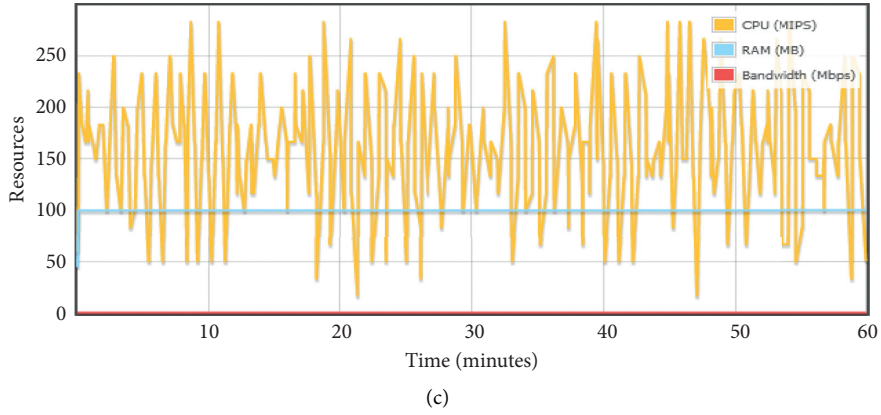


FIGURE 8: Analysis of VM clusters with task execution, (a) VM cluster 1. (b) VM cluster 2. (c) VM cluster 3.

