International Journal of Electrical and Computer Engineering (IJECE)
Vol. 12, No. 5, October 2022, pp. 5630~5643
ISSN: 2088-8708, DOI: 10.11591/ijece.v12i5.pp5630-5643

A combined computing framework for load balancing in multi-tenant cloud eco-system

Amith Shekhar Chandrashekhar[1], Sharvani Gunakimath Suryakanth[2]
[1]Department of Information Science and Engineering, GM Institute of Technology, Davangere, India
[2]Department of Computer Science, RV College of Engineering, Bengaluru, India

Article Info

Article history:

Received Jul 23, 2021
Revised May 27, 2022
Accepted Jun 25, 2022

Keywords:

Cloud computing
Load balancing
Multi-tenancy
Task scheduling
Virtualization

ABSTRACT

Since the world is getting digitalized, cloud computing has become a core part of it. Massive data on a daily basis is processed, stored, and transferred over the internet. Cloud computing has become quite popular because of its superlative quality and enhanced capability to improvise data management, offering better computing resources and data to its user bases (UBs). However, there are many issues in the existing cloud traffic management approaches and how to manage data during service execution. The study introduces two distinct research models: data center virtualization framework under multi-tenant cloud-ecosystem (DCVF-MT) and collaborative workflow of multi-tenant load balancing (CW-MTLB) with analytical research modeling. The sequence of execution flow considers a set of algorithms for both models that address the core problem of load balancing and resource allocation in the cloud computing (CC) ecosystem. The research outcome illustrates that DCVF-MT, outperforms the one-to-one approach by approximately 24.778% performance improvement in traffic scheduling. It also yields a 40.33% performance improvement in managing cloudlet handling time. Moreover, it attains an overall 8.5133% performance improvement in resource cost optimization, which is significant to ensure the adaptability of the frameworks into futuristic cloud applications where adequate virtualization and resource mapping will be required.

Corresponding Author:

Amith Shekhar Chandrashekhar
Department of Information Science and Engineering, GM Institute of Technology
Davangere, Karnataka, India
Email: amithsc@gmit.ac.in

1. INTRODUCTION

The cloud computing (CC) paradigm has been evolving for the last two decades to improve data and information systems' quality and service modeling. The evolution witnessed the transition from desktop computing technologies to client-server architecture where virtualization plays a vital role in resource provisioning and sharing. With the invention of cloud computing models, the researchers explored the limited features associated with the web-enabled computing models in practice due to their limited capability to handling a large corpus of data streams with the uncertainty involved [1], [2]. The researchers also realized that CC technologies could be a good substitution for traditional web-based computing systems. The CC technologies offer better solution space to deal with a large corpus of data processing and offer a basis to provide a different level of agreements upon software frameworks as a service, virtualized platform-as-a-service, and Infrastructural resources as a service [3]–[6]. The computing models deliver various resources over the Internet with ease of processing and task execution through these different service layers. The service delivery models are also associated with various profit-making policies which leverage the economic balance [7]–[9]. However, as the streaming services in cloud ubiquitous computing demand faster execution of workflow along with adequate resource provisioning requirements, here it indicates that the user requests should be processed and handled at the server side with timely execution and uniform distribution of tasks considering resource sharing under multi-tenancy. It is also needed to be maintained to provide a better quality of user experience at the user base (UBs) [10]. However, the existing load balancing techniques fail to handle the uncertain load of tasks at the virtual server-side and cannot cost-effectively manage resource sharing and distribution.

The study addresses this problem and introduces two different computing frameworks to deal with cloud resource management. This work jointly integrates two frameworks: data center virtualization framework under multi-tenant cloud ecosystem (DCVF-MT) and collaborative workflow of multi-tenant load balancing (CW-MTLB), to perform effective load balancing in virtualized multi-tenant cloud eco-system. Both the systems are analytically modeled and implemented considering numerical computing environments where algorithms are presented to deal with the resource mapping for task execution. The outcome of the DCVF-MT shows that the hybrid approach of resource allocation and virtual machine (VM) placement with link formation attains 24.778% better performance than the one-to-one isolated approach. On the other hand, another system, CW-MTLB, performs better with 40.33% in cloudlet handling and processing time and performs 8.5133% improvement in resource cost optimization. The entire manuscript is organized as section 2 discusses the proposed procedure correspond to the research, followed by the research method in section 3. Further section 4 discusses the extensive result and discussion for both frameworks. Finally, section 5 concludes the overall research contributory aspects.

In the context of cloud computing, different facets must be handled when it comes to load balancing. It comprises diverse problems such as data center virtualization, link scheduling, task allocations, prioritization, and resource sharing. In recent years, several research works have been published regarding the exploration of load balancing problems and their countermeasure. In this direction, a survey study [11] highlighted significant terminologies and explored load balance and traffic management issues in cloud computing. It also shows serious concern about link density and the complex design of the existing solution for normalizing traffic congestion. Another recent study provided an interesting analytical review work regarding the load balancing issue. A comprehensive study is carried out concerning different static, dynamic, and nature-inspired techniques to solve problems related to data center response latency and overall performance [12]. However, the conventional static methods are no longer efficient due to non-uniform load distribution on nodes and delay due to context switching [13]. Static approaches are also unsuitable for dynamic changes to the load during runtime and have low fault tolerance factors [14]. Researchers have focused on dynamic methods to address such issues due to their adaptive nature, flexibility, suitability to the dynamic environment, and it does not require any previous information [15]. A priority-based dynamic method for load balancing is given by [16], where a customized throttled algorithm is designed to allocate incoming tasks to run high priority tasks instead of low priority tasks. It also enables uniform distribution workload among several virtual machines. However, this approach has a high response time for low-priority tasks. Another work based on the dynamic approach is presented by [17] to enable even distribution of workload by keeping tables of virtual machines with header available and busy. The outcome shows slightly less response time, but it needs optimization in its design and implementation process to enhance the overall performance. Another recent work [18] introduced a resource-oriented load-balanced min-min scheme to diminish makespan and maintain an effective balance among the workload on virtual machines.

Similarly, a resource-efficient is suggested in [19] to design an enhanced version of the conventional min-min scheme based on considering execution and completion time in a matrix. However, this work does not consider the updated workload of the virtual machine during task allocation. The adoption of metaheuristics techniques has also received wide attention towards mathematical modeling to perform better efficient load balancing in the complex and dynamic environment [20]. Hashem *et al.* [21] used a nature-inspired honey bee algorithm to perform load balancing by considering updated load count in the task allocation. However, this approach lacks the issue of overloading the virtual machine. Recent work in a similar direction by [22] presented heterogeneous initialized workload balancing for task scheduling.

On the other hand, the authors use a genetic algorithm with heterogeneous initialized to obtain optimal fitness value to minimize the load deviation. A method based on workload processing and connected data for task scheduling in a dynamic computing environment were suggested by [23] based on MapReduce computing technique. The work carried out by authors in [24] presented a resource-efficient job allocation task for a multi-tenant cloud system using query classification and worker sorting. In a similar direction, recent work [25] suggested a model based on traffic and link density analysis to evaluate multi-tenant load balancers for cost-effective resource distribution in cloud computing. In another study [26], the load balancing factors for virtual machine allocation as NP-harp problems are addressed by adopting ant colony optimization. The adoption of OpenFlow and distributed virtual switch techniques are considered in the work of [27] to address issues of load balancing, flexibility, and scalability in the multi-tenant-based virtual datacenter. Similar work can also be seen in the study of [28], where a concept of a distributed virtual switch is considered to address issues associated with virtual server placement mode (VSP) and the link establishment phase. This work has provided an effective workload balancing in a multi-tenant-cloud system. A load balancing scheme for the dynamic and shared system is given by [29]. In the study, the authors have used max-max, min-max, and folded crossed cube network approach to reduce load imbalance factors and improve overall performance concerning average resource utilization, run time complexity, and makespan.

