IJARETY

International Journal of Advanced Research in Education and TechnologY (IJARETY)

Volume 12, Issue 2, March-April 2025

Impact Factor: 8.152

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INNO SPACE SJIF Scientific Journal Impact Factor

doi cross ref

निस्केयर NISCAIR

International Journal of Advanced Research in Education and TechnologY(IJARETY)

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||

DOI:10.15680/IJARETY.2025.1202044

A Load Balancing Algorithm Designed for Data Centres to Enhance the Efficiency of Cloud Computing Tasks

Byravarapu Prashant[1], Gullapalli Sravani[2], Jogi Naga Vamsi Ram[3], Gubbala Vamsi Priya[4]

Associate Professor & HOD, Department of CSE-Data Science, Eluru College of Engineering & Technology, Eluru, India[1]

B. Tech Student, Department of CSE, Eluru College of Engineering & Technology, Eluru, India[2,3,4]

ABSTRACT: Despite the extensive research conducted in the field of Cloud Computing, challenges still persist regarding workload balancing in cloud-based applications, particularly within the Infrastructure as a Service (IaaS) model. Efficient task allocation is vital in cloud computing due to the limited availability of resources and virtual machines. IaaS is a model that manages the backend infrastructure, including servers, data centres, and virtual machines. Cloud Service Providers must ensure optimal service delivery performance in these models to prevent issues like host overload or underload, which can lead to increased execution times or machine failures. Task Scheduling plays a significant role in load balancing, and it is essential that task scheduling aligns with the requirements outlined in the Service Level Agreement (SLA), a document provided by cloud developers to users. Key SLA parameters, such as deadlines, are considered in the load balancing algorithm. The proposed algorithm aims to optimize resource usage and enhance load balancing by taking into account Quality of Service (QoS) task parameters, VM priorities, and resource allocation. This load balancing algorithm addresses the identified challenges and fills the current research gap based on findings from the literature. Results indicate that the proposed algorithm achieves an average resource utilization of 90% compared to the existing Dynamic LBA algorithm, while also demonstrating improved performance in terms of reduced execution time and makespan.

KEYWORDS: Cloud Computing, Service Level Agreement, Load Balancing Algorithm and Performance.

I. INTRODUCTION

As we shift more towards online storage and services, Cloud Computing technology becomes an essential part of the business. This technology provides services through various kinds such as in software via web browsers, in Platforms such as designing and developing cloud-based applications. In the Infrastructure, the backend is managed by Cloud Service Providers (CSPs) such as maintaining Data Centres, servers, etc. Although there exist many other service delivery models in this technology, however, in this research, the focus is on the Infrastructure as a Service (IaaS) model. It deals with the server-side of this technology for resource allocation.

Virtualization is the backbone and essential feature of cloud-based applications. This technique can significantly affect the performance of the scalable and on-demand services provided to clients if the migration process and allocation of virtual machine resources are handled inefficiently. According to, cloud performance is proved to be in the top three Cloud Computing challenges. This research aims to enhance resource allocation in the IaaS model; this concept is fundamental as it deals with the balancing of resources provided to clients and the workload/user requests on servers.
The cloud users access services by sending requests; these are represented in Virtual Machines (VMs) in the cloud environment. CSPs should deliver services that are beneficial to businesses and increase user satisfaction. Thus, the proposed Load Balancing algorithm is developed mainly focusing on the IaaS model out of the three service models in the cloud where authors deal with the Cloud Computing technology's backend, such as server workload.

There are two components in a typical cloud environment: the frontend is the user side, and it is accessible by connecting to the Internet. The backend side handles the cloud service models where the Data Center store multiple physical machines (known as servers). Incoming user requests are received from the application are dynamically scheduled, and through virtualization, the necessary resources are allocated to clients. The virtualization technique is also responsible for balancing the load in the entire system, scheduling, and efficient allocation of resources. CSPs and

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||

DOI:10.15680/IJARETY.2025.1202044

cloud users can leverage the advantage of virtualization as well as dynamic task scheduling techniques. Thus, efficient scheduling can highly reduce execution time and increase the ratio of resource utilization in cloud-based applications.