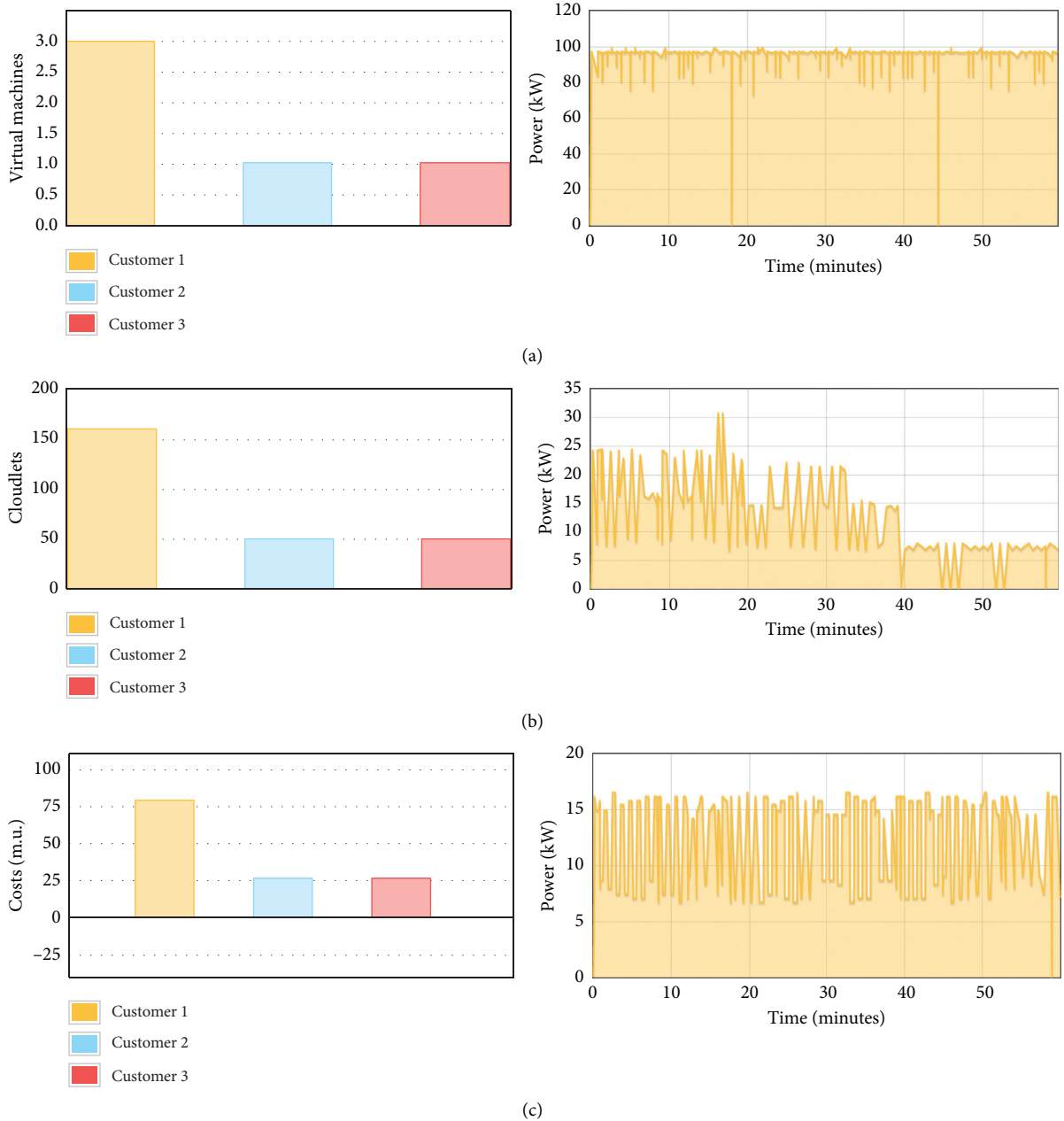


FIGURE 9: Comparison of VM allocation, execution and cost with power consumption. (a) VM allocation vs. power consumption. (b) Task execution vs. power consumption. (c) Data center cost vs. power consumption.

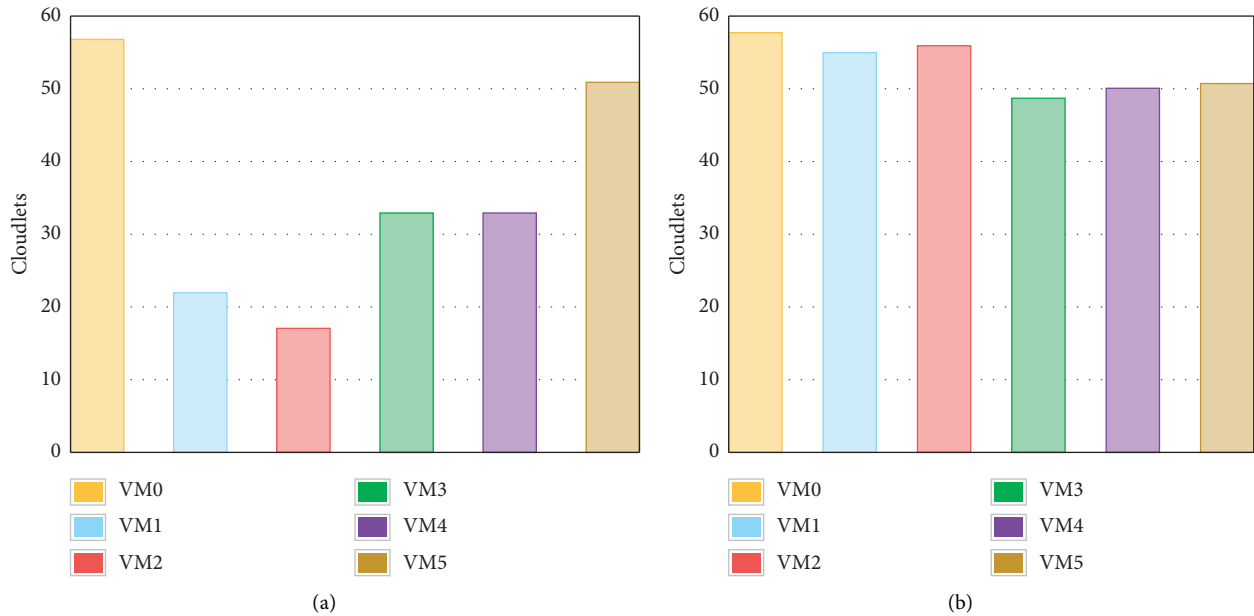


FIGURE 10: Comparison of VM vs. task allocation.