It is essential to elaborate on the significance of the virtualization for the effective synchronization of the client traffic resource requirements and design a load balancing process accordingly in the multi-tenancy environment so that the models archive various performance metrics, including the cost of the infrastructure, resource availability, and flexibilities. Despite having potential features, the study realizes that the traditional load balancing mechanism lacks effectiveness. Their practicability with varied traffic conditions cannot be easily extended to the cloud zonal data centers (DC→Zi). It is also observed that- significantly less research scope is inclined towards handling the link density problem at data centers, which often leads to traffic congestion problems and link failure problems. The prime reason behind the congestion of data traffic at the direct current (DC) components is that link density is comparatively higher at DC than at the enterprise network. It makes the operational situation in the cloud environment in the form of a worst-case scenario, and traditional load balancing methods to be applied suffer significant consequences with the variable traffic load across the cloud computing network. Thereby to handle the dynamic traffic at the server-side under the cloud multi-tenancy ecosystem, an adequate resource provisioning approach is highly envisioned for proper task allocation without compromising the quality of user experience. However, designing such systems analytically with variable conditions of traffic is a highly challenging research task.

2. PROPOSED PROCEDURE

In this research study, a combined framework of the distributed virtual switch-based load balancing algorithm is designed using a virtual link establishment (VLE) algorithm where two phases like: i) VSP mode and ii) VLE are designed to balance the optimal traffic, bandwidth demand, and the link count. The framework also integrates another mechanism of CW-MTLB which also follows analytical modeling of host-cloud multi-tenancy for task allocation in a distributed environment. The system here aims to introduce two different traffic scheduling approaches: i) an open form of collaboration and ii) a close form of collaboration, respectively. Figure 1 shows the proposed combined architecture to ensure adequate resource provisioning with better task allocation in the cloud computing ecosystem.

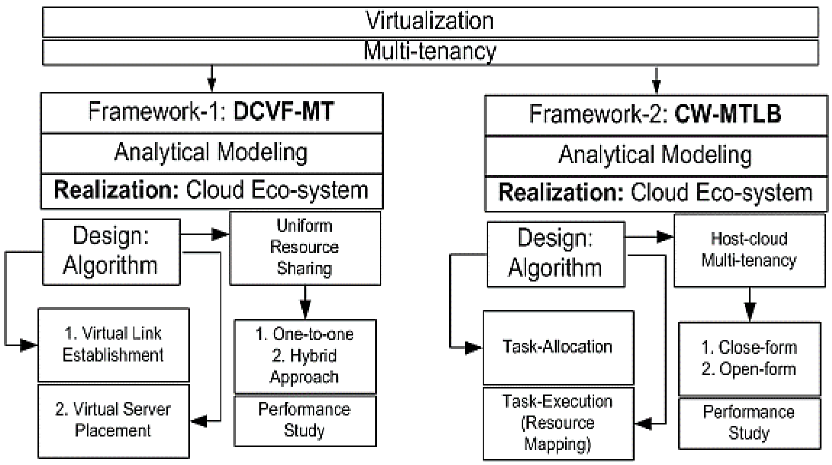

Figure 1. Proposed combined architecture

Finally, the study outcome exhibits the performance justification for both CW-MTLB and DCVF-MT. Extensive numerical analysis has been performed for both the framework DCVF-MT and CW-MTLB, where it also performs inferencing of the data points based on the outcome and reasoning to justify the effectiveness of both the systems.

The prime contribution of the study shows that in DCVF-MT, the inter and intra mode of traffic metrics are supported dynamically in virtual private cloud (VPC). Here, the components corresponding to

VPC enable efficient interaction between the virtual server and destination servers under variable traffic load in a virtual networking scenario that flexibly handles the network traffic. The LB component installed in DCVF-MT also enables better traffic management under predictable conditions. The performance assessment scenario also shows that in both intra-edge and inter-edge traffic conditions for bandwidth demand, DCVF-MT manages computing resources effectively without affecting the communication scenario. The analysis also shows that the system flexibly handles the traffic when traffic load increases, and for simulation time, it allocates resources for VPC to another edge-switch when required, which is quite a novel aspect in this is proposed system and assist in a well-balanced flow of traffic and resource allocation. DCVF-MT is also capable of formulating the maximum possible links during resource provisioning.

The proposed solution corresponds to the design procedure of CW-MTLB shows that it performs logical operations with simplified workflow modeling in virtualized cloud eco-system modeling to deal with uncertain and complex data traffic. The system evaluated for three different volumes of cloudlet in the context of the incoming flow of traffic and the performance outcome of the formulated approach of load balancing under host-tenant multi-tenancy assessed for the metrics cloudlet handling time and unused resources. However, the study considers four other baseline approaches as related studies and evaluates their models under the same operating conditions to see the behavior of traffic management instances. The study's outcome shows that incorporation of open form and close form of collaboration under multi-tenancy and virtualization brings a novel aspect in handling the traffic and resource mapping in cloud eco-system. It also contradicts the theories in the related studies to some extent. The experimental results show that CW-MTLB attains better execution management and resource utilization, which ensures better resource cost optimization in the cloud. The better execution along with resource cost management capability makes CW-MTLB flexible and better than the related approaches.

This part of the work jointly integrates two models, namely i) CW-MTLB and ii) DCVF-MT, into a combined framework that is numerically modeled considering the baseline theory of virtualization and multi-tenancy. Both the frameworks are analytically modeled where both the frameworks DCVF-MT and CW-MTLB have been realized, considering the representation modeling of cloud eco-system to emulate the real-time scenario with a computing framework. The proposed design procedure for DCVF-MT shows that it performs both uniform resource sharing and virtual link establishment for different job requests. Its virtual server placement assignment makes the system more robust and operational under multi-tenancy, which makes it capable of handling simultaneous processes for execution. The resource mapping and resource sharing capability aim to provide better job requests execution under a one-to-one and hybrid approach.

On the other hand, the study also aims to model another framework CW-MTLB in such a way that the proposed solution under the same framework provides better task allocation and execution with a different approach of resource sharing capability under virtualization and multi-tenancy. The host-cloud multi-tenancy here in this approach provides i) close form and ii) an open form of collaboration among the computing resources to leverage the performance of the cloud traffic management. Both the proposed analytical designs are further formulated with extensive discussions in the consecutive section of the study.

3. METHOD

3.1. Research method of data-center virtualization framework

The methodology adopted for "modeling of load balancing and multi-tenancy-oriented data-center virtualization framework" is analytical modeling, where typically, the solution of data-center virtualization is classified as either static approach or the dynamic approach. Similarly, the operating environment of DCVF-MT also constructs by employing two different approaches, where assigned network functions and computing resources become permanent with the initial modeling, whereas, in the second approach, DCVF-MT provides another mode of operation to manage the reconfigurable resources in the cloud eco-system. The core functional modules of the algorithm involve two distinct modes of networking operation as: i) in the first mode, it assigns adequate virtual server localization in VSP mode and ii) DCVF-MT also enables virtual link establishment and efficient resource mapping mechanism to leverage the cloud service modeling significantly.