1.1 MOTIVATION

"With the exponential growth of cloud computing, data centers are facing unprecedented challenges in managing workload demands, ensuring scalability, and optimizing resource utilization. Traditional load balancing algorithms often struggle to efficiently distribute tasks across available resources, leading to decreased performance, increased latency, and wasted energy. To address these limitations, this work aims to design and develop an innovative load balancing algorithm tailored specifically for data centers, with the goal of significantly enhancing the efficiency of cloud computing tasks. By leveraging advanced mathematical modeling, machine learning techniques, and real-time monitoring, our proposed algorithm seeks to optimize resource allocation, minimize response times, and maximize overall system throughput, ultimately enabling data centers to deliver high-performance, scalable, and sustainable cloud computing services."

1.2 PROBLEM DEFINITION

"The efficient distribution of workload across available resources in data centers is a complex problem, exacerbated by the dynamic and heterogeneous nature of cloud computing environments. Existing load balancing algorithms often struggle to adapt to changing workload patterns, leading to resource underutilization, increased latency, and decreased overall system performance. Furthermore, the lack of consideration for energy efficiency, scalability, and fault tolerance in traditional load balancing approaches can result in significant economic and environmental costs. Therefore, there is a pressing need for a novel load balancing algorithm that can efficiently and adaptively manage workload distribution in data centers, while also ensuring energy efficiency, scalability, and reliability."

1.3 OBJECTIVE OF THE WORK

"The primary objective of this work is to design and develop an efficient load balancing algorithm for data centers, with the following specific goals:

1. Improve Resource Utilization: Optimize workload distribution across available resources to minimize idle times and maximize resource utilization.
2. Enhance System Performance: Reduce response times and increase overall system throughput to improve the quality of cloud computing services.
3. Ensure Energy Efficiency: Minimize energy consumption and reduce the carbon footprint of data centers.
4. Scalability and Fault Tolerance: Develop an algorithm that can adapt to changing workload patterns and ensure continuity of services in the event of resource failures."

II. LITERATURE SURVEY

Cloud computing is a new technology which managed by a third party "cloud provider" to provide the clients with services anywhere, at any time, and under various circumstances. In order to provide clients with cloud resources and satisfy their needs, cloud computing employs virtualization and resource provisioning techniques. The process of providing clients with shared virtualized resources (hardware, software, and platform) is a big challenge for the cloud provider because of over-provision and under-provision problems. H. Shukur et al., highlighted some proposed approaches and scheduling algorithms applied for resource allocation within cloud computing through virtualization in the datacenter. The paper also aims to explore the role of virtualization in providing resources effectively based on clients' requirements. The results of these approaches showed that each proposed approach and scheduling algorithm has an obvious role in utilizing the shared resources of the cloud data center. The paper also explored that virtualization technique has a significant impact on enhancing the network performance, save the cost by reducing the number of Physical Machines (PM) in the datacenter, balance the load, conserve the server's energy, and allocate resources actively thus satisfying the clients' requirements. Based on our review, the availability of Virtual Machine (VM) resource and execution time of requests are the key factors to be considered in any optimal resource allocation algorithm. As a results of our analyzing for the proposed approaches is that the requests execution time and VM availability are main issues and should in consideration in any allocating resource approach.

Introduction Cloud computing is a new trend of computing where resources like storage, computation power, network, applications etc. are delivered as services. This services are available to the customers as subscription-based model i.e. pay-as-you go. M. Agarwal et al., proposed a model in which the customers can get these services on their demands regardless of where these services are hosted and customers have to pay depending on their usage of services. In cloud computing, resources are made virtual and unlimited. Also, the resources can be provisioned from anywhere i.e. always