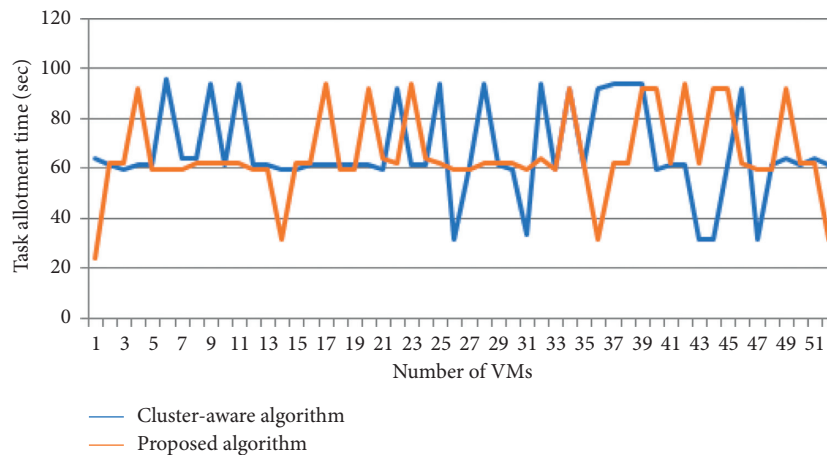


FIGURE 11: Comparison of task allocation time.

Cloud report is a model that is used for the simulation of the cloud services with various supporting packages to enable the analysis of the cloud-computing environment [17]. This mode has meta-data for allocating resources to the data centers such as host, policy, VM with configuration, and cloud broker policies. Analysis of the resource allocation and related consumption of energy is made with different parameters at the data center. The VMs are allocated to the data center based on the task category and analyzed with various configurations [18, 19]. The user requires knowledge of how the resources are utilized with their task at the data center. The execution time of the task from the requesting time to the completion time is also analyzed to provide satisfaction to the customer. Three VM clusters are considered for the proposed evaluation of the algorithm. Three clusters are modeled in the proposed system for executing tasks at the resources of the cloud. Resource utilization is analyzed based on the CPU, RAM, and bandwidth in the VM clusters as shown in Figure 8.

The energy consumption of VM before and after allocation of tasks over the data center is represented in Figure 9. The power consumption is high in cluster 1 and cluster 3 as the number of task assignments to the VM is large when compared to cluster 2.

A comparison of task allocation to the VM is shown in Figure 10. The VMs are created and scheduled by the VM monitor, which is responsible for the task assignment. The tasks, which are considered as cloudlets, are configured with the customer requirement and achieved efficiency of a high order.

7. Results and Discussion

The ATOM-based VM clustering method helps in the evaluation of the performance measurement with parameters like accuracy and reliability with precision, recall, and F-measure as 96.08%, 95.10%, and 95.59%, respectively [20]. The Piccolo system is used for minimizing the traffic that