The analytical design of the DC-VF-MT framework incorporates a core component of distributed virtual switch-based load balancing schema which enables well-managed link establishment with the virtual servers. The prime underlying motive of this conceptualized design is to schedule and process simultaneous job requests $J_{req}(i) \rightarrow \forall i \in N$. The job requests originated from the UBs can be mathematically represented with the (1).

$$J_{req}(i): J_{req}(i) \rightarrow \forall i \in N \quad (1)$$

The functional entities of load balancing basically incorporate a switch-based strategy by means of two work-flow execution modeling. Here DCVF-MT manages datacenters with multi-tenancy-oriented virtualization and profiling. The prime agenda is to manage the incoming uncertain job requests from the UBs.

Initially, the DCVF-MT analytical model considers the number of servers (nS) to be deployed within particular geographical regions of the cloud ecosystem and termed zones (Zi)$ns \rightarrow nS \cup Zi$. Here nS and Zi are two different sets from the mathematical viewpoint. The system also considers another set of DCi located within a particular zone as discussed above.

The prime underlying motive of DCVF-MT is to handle the uncertain flow of incoming/outgoing requests from cloud tenants by balancing the inter-traffic flow movements. The system of DCVF-MT is also capable of handling specific events that correspond to the starting or ending points of inter-traffic flows. The modeling of the system framework considers two different modes of traffic metrics, where the first one is i) one-to-one approach and ii) hybrid approaches. The prime motive of two different approaches is to perform load balancing of UB-generated requests by creating several VM instances in virtual servers to optimize the bandwidth requirement for Dsize and optimizes the traffic performance in both the context of Inter-edge and Intra-edge modes.

The system formulation of DCVF-MT performs virtual like establishment in such a way so that it achieves optimal bandwidth performs with variable bandwidth demands (βd) from UBs. The execution workflow of DCVF-MT incorporates a distributed virtual switch-based load balancing algorithm using a VLE algorithm with two phases. It includes i) virtual server (Vservers) placement and ii) VLE is designed to balance the optimal traffic, bandwidth demand, and the link count. The prime reason behind VLE is to facilitate the load balancer component to significantly assign a coming user request to the particular V-server with a proper resource mapping mechanism.

The key design features associated with DCVF-MT is: i) the system aims to offer better resource utilization, i.e., network resources with optimized infrastructure investment cost; ii) the DCVF-MT load balancing approach is flexible enough to deal with the variable load of traffic, which also involves time series uncertainty. It can efficiently allocate V-server to the users according to their requirements and does not pose overhead to the overall networking system; and iii) the system is designed in such a way where it also improves the network functions associated with the traffic-locality factors (i.e., Inter-edge (¨inter) traffic and Intra-edge (¨intra) traffic) thus, virtual resources can be utilized more efficiently.

The methodology adopted to design and conceptualize the DCVF-MT is a flow of mathematical modeling schema which aims the load balancing to take place under virtualized multi-tenancy-oriented data-center framework". Figure 1 represents the DVCF-MT schema with a block-based design where different components are involved in making system functions with different roles. The module user-base (UBs) generates requests to the cloud where the initial stage of processing of requests (U_{req}) is streamlined and routed through core switches $Ci \rightarrow Cs$, here Cs refer to the cloud size and Ci referred as cloud core switches. The core switches are integrated with the edge switches Ei. Here Ei functions depending upon the arrival of the U_{req}. The bandwidth demand (βd) is also evaluated for respective UBs → Ui.

3.1.1. DCVF-MT: virtual server (VS) placement

In this stage, the framework assigns the quantity of virtual-server modules (Vservers) according to the client's solicitation/requirement. In this virtual server localization stage process, the framework also keeps track of all traffic flow in the different levels of traffic zones. It also means that the framework also tracks the directives of expanding the degree of traffic zone all through the interaction. That is to say, the server modules in the virtual private cloud are arranged in the topological request, which is also assigned in closer proximity with one another, to keep massive scope network traffic on progressive low-level switches. Two virtual server modules (Vservers) can be an example connected with the single edge switch. In this case, the request flows per demand stream from one node considering switching power associated with that edge-switch component. Then again, assume a set of Vservers is associated with instantly recognizable switches. In that case, the DCVF-MT explores a similar set of requests streaming between the Vserver, and it needs to utilize extra network organization functions between the 2-layers and external switching power from the center switch. Thus, it is necessary to distribute the Vserver similarly to resource provisioning; thus, some accessible edge switches are involved for the assignment and service delivery of the Vserver.

3.1.2. DCVF-MT: establishment of virtual link (VL)

DVCF-MT follows the VL establishment process after finalizing the phase mentioned above of VS placement. It performs computation and establishment of network lines as edges between the servers. Here the DCVF-MT scheduling process mechanizes a structured implementation of selecting appropriate core switches (Ci). It helps in establishing the links between two distinct layers, such as core and aggregation layers. Here this establishment stretch follows the route for ¨inter traffic flow setup and scheduling. A similar process is observed in the previous phase of computation, where VServer placement is taken care of. The DCVF-MT also adopts a dynamic approach for scheduling traffic. In the proposed approach, practically no physical connection is being established between the hardware switches, which are not allocated to the virtual private cloud. As per the requested services, the DCVF-MT scheduling process finds the appropriate route for link establishment and performs connections among the allocated switches. The step-by-step execution of the algorithm is modeled as follows and the designed numerical algorithm work flow of DCVF-MT follows execution of the above analytical models which are illustrated is as:

Algorithm for proposed DCVF-MT scheduling

Input: [U_{req}, C_i, E_i, $V_{servers}$]
Output: [Δ_{inter}, Δ_{intra}, S_{time}, L_{count}]
Start
1. Perorm U_{req} operation
2. Init. U_{name}, U_{zone}, R_{time}, and D_{size}
3. Enter user information record
4. Add number of U_{req}
5. Compute U_{req} per zone
6. Update by adding traffic
7. Compute→bar:freq(Ureq) and plot → traffic/Z_i
8. Perform VSP Mode
   Localize cloud $\sum\left(\frac{A}{2}, dx.C_s - \frac{C_s}{2}\right)$
9. Initialize C_i, E_i,
10. Compute localization of C_i and E_i
11. Add $V_{servers}$
12. Compute time series localization as
   $X_t \rightarrow \sum Sl_{xi} + S_x * sSize - \frac{sSize}{2}$
   $Y_t \rightarrow \sum Sl_{yi} + S_y * sSize$
13. Visualize the server
14. Establishement→VL: Edges
15. Perform Load blancing algorithm
    a. One to one
    b. Hybrid
16. Assing the value of bandwidth
17. Repeat the step (8→13)
18. Compute traffic per zone, link count
19. Anlyze: traffic proportition V_s simulation time.