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||

DOI:10.15680/IJARETY.2025.1202044

available at any location. So, cloud computing is a new paradigm where we can provision resources dynamically, deploy applications, and can access platform-independent services. Cloud computing, successor of internet computing, is a technology, where the concept of utility, scalability, on-demand services are incorporated. Figure 1 illustrates " Internet Computing " vs. " Cloud Computing ". Defining Cloud in IT According to the U.S. National Institute of Standards and Technology (NIST), Cloud is a classical model which enable omnipresent, convenient, on-demand network access to a publicly accessible pool of configurable resources like servers, storage, network components, applications; that can be accessed, manipulated and released with minimal management effort, less cost and minimal service provider interaction. Cloud computing can be defined by the following important properties. Service on demand: Cloud users can use services on their demands, whenever they need from any place and at any time without making any direct communication with cloud service provider. Wide network access: Services can be accessed over the network using different devices (like laptops, mobile phones, PDA, tablets, office computer etc.). Services can be provisioned in any platform, which means cloud services are platform independent. Pooled Resources: In cloud computing, resources are pooled together so that cloud providers can offer multi-tenant services. Multi-tenant supports multiple users to be served at a time with physical and virtual resources. These resources can be dynamically assigned and released according to the user's choice. Increased elasticity: There is no limit for provisioning resources via cloud. So services can be easily and quickly scale in and scale out. For example, an online shopping site uses the resources from the cloud in terms of users.

Security and performance are basic requirements for any system. They are considered the criteria for the measurement of any progress in a security system. Security is an indicator that affects the level of performance through the threats that influence the performance of parts of the cloud during the rendering of services. Both security and performance demonstrate the efficiency of cloud computing which indicates that the performance and security are measurements for the extent of the development of the cloud. N. Zanoon developed a method which gives the relationship between performance and security will be examined to know the extent of their impact on the progress of cloud computing.

The loss of business and downturn of economics almost occur every day. Thus technology is needed in every organization. Cloud computing has played a major role in solving the inefficiencies problem in organizations and increase the growth of business thus help the organizations to stay competitive. It is required to improve and automate the traditional ways of doing business. Cloud computing has been considered as an innovative way to improve business. Overall, cloud computing enables the organizations to manage their business efficiently. Unnecessary procedural, administrative, hardware and software costs in organizations expenses are avoided using cloud computing. Although cloud computing can provide advantages but it does not mean that there are no drawbacks. Security has become the major concern in cloud and cloud attacks too. Business organizations need to be alert against the attacks to their cloud storage. Benefits and drawbacks of cloud computing in business will be explored in C. T. S. Xue et al., work. Some solutions also provided in this paper to overcome the drawbacks. The method has been used is secondary research, that is collecting data from published journal papers and conference papers.

Cloud Computing (CC) is a fast growing services that make use of pay per use model. The technology provides various services in terms of storage, deployment, web services etc. however the expand of these services and the tremendous increase of user demand has resulted in many challenges to keep up the performance in line with QoS measurement and SLA document provided by cloud providers to enterprises. This expand resulted in challenges such as load balancing. Besides that, user's requirements became hard to fulfil in terms of response time and deadline regarding task scheduling. To address these challenges, D. A. Shafiq et al., proposed an optimized algorithm with the use of Machine Learning Classification technique based on deadline constraints. The main objective of the proposed algorithm is to enhance the efficiency, optimize the server resources by considering the priority of different users' tasks and avoid server breakdown. Our proposed algorithm will address the mentioned issues and current research gap based on the recent literature.

Scheduling or the allocation of user requests (tasks) in the cloud environment is an NP-hard optimization problem. According to the cloud infrastructure and the user requests, the cloud system is assigned with some load (that may be underloaded or overloaded or load is balanced). Situations like underloaded and overloaded cause different system failure concerning the power consumption, execution time, machine failure, etc. Therefore, load balancing is required to overcome all mentioned problems. This load balancing of tasks (those are may be dependent or independent) on virtual machines (VMs) is a significant aspect of task scheduling in clouds. There are various types of loads in the cloud network such as memory load, Computation (CPU) load, network load, etc. Load balancing is the mechanism of detecting overloaded and underloaded nodes and then balance the load among them. Researchers proposed various load balancing approaches in cloud computing to optimize different performance parameters. We have presented a

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||

DOI:10.15680/IJARETY.2025.1202044

taxonomy for the load balancing algorithms in the cloud. A brief explanation of considered performance parameters in the literature and their effects is presented by S. K. Mishra et al., in their work. To analyze the performance of heuristic-based algorithms, the simulation is carried out in CloudSim simulator and the results are presented in detail.

The tremendous growth of virtualization technology in cloud environment reflects the increasing number of tasks that require the services of the virtual machines (VMs). To balance the load among the VMs and minimizing the makespan of the tasks