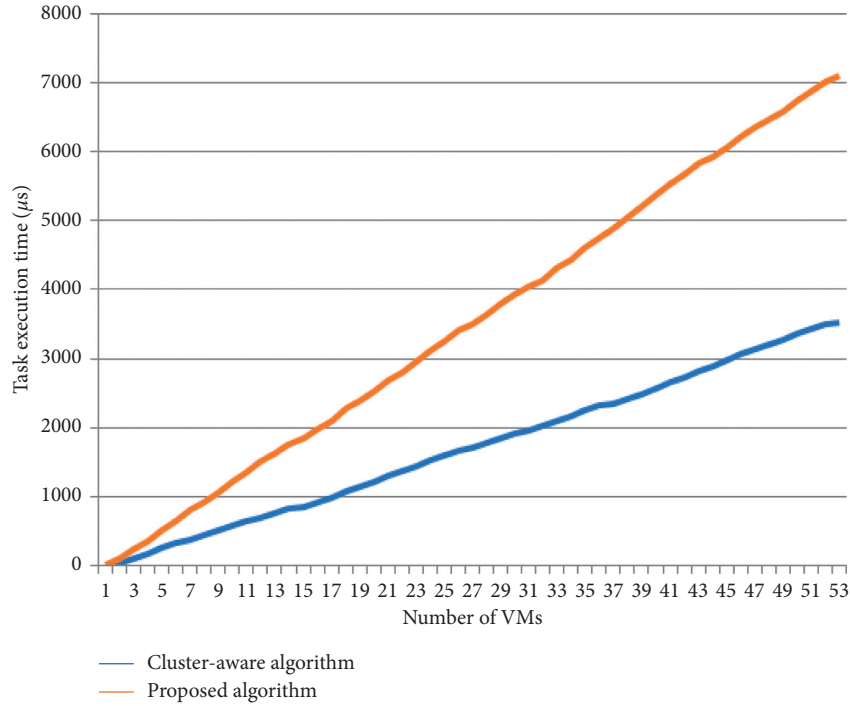


FIGURE 12: Comparison of task execution time.

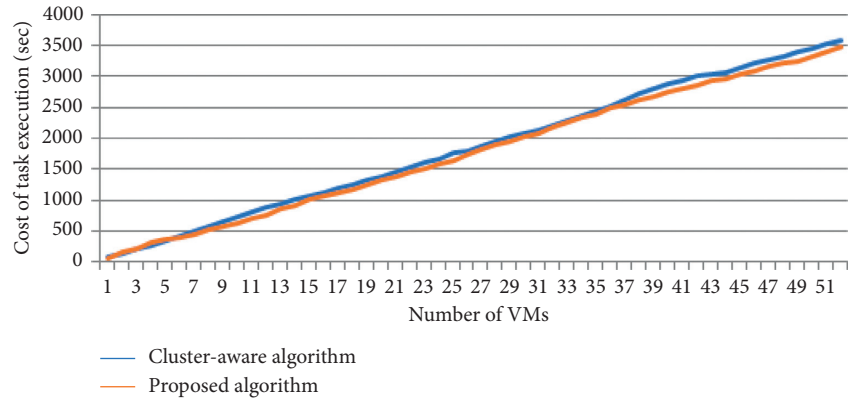


FIGURE 13: Comparison of task allocation time.

occurs during the rollback of the entire VM clusters. This helps obtain less time and network-related overhead. However, it suffers an overload problem due to the handling of entire VM clusters [21, 22]. Cloud radio access network (C-RAN) allocates the VM with low cost and maximum performance but needs examination with constraints like those relating to data and capacity [23]. Cluster-aware VM clustering with two-tier model clusters does the most frequent communication among VMs in the cloud. It contains two stages of the clustering process, namely, host and partitioned oriented. This method is restricted to a homogeneous cloud and does not have application to heterogeneous cloud [24, 25]. ACTor algorithm matches with the historical pattern to the VM clustering process. It uses the passive way of clustering VM with dynamic task and resources and collocated VM [26, 27]. The proposed algorithm

compared with the cluster-aware algorithm based on the factors such as allocation time, execution time, and task completion time of various clusters is shown in Figures 11–13, respectively. The proposed algorithm takes less time for task execution compared to cluster-aware algorithm.