End

The design and development of the DCVF-MT framework follows analytical modeling and computing. In the initial stages of computation, the process generates user requests (U_{req}) which are being queued in the form of array structure and the sequence of work-flow send each U_{req} to the cloud as U_{req}→ Cloud ecosystem. The first step of DCVF-MT algorithm subjected to initialization of set of parameters which are associated with the number of users generated U_{req} which comprises of master user information such as U_{name}, U_{zone}, R_{time}, and D_{size}. This all information is sent to the cloud eco-system by individual user taking the support of DCVF-MT computing system. The DCVF-MT computing system here checks for individual request correspond to each Z_i and perform a comparison with other zone and update the database with added traffic. The system further performs localization of the coordinates correspond to cloud eco-system and with a deployment phase. During the deployment phase of cloud eco-system, the system also performs deployments of coordinates correspond to a single core switch (C_i) along with the deployment of four edge switches (E_i). The primary responsibility of the core switches here is to balance the inter-traffic flow of request and execution considering virtual server placement. However, on the other hand edge switches perform effective route establishment through link formation which interconnects VSs with the network in such a way where it becomes capable of handling intra-traffic flow also. It can also handle the traffic between zones (Z_i) as well. Along with the cumulative placement and traffic management through C_i and E_i, the system also evaluates the number of Vserver and add those required servers in the cloud as per the request originated from the UBs. The efficiency with respect to cost of operation and functional evaluation for both the approaches also computed with respect to a variable parameter namely Bandwidth demand which ranges between 1-6 units. In both the process the system is mechanized in such a way where it looks for the computation of different matrices such as the traffic per zone, traffic locality and link count. Finally, an updating process take place where it updates the traffic locality as per the request and user demand for network resources and then the process moves to each and other zone by updating the VL link formation process. Another approach of CW-MTLB is also analytically designed and modelled in such a way where it also aims to provide adequate resource provisioning in the multi-tenant cloud environment. The extensive discussion of another approach is illustrated as.

3.2. Research method: CW-MTLB

The mathematical model for the complex system is designed considering the cloudlet of different type of the traffic classifier synchronized with the load balancer and the cloud resources. The framework name for efficient cloud load balancing and traffic management is referred as CW-MTLB which follows the principle of round-robin scheduling. Finally, after developing the mathematical model the simulation results are benchmarked with the existing state of the art work for the load balancing.

It is not easier to represent a complex real-time cloud system with the aid of mathematical modeling and designs. However, the study conceptualizes a workflow model of multi-tenancy, considering a cooperative load balancing approach to provide adequate resource sharing in virtualized cloud computing environments. The CW-MTLB is proposed based on the philosophy of ubiquitous computing and dynamic application resource scheduling for integrated applications. The design plan also aims to deal with distributed integrated applications that require seamless connectivity in the context of continuous task processing and traffic load balance with resource sharing in the isolation fashion of cloud subscribers (CS)/tenants.

It further designs and represents the overall system model with the mathematical notion from an analytical analogy. The multi-tenant cloud computing system in this formulated approach also deals with a variable traffic load of executable tasks considering shared infrastructures, applications, and databases among a set of tenants. The services are streamlined in an isolated fashion of resource sharing and task computations. It also enables the LB to allocate the incoming frequent tasks/job requests from tenants with the mapping to the adequate and appropriate computing resources (CR) for processing and execution. Here arrived job requests that correspond to the variable traffic factor are also considered in the form of cloudlet requests. It is classified as a set of three categories as $C=\{T_1, T_2, T_3\}$.

In this study, the cloudlets as a task of requests get generated by multiple tenants looking for different types of cloud services. Also, during the system modeling and design, it is assumed that in the context of downside multi-tenancy modeling, the cloud subscribers or tenants may not have the provisions to customize the mode of cloud services to accomplish their specific needs. There are three different types of cloudlet requests that are considered during design and modeling, such as T_1 (small volume cloudlets), T_2 (medium volume cloudlets), and T_3 (large volume cloudlets). The analytical system design and development of CW-MTLB, initially the group of cloudlets (C) is initialized in different tasks. A collaborative set of resources is considered a model parameter referred to as an isolated cloud cluster (i-Cc) for CW-MTLB. The Cc here is also used as a group collaborative infrastructure of computing resources where the load-balancer (L.B.) is aimed to be executed. Here the system also realizes the Cloud of cloud as a multi-cloud environment modeling for load balancing. Analytically i-Cc can also be represented as a collection of collaborative computing elements i-Cc→CE_i.

Here i-Cc→CE_i hosts their respective (N_{vm}) virtual machine instances to manage the computing resources to serve the tenant which has requested cloud services with better storage capacity (S). The cumulative number of collaborative computing elements at resource side can also be mathematically expressed:

$$i - Cc(total) = \sum_{i=1}^{n} CE_i \tag{2}$$

Here n represents the number of indexes to identify individual C.E. at the resource side of the cloud model. Here the server strength is directly proportional to the number of VM it can host through individual C.E. depending upon the resource requests and traffic demand. The i-Cc also supports different forms of C.E. as clusters with constraint resources which are optimally used depending upon the requirements and incoming variable load of traffic on a sharing and scalable basis. The L.B. component plays a crucial role here in managing the computing and resource scheduling towards task execution. In this context of the research, the number of individual C.E. (n) is assumed to be ($n \leq 4$), where the upper bound (U.B.) of computing clusters are fixed to ($max \rightarrow 4$) and yet flexible with resource constraints. If n=4, then the i-Cc also can be represented as a set of cluster elements as i-Cc=$\{CE_1, CE_2, CE_3, CE_4\}$ where each $CE_i \in$ i-Cc maintains and hosts its strength of resources in the instances of VM. The different cloud clusters of resources with operational constraints are segmented into four major types according to their capability vector of computing and processing abilities. The analytical modeling here considers CE_1 as computing resources with lower computation and processing capabilities, whereas CE_2 refers to computing resource clusters with medium capacity.

On the other hand, CE_3 refers to high-performance resource computing with higher capability, and CE_4 refers to very high cluster computing capacity. Thus, the analytical model formulation considers cloud clusters of different abilities to authenticate and validate the model for both scalable hybrid cloud usage and on-premise portable cloud systems, including the traditional working model of private and public cloud infrastructures on sharing basis. CW-MTLB here considers the architecture model in Figure 1 of framework-2 as a base for load balancing with on-premise multi-tenancy-based cloud computing modeling.

It clearly shows how the workflow modeling is proposed with a block-based representation where the first block represents the cloud tenant layer. Here tenants request a specific service to be executed under the isolation fashion of multi-tenancy support by the CW-MTLB approach. The tenant layer generates different types of cloudlet requests (C) categorized into three categories, as already discussed above. In addition, the CW-MTLB incorporates a programmed computing unit of cloudlet classifier which assists in organizing and classifying the incoming tasks of C={T_1, T_2, T_3} according to their associated traffic load and variable strength, which the work-flow manager and scheduler further manage. CW-MTLB also considers another computing reference model of the L.B. algorithm, which helps manage the incoming traffic of requests and allocates the computing resources in the back-end efficiently. The cluster of resources and computing elements corresponds to the infrastructure-as-a-Service (IaaS) layer, which helps in virtualization and manages incoming cloudlets in an isolated fashion. Also, it involves a resource-sharing concept to generate scalable performance of task execution for cloudlet types T1 to T3. The computing resources i-Cc are managed by a layer of virtualization considering four different isolated resource clusters share their computing resources on a requirement basis. In order to validate the performance of the computing for different traffic management approaches, it also studies the behavior of the models by considering their outcome obtained for different simulation parameters such as i) cloudlet handling time (msec), ii) unused resources, iii) unused memory, and iv) resource cost which assists in formulating the performance metric and provides a better insight into the experimental outcome trend and possibilities and limitations. The system also aims to handle and exploit the adaptive resource allocation with optimized performance of computing the dynamic heterogeneous cloudlets in the multi-tenant CCE. It also targets economical and well-balanced resource sharing and utilization to maximize uninterrupted services to the on-premise tenants with flexibility and maximum availability with load balancing despite variable traffic conditions.

3.2.1. Load assessment for different traffic-management policies as a baseline

The traffic management modeling of the proposed system evaluates the performance of CW-MTLB and further checks and validates the behavioral aspects associated with the other four baseline approaches. All the baseline approaches promote different strategic policies of task-scheduling and load balancing under the identical on-premise multi-tenant cloud environment adopted for CW-MTLB. The added traffic management protocols along with P1 also can be represented mathematically with a set representation of (Pt)={P1, P2, P3, P4, P5}. Here for the traffic scheduling objectives where: P1=proposed multi-tenant load balance (CW-MTLB), P2=round robin at server side (RRS), P3=round-robin at host side (RRH), P4=hybrid of RRS and RRH, namely Hy-RRSH-H with higher capacity of servers and P5=hybrid of RRS and RRH, namely Hy-RRSH-L with a lower capacity of servers as compared to the Hy-RRSH-H. The analytical formulation of the CW-MTLB considers Ci where i→1,2,3, which considers the traffic-load type on the prioritization basis, and further for each kind of Ci, the system finds four types of job requests Jr $\in$ {Jr_1, Jr_2, Jr_3, Jr_4}. The consideration of job requests within an on-premise cloud environment for each type of Ci on a prioritization basis can mathematically be expressed as (3).

$$JTotal_m \leftarrow \sum_{i=1}^{m} Jr(i)\ here\ 1 \leq m \leq 4 \quad (3)$$

As show in (3) is generalized four different traffic loads from cloudlet job requests that arise from the cloud tenant layer. Here the upper bound of Jr is considered a maximum 4 for the numerical modeling and optimization of resource modeling. The system modeling of CW-MTLB also employs a dispatcher unit combined with a cloudlet generator unit which helps in pipelining the incoming variable job requests within a particular time slot (Ts). The CW-MTLB system formulation also considers VM correctly mapped with CE. VM are considered to be of different computing capabilities and processing elements such as memory, and computing power. The computational capacity (CC) corresponds to each VM is depicted with a weight factor of resource mapping and execution of tasks. The CW-MTLB also computes the capability vector (C_v) for each host-machines from nHm→CE_i and calculates the outcome in the variable U_{Ri}, indicating the resource status. This process can be computed considering the (4) and already shown in the algorithm of CW-MTLB.

$$U_{Ri} \leftarrow \sum_{i=1}^{nHm} Cv(i) \quad (4)$$

The algorithm also computes the resource metric with $Log_{10}(U_{Ri})$ and resource cost with the (5).

$$R_{cost} = \sum C_{matrix}: M_N \quad (5)$$

The cost matrix C_{matrix} computed concerning the resource metric as memory, M_N={M_1, M_2, M_3, M_4, M_5}. The step-by-step algorithm overview of the analytical algorithm of CW-MTLB is formulated as.

Algorithm for multi-tenant cloud computing in CW-MTLB

```
In: Cj ∈ { T1, T2,T3}1≤j≤3
Out: VMi, (Task) Ta | Ti allocation
Initialize: Cj, Pt, (Jr)m, VMn
Process:start
Check type of :Cj
        Compute (3): Cj ∈ {T1, T2, T3}1≤j≤3
Update :(Jr)m
        (Jr)_m = w × Σ_{i=1}^{m} J_i || J_m where w = {1,2,3}, 1 ≤ m ≤ 4
Compute (VMTotal)
        VM_Total = Σ X_{i,j} ∈ (Jr)_n where i = 1, j&n = 1 → m
Initialize Htot
Create VMi: Htot
        for j←1: 3
                for t←1:5
                        [VMn]←VMi(Pt , Cj)
                end
        end
Perform: Task Allocation
    1.  Load Balancing: CFC: HM(availability_check)
    2.  Load Balancing: OFC: HM(availability_check)
        URi←Σ_{i=1}^{nHm} Cv(i)
        Compute: Log10(URi).
        Compute: Rcost= Σ Cmatrix:MN
End
```

The virtual machines are dynamically created based on C_j and load balancer or scheduler traffic management protocol based on the capacity vector of multi-dimension. The analytical algorithm forms an execution flow concerning different types of cloudlets. The job processing and execution occur by mapping VM instances with specific server racks managed by the load-balancer. The execution flow of the above algorithm shows how the VM instances are finally created considering the type of vectors corresponding to P_t, C_j using the respective host machines HM. The above algorithm CW-MTLB decomposed into a 2-layer workflow, which deals with H_M-assisted efficient traffic management and load balancing in a multi-tenant cloud ecosystem considering the CE_i∈ i-Cc.

4. RESULTS AND DISCUSSION

This section illustrates the outcome obtained for both the combined frameworks and exhibits in this presented section. The analysis was carried out for both DCVF-MT and CW-MTLB. The analysis in the initial phase shows how DCVF-MT performs better resource allocation and task processing concerning adequate VLE and VM placement.

4.1. DCVF-MT experimental results

In virtual server localization in the VSP mode of execution, a virtual private cloud eco-system is constructed systematically. It aims to exploits the resources that correspond to successive server slices. However, DCVF-MT aims to accomplish better traffic locality with adequate resource provisioning, indicating that it must minimize the inter-traffic load. Here inter-traffic load means optimizing the work of the VL establishment process with minimum infrastructure and cost. The experimental outcome here shows that variation exists in the outcome trend related to both inter-edge and intra edge traffic flow concerning the increasing simulation time. Here, the scenario is executed considering a set of virtual servers with a minimum of 10 with bandwidth demand =6. The system also assesses the traffic distribution metric, which indicates a uniform distribution of computing resources with balanced traffic flow and shows the adequate load placement distribution among the virtual servers.

The Figure 2 also exhibits the trend of the outcome correspond to the network traffic flow in the network environment. To evaluate the traffic locality in terms of both Δ_{inter} and Δ_{intra} traffic, the observation with respect to increasing simulation time shows a projection of the traffic locality where it is seen that in the massive network environment, if the proportion of the Δ_{inter} traffic load increases then it negatively influences the proportion of proportion of Δ_{intra} traffic load, that means it decreases to a greater extent. Another observation from the experimental simulation shows that when the size of the virtual cloud become large then the system flexibility shows that one server component which belongs to VPC can be localized to another edge-switch.

The DCVF-MT system performance also evaluated with respect to variable network traffic conditions. Here it is observed that how much percentage of network traffic is well-managed. In this regard the system evaluates the metric correspond to link-count (Lcount) for variable payload of distinct bandwidth requirement (βd). The analysis from the Figure 3 shows the performance metric of Lcount vs. βd. Here it can be noted that the trend of link count increases with respect to higher values of corresponding βd. The closer interpretation also shows that in both the load balancing cases link count increases with respect to bandwidth requirement. However, in the case of hybrid approach the number of link count is more as it serves maximum possible resources in the same bandwidth and also explicitly it can be said that higher range of links involve higher bandwidth dependencies.

This extensive analysis has been carried out in different network traffic scenarios and simultaneously the load balancing algorithms are executed for traffic management from the VLE and placement process. The scheduling in the context of DCVF-MT also shows that the system here attains better outcome and comply with the bandwidth demand and process the tasks effectively with timely execution. Therefore, it can be said that the traffic flow is well balanced and maintained uniformly during the resource allocation in DCVF-MT framework.