8. Conclusion and Future Work

The focus of the proposed system is on the clustering of VM based on various performance parameters like the type of the requesting task and bandwidth. The requesting tasks are classified into CPU-based, storage-based, and IO-based mixed types. To start with, the requesting task is validated, and then, the tasks are allocated to the task classification process. The task classification process categorizes the task

into different types depending upon the properties, which exist in the task. These tasks are clustered using the clustering algorithm, in which the categories are grouped together. There are two types of clustering process being carried out, namely, preclustering and postclustering. The bandwidth is also clustered based on the task in the task cluster. There are two types of clusters maintained in the proposed technique, namely, the task cluster and the bandwidth cluster with the same features. The VM is classified and mapped to the suitable requesting task for execution. The main objective of the proposed technique is to map the requesting task to a suitable VM, in which the latency in the service handling is minimized with higher efficiency.

Data Availability

The proposed model uses the CloudSim package, which is available at <https://github.com/Cloudslab/cloudsim>. The data are selected for implementation of this model.

Conflicts of Interest

All authors declare that there are no conflicts of interest.

Authors' Contributions

All authors have participated in conception and design, analysis and interpretation of the data, drafting of the article and revising it critically for important intellectual content, and approval of the final version.

References

- [1] Y. Tang, Y. Hu, and L. Zhang, "A classification-based virtual machine placement algorithm in mobile cloud computing," *KSII Transactions on Internet and Information Systems*, vol. 10, no. 5, pp. 1998–2014, 2016.
- [2] M. R. Rahimi, J. Ren, C. H. Liu, A. V. Vasilakos, and N. Venkatasubramanian, "Mobile cloud computing: a survey, state of art and future directions," *Mobile Networks and Applications*, vol. 19, no. 2, pp. 133–143, 2014.
- [3] M. Jo, T. Maksymuk, B. Strykhalyuk, and C.-H. Cho, "Device-to-device-based heterogeneous radio access network architecture for mobile cloud computing," *IEEE Wireless Communications*, vol. 22, pp. 50–58, 2015.
- [4] F. Xu, F. Liu, H. Jin, and V. V. Athanasios, "Managing performance overhead of virtual machines in cloud computing: a survey, state of the art, and future directions," *Proceedings of the IEEE*, vol. 102, pp. 11–31, 2013.
- [5] H. Yao, C. Bai, D. Zeng, Q. Liang, and Y. Fan, "Migrate or not? Exploring virtual machine migration in roadside cloudlet-based vehicular cloud," *Concurrency and Computation: Practice and Experience*, vol. 27, no. 18, pp. 5780–5792, 2015.
- [6] F. Xu, F. Liu, L. Liu, H. Jin, and B. Li, "iAware: making live migration of virtual machines interference-aware in the cloud," *IEEE Transactions on Computers*, vol. 63, no. 12, pp. 3012–3025, 2014.
- [7] H. Ghiasi and M. Ghobaei-Arani, "Smart virtual machine placement using learning automata to reduce power consumption in cloud data centers," *Smart Computing Review*, vol. 5, no. 6, pp. 532–562, 2015.
- [8] Y. Xia, X. Li, and Z. Shan, "Parallelized fusion on m transportation data: a case study in CyberITS," *International Journal of Intelligent Systems*, vol. 28, no. 6, pp. 540–564, 2013.
- [9] Y. Xia, T. Zhang, and S. Wang, "A generic methodological framework for cyber-ITS: using cyber-infrastructure in ITS data analysis cases," *Fundamenta Informaticae*, vol. 133, no. 1, pp. 35–53, 2014.
- [10] K. T. Raghavendra, "Virtual CPU scheduling techniques for kernel based virtual machine (KVM)," in *Proc. of IEEE International Conference on Cloud Computing in Emerging Markets*, pp. 1–6, IEEE, Bangalore, India, October 2013.