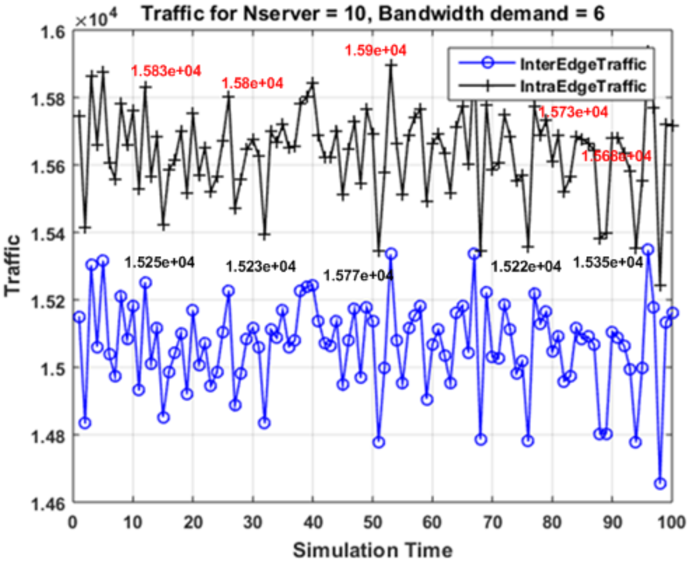

Figure 2. Outcome of the inter-edge/intra-edge traffic with respect to simulation time

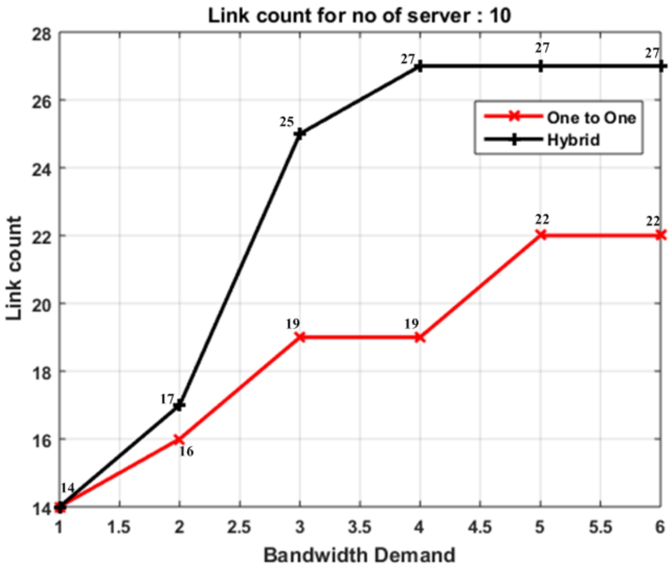

Figure 3. Graphical representation of the outcome of the link-count with respect to simulation time

4.2. CW-MTLB experimental results

The research study in this section presents a behavioral performance analysis of 4 different traffic management protocols and the formulated approach of CW-MTLB on a multi-tenancy cloud environment. The framework is designed in a way where protocols like RRS, RRH, HyRRSH-H, and HyRRSH-L are numerically modeled to be integrated with the proposed multi-tenant cloud environment. The research study assesses the performance of the traffic management protocols towards dynamic resource provisioning with their approach of underlying load balancing modeling. The numerical modeling is structurally designed for a framework where four above-highlighted procedures such as RRS, RRH, HyRRSH-H, HyRRSH-L are executed for performance validation purposes. The outcome obtained for different instances of analysis is shown in Figure 4.

The analysis in shows that for three different volumes of cloudlets such as i) Large volume, ii) medium volume, and iii) small volume, out of all the approaches, CW-MTLB obtains better and superior outcomes towards handling the cloudlet processing and execution of task allocation with better resource provisioning. On the other hand, when the other four approaches are concerned, Hy-RRSH-L and Hy-RRSH-H outperform RRH and RRS in cloudlet handing management and task execution scenarios. The visualization of the numerical outcome is further illustrated with the following Figure 4, which exhibits that the proposed approach CW-MTLB attains better performance when time complexity of cloudlet request handling for better task management is concerned. The cloudlet handling time from complexity view-point measured with respect to the cumulative time to execute the tasks and its associative resource allocation as cloudlet: $O(T^n)$, it indicates that the model behavior to three different types of cloudlets is found providing consistent performance even if the incoming data of different volume is of n dimension.

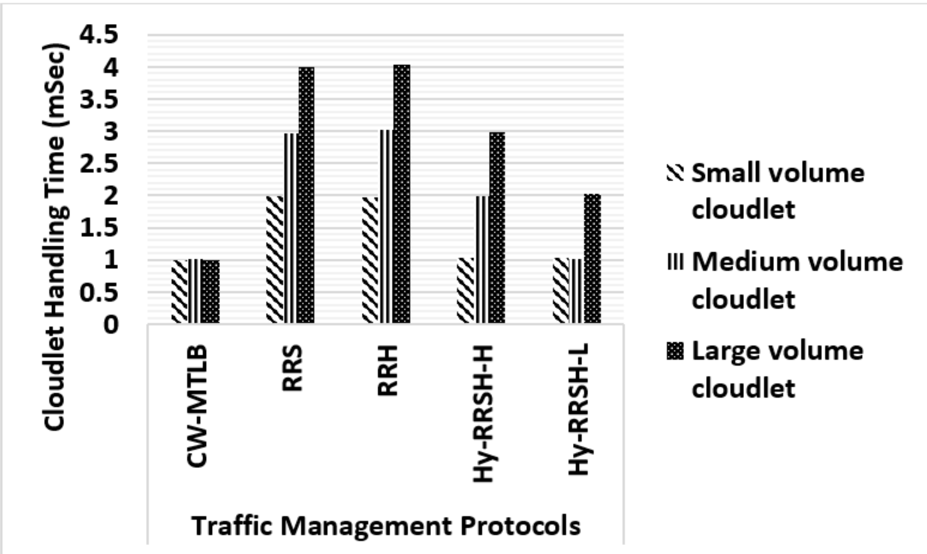

Figure 4. Behavioral analysis of cloudlet handling time (msec)

The framework simulates all the load balancing approaches in the presence of three different traffic modeling, and the outcome corresponding to their behavioral analysis shows that CW-MTLB effectively utilized its resources by virtualization and mapping. A closer interpretation of the quantified outcome corresponding to cloudlet handling time (mSec) shows that as compared to CW-MTLB, the nearest best performing protocol is Hy-RRSH-L which exhibits an outcome of 1.025 mSec cloudlet handling time for small volume cloudlet. Thereby it is measured that CW-MTLB outperforms Hy-RRSH-L by approximately 3.43%, whereas when compared with RRS protocol, CW-MTLB exhibited 50.23% improvement over RRS protocol. Similarly, compared to RRH and Hy-RRSH-H, the proposed CW-MTLB exhibits superior performance improvement, which lies in between (3.43%-50.23%).

For medium volume cloudlet, the maximum cloudlet handling time is observed in RRH, which is approximately 3.027, whereas the proposed CW-MTLB outperforms this model with 66.66%. On the other hand, the computational performance of Hy-RRSH-L exhibits a closer outcome as CW-MTLB, which is 1.015 cloudlet handling time for medium volume cloud. Here the performance improvement shows that CW-MTLB outperforms Hy-RRSH-L by .59%. The observation was also carried out for two other different protocols, such as RRS and Hy-RRSH-H, where it is found that the proposed system attains performance improvement approximately in between (.059%-66...66%). The behavioral analysis corresponding to cloudlet handling time is also assessed in the presence of a large volume cloudlet, which shows that RRH attain maximum cloudlet handling time, which is 4.036 mSec as compared to the other approaches. Also, here CW-MTLB outperforms RRH by approximately 75% for large volume cloudlets. It is also observed that the smallest value of more considerable volume cloudlet handling time is obtained for Hy-RRSH-L, which is 2.023. Here it implies that CW-MTLB is subjected to better processing tasks as it exhibits a performance improvement of 50.64% over the Hy-RRSH-L protocol. For the other two protocols, such as RRS and Hy-RRSH, the formulated CW-MTLB approach exhibits performance improvement between approximately (50.64%-75%). The interpretation of the outcome shows that CW-MTLB outperforms the other approaches such as Hy-RRSH-L, Hy-RRSH-H, RRH, and RRS to a greater extent even though the incoming cloudlet volumes differ in size from small to large in the dynamic, multi-tenant cloud environment.

The visualization of the numerical outcome is shown in the following Figure 5 for computation for unused resources. A closer interpretation exhibits that among all the traffic management protocols apart from the proposed approach CW-MTLB, both Hy-RRSH-L and Hy-RRSH-H obtain better performance on a log scale. The behavioral performance analysis of unused resources shows that for small volume cloudlet, remaining unused resource in the context of CW-MTLB is found 1.369, which is relatively lesser compared to the closer outcome by Hy-RRSH-L, which is 2.335. Here CW-MTLB outperforms Hy-RRSH-L by 41.37%. It indicates that CW-MTLB utilizes the resources very efficiently.

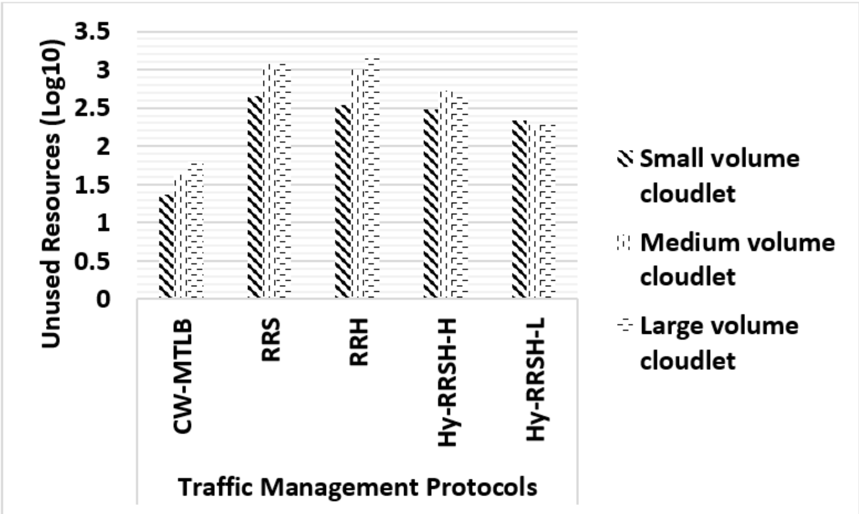

Figure 5. Computed outcome corresponds to the analysis of unused resources log (10)

On the other hand, it also outperforms the RRS by 48.5%. Whereas for small volume cloudlets, the formulated approach of CW-MTLB attains significant performance improvement over RRH and Hy-RRSH-H, where the quantified outcome is found between (41.37%-48.5%). However, for the other two types of cloudlets, the CW-MTLB system attains superior performance, which indicates that the load balancing works very efficiently and is subjected to perform better traffic management with maximum possible cost-effective resource utilization compared to the conventional baseline models.

The study further also performs extensive numerical computation corresponding to analyze the outcome of unused memory during the resource provisioning aspect in the scale of log10. The computation of unused resources (U_{Ri}) is taken place considering the following mathematical expression.

$$U_{Ri} \leftarrow \sum_{i=1}^{nHm} Cv(i) \quad (6)$$

$$Compute\text{: } \log 10\ (U_{Ri}) \quad (7)$$

It shows that memory management is quite superior in the case of CW-MTLB and it utilizes maximum possible memory for cloudlet processing and task allocation to the particular resources with proper resource mapping and virtualization in multi-tenant environment of CC. Here in the context of memory management RRH and RSS also performs better as compared to Hy-RRSH and Hy-RRSH-L.

5. CONCLUSION

Cloud computing (CC) is tremendously growing because of its higher scope of applicability, bringing new possibilities to leverage the enormous benefits of sharing information resources to various companies and subscribers. This research area envisioned providing on-demand resources and services to its users/subscribers over a network of clustered and distributed physical machines (PMs), usually the Internet. It also ensures the reliability of data delivery and distribution of scalable resources through service level agreement and data center (DC) placement. The study introduces a combined approach of resource scheduling in the cloud ecosystem with both CW-MTLB and DCVF-MT approaches. Both the approaches, i) DCVF-MT and ii) CW-MTLB, jointly addresses the problem corresponding to the cloud traffic management under the variable load of incoming job requests/load. The design and development of DCVF-MT analytically model the network functions that are well-capable for handling traffic locality problems in inter-edge and intra-edge conditions. For this purpose, the system balances the network load in such a way, where the virtual servers are allocated according to the user's requirement for task processing and execution. The system considers master user profiling which is quite a distinctive feature in this proposed study.

The system also implements two load balancing approaches, namely one-to-one and a mix mode approach of traffic load balancing in a cloud environment which finds its practicability due to managing both intra and inter traffic conditions with ease of execution for different zones. The virtual server place and availability of edge switches depending upon various traffic instances also make this model flexible. Another approach of cloud tenant-based load balancing is introduced in CW-MTLB. The design procedure for CW-MTLB is optimized for better resource allocation under three different traffic scenarios. It also ensures timely execution of resource mapping and sharing tasks under close and open forms of collaboration among computing resources under multi-tenancy. The performance outcome shows that it positively impacts the total simulation time, traffic management, and link proportion. Herewith limited bandwidth capacity, the DCVF-MT serves the maximum possible UBs with two different flexible modes of load balancing approach.

On the other hand, CW-MTLB attains 8.5133% performance improvement in overall cost optimization compared to the baseline approaches such as Hx-RRSH-L, Hy-RRSH-H, RRH, and RRS. It also accomplishes overall performance improvement on cloudlet processing time with 40.333%. It balances the overall performance of the system and makes it quite realistic in practice.

REFERENCE

[1] X. Li, Y. Wang, X. Li, and Y. Wang, "Parallel skyline queries over uncertain data streams in cloud computing environments," *International Journal of Web and Grid Services*, vol. 10, no. 1, 2014, doi: 10.1504/IJWGS.2014.058759.

[2] B. Yu, R. Sen, and D. H. Jeong, "An integrated framework for managing sensor data uncertainty using cloud computing," *Information Systems*, vol. 38, no. 8, pp. 1252–1268, Nov. 2013, doi: 10.1016/j.is.2011.12.003.

[3] I. Pietri and R. Sakellariou, "Mapping virtual machines onto physical machines in cloud computing: a survey," *ACM Computing Surveys*, vol. 49, no. 3, pp. 1–30, Dec. 2016, doi: 10.1145/2983575.

[4] C.-C. Lin, P. Liu, and J.-J. Wu, "Energy-aware virtual machine dynamic provision and scheduling for cloud computing," in *IEEE 4th International Conference on Cloud Computing*, Jul. 2011, pp. 736–737, doi: 10.1109/CLOUD.2011.94.

[5] L. B. A. Rabai, M. Jouini, A. B. Aissa, and A. Mili, "A cybersecurity model in cloud computing environments," *Journal of King Saud University - Computer and Information Sciences*, vol. 25, no. 1, pp. 63–75, Jan. 2013, doi: 10.1016/j.jksuci.2012.06.002.

[6] L. B. A. Rabai, M. Jouini, M. Nafati, A. Ben Aissa, and A. Mili, "An economic model of security threats for cloud computing systems," in *2012 International Conference on Cyber Security, Cyber Warfare and Digital Forensic (CyberSec)*, Jun. 2012, pp. 100–105, doi: 10.1109/CyberSec.2012.6246112.