- [11] Y. Xia, Y. Liu, Z. Ye, W. Wu, and M. Zhu, "Quadtree-based domain decomposition for parallel map-matching on GPS data," in *Proceedings of 15th IEEE Conference on Intelligent Transportation Systems Conference*, pp. 808–813, IEEE, Anchorage, AK, USA, September 2012.
- [12] Y. Mao, J. Oak, A. Pompili, D. Beer, T. Han, and P. Hu, "Draps: dynamic and resource-aware placement scheme for docker containers in a heterogeneous cluster," in *Proceedings of the 2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC)*, December 2017.
- [13] R. Zhang, A. Zhong, B. Dong, and F. Tian, "Container-VM-PM architecture: a novel architecture for docker container placement," in *Proceedings of the International Conference on Cloud Computing*, pp. 128–140, Springer, Saettle, WA, USA, June 2018.
- [14] Y. Fu, S. Zhang, J. Terrero, Y. Mao, G. Liu, S. Li et al., "Progress-based container scheduling for short-lived applications in a kubernetes cluster," in *Proceedings of the 2019 IEEE International Conference on Big Data (Big Data)*, pp. 278–287, IEEE, Los Angeles, CA, USA, December 2019.
- [15] M. Lee and Y. I. Eom, "Transaction-aware data cluster allocation scheme for qcow2-based virtual disks," in *Proceedings of the 2020 IEEE International Conference on Big Data and Smart Computing (BigComp)*, pp. 1–6, IEEE, Busan, Korea, February 2020.
- [16] S. F. Piraghaj, R. N. Calheiros, J. Chan, A. V. Dastjerdi, and R. Buyya, "Virtual machine customization and task mapping architecture for efficient allocation of cloud data center resources," *The Computer Journal*, vol. 59, no. 2, pp. 208–224, 2016.
- [17] Q. Liu, C. Weng, M. Li, and Y. Luo, "An in-VM measuring framework for increasing virtual machine security in clouds," *IEEE Security & Privacy*, vol. 8, no. 6, pp. 56–62, 2010.
- [18] L. Yu and C. Weng, M. Li and Y. Luo, "SNPdisk: An efficient para-virtualization snapshot mechanism for virtual disks in private clouds," *IEEE Network*, vol. 25, no. 4, pp. 20–26, 2011.
- [19] P. Graubner, M. Schmidt, and B. Freisleben, "Energy-efficient virtual machine consolidation," *IT Professional*, vol. 15, no. 2, pp. 28–34, 2013.
- [20] M. Du and F. Li, "ATOM: efficient tracking, monitoring, and orchestration of cloud resources," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 8, 2017.
- [21] L. Cui, Z. Hao, and X. Yun, "Piccolo: a fast and efficient rollback system for virtual machine clusters," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 8, 2017.
- [22] X. Bu, J. Rao, and C.-Z. Xu, "Coordinated self-configuration of virtual machines and appliances using a model-free learning approach," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 4, pp. 681–690, 2013.
- [23] J. Tang, P.T. Wee, Q. S. Q. Tony, and L. Ben, "system cost minimization in cloud RAN with limited fronthaul capacity," *IEEE Transactions on Wireless Communications*, vol. 16, no. 5, 2017.
- [24] W. Zhang, Y. Chen, X. Gao, Q. Z. ZhichaoMo, and Z. Lu, "Cluster-Aware virtual machine collaborative migration in

- media cloud,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 10, 2017.
- [25] C. Saravanakumar and C. Arun, “Efficient idle virtual machine management for heterogeneous cloud using common deployment model,” *KSI Transactions on Internet and Information Systems*, vol. 10, no. 4, pp. 1501–1518, 2016.
- [26] J. Vallone, R. Birke, and L. Y. Chen, “Making neighbors quiet: an approach to detect virtual resource contention,” *IEEE Transactions on Services Computing*, vol. 13, no. 5, pp. 843–856, 2020.
- [27] S. T. Teixeira, R. Calheiros, and D. Gomes, “Cloud reports: an extensible simulation tool for energy-aware cloud computing environments,” in *Cloud Computing. Computer Communications and Networks*, Z. Mahmood, Ed., Springer, New York, NY, USA, 2014.