[7] S. Bibi, D. Katsaros, and P. Bozanis, "Business application acquisition: on-premise or SaaS-based solutions?," *IEEE Software*, vol. 29, no. 3, pp. 86–93, May 2012, doi: 10.1109/MS.2011.119.

[8] J. Espadas, A. Molina, G. Jiménez, M. Molina, R. Ramírez, and D. Concha, "A tenant-based resource allocation model for scaling software-as-a-service applications over cloud computing infrastructures," *Future Generation Computer Systems*, vol. 29, no. 1, pp. 273–286, Jan. 2013, doi: 10.1016/j.future.2011.10.013.

[9] A. Jasti, P. Shah, R. Nagaraj, and R. Pendse, "Security in multi-tenancy cloud," in *44th Annual 2010 IEEE International Carnahan Conference on Security Technology*, Oct. 2010, pp. 35–41, doi: 10.1109/CCST.2010.5678682.

[10] L. Wu and C. Yang, "A solution of manufacturing resources sharing in cloud computing environment," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6240, Springer Berlin Heidelberg, 2010, pp. 247–252, doi: 10.1007/978-3-642-16066-0_36.

[11] A. Shekhar and D. Sharvani., "An extensive survey over traffic management/load balance in cloud computing," *International Journal of Advanced Computer Science and Applications*, vol. 8, no. 3, 2017, doi: 10.14569/ijacsa.2017.080338.

[12] D. A. Shafiq, N. Z. Jhanjhi, and A. Abdullah, "Load balancing techniques in cloud computing environment: a review," *Journal of King Saud University - Computer and Information Sciences*, Mar. 2021, doi: 10.1016/j.jksuci.2021.02.007.

[13] A. Adaniya, "A proposed load balancing algorithm for maximizing response time for cloud computing environment using fusion algorithm," *International Journal for Research in Applied Science and Engineering Technology*, vol. 7, no. 4, pp. 2911–2914, Apr. 2019, doi: 10.22214/ijraset.2019.4490.

[14] A. Ali and M. M. Iqbal, "A cost and energy efficient task scheduling technique to offload microservices based applications in mobile cloud computing," in *IEEE Access*, vol. 10, pp. 46633-46651, 2022, doi: 10.1109/ACCESS.2022.3170918.

[15] M. Alam and Z. A. Khan, "Issues and challenges of load balancing algorithm in cloud computing environment," *Indian Journal of Science and Technology*, vol. 10, no. 25, pp. 1–12, Jun. 2017, doi: 10.17485/ijst/2017/v10i25/105688.

[16] S. Ghosh and C. Banerjee, "Priority based modified throttled algorithm in cloud computing," in *International Conference on*

Inventive Computation Technologies (ICICT), Aug. 2016, vol. 2016, pp. 1–6, doi: 10.1109/INVENTIVE.2016.7830175.
[17] N. X. Phi, C. T. Tin, L. N. Ky Thu, and T. C. Hung, "Proposed load balancing algorithm to reduce response time and processing time on cloud computing," *International journal of Computer Networks and Communications*, vol. 10, no. 3, pp. 87–98, May 2018, doi: 10.5121/ijcnc.2018.10307.
[18] H. Chen, Q. Liu, and Q. Ai, "A new heuristic scheduling strategy LBMM in cloud computing," in *8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Aug. 2016, vol. 1, pp. 314–317, doi: 10.1109/IHMSC.2016.28.
[19] S. A. Ali, S. Khan, and M. Alam, "Resource-aware min-min (RAMM) algorithm for resource allocation in cloud computing environment," *International Journal of Recent Technology and Engineering*, vol. 8, no. 3, pp. 1863–1870, Sep. 2019, doi: 10.35940/ijrte.C5197.098319.
[20] A. Thakur and M. S. Goraya, "A taxonomic survey on load balancing in cloud," *Journal of Network and Computer Applications*, vol. 98, pp. 43–57, Nov. 2017, doi: 10.1016/j.jnca.2017.08.020.
[21] W. Hashem, H. Nashaat, and R. Rizk, "Honey bee based load balancing in cloud computing," *KSII Transactions on Internet and Information Systems*, vol. 11, no. 12, pp. 5694–5711, Dec. 2017, doi: 10.3837/tiis.2017.12.001.
[22] W. Saber, W. Moussa, A. M. Ghuniem, and R. Rizk, "Hybrid load balance based on genetic algorithm in cloud environment," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 11, no. 3, pp. 2477–2489, Jun. 2021, doi: 10.11591/ijece.v11i3.pp2477-2489.
[23] P. Zhang and M. Zhou, "Dynamic cloud task scheduling based on a two-stage strategy," in *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 2, pp. 772-783, Apr. 2018, doi: 10.1109/TASE.2017.2693688.
[24] P. G. B. and P. Jayarekha, "An efficient resource sharing technique for multi-tenant databases," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 10, no. 3, pp. 3216–3226, Jun. 2020, doi: 10.11591/ijece.v10i3.pp3216-3226.
[25] C. A. Shekhar and G. S. Sharvani, "MTLBP: a novel framework to assess multi-tenant load balance in cloud computing for cost-effective resource allocation," *Wireless Personal Communications*, vol. 120, no. 2, pp. 1873–1893, Sep. 2021, doi: 10.1007/s11277-021-08541-w.
[26] P. Xu, G. He, Z. Li, and Z. Zhang, "An efficient load balancing algorithm for virtual machine allocation based on ant colony optimization," *International Journal of Distributed Sensor Networks*, vol. 14, no. 12, Dec. 2018, doi: 10.1177/1550147718793799.
[27] J. Duan and Y. Yang, "A load balancing and multi-tenancy oriented data center virtualization framework," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 8, pp. 2131–2144, Aug. 2017, doi: 10.1109/TPDS.2017.2657633.
[28] A. S. Chandrashekhar and S. G. Suryakanth, "MLBM-DVF: modeling load balancing and multitenancy-oriented datacenter virtualization framework," *Global Journal of Engineering Science and Researches*, pp. 188–197
[29] Z. Khan, M. Alam, and R. A. Haidri, "Effective load balance scheduling schemes for heterogeneous distributed system," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 7, no. 5, pp. 2757–2765, Oct. 2017, doi: 10.11591/ijece.v7i5.pp2757-2765.

BIOGRAPHIES OF AUTHORS

Amith Shekhar Chandrashekhar is research scholar, pursuing Ph.D. in VTU under the guidance of Dr. Sharvani, Associate Professor, RVCE, Bengaluru. He is currently working as an Assistant Professor in the department of Information science and Engineering, GM Institute of Technology, Davangere. He is having around 10 years of teaching experience with area of interest as Cloud computing. He is having 4 international journal publications and presented his works in 3 national and 1 international conferences. He can be contacted at email: amithsc@gmit.ac.in.

Sharvani Gunakimath Suryakanth is currently working as an Associate Professor in the Department of Computer Science, RV College of Engineering, Bangalore, India. She has completed Ph.D. in Computer Science from Avinashilingam University for Women, Coimbatore, in the year 2014. She is having around 15 years of experience in teaching and 2 years of industry experience. Her research interests are cloud computing, internet of things, swarm intelligence, grid computing. She has guided more than 50 UG and PG Projects. She had presented papers in more than 20 national/International conferences along with 20+ reputed journal publications. She can be contacted at email: sharvanigs@rvce.edu.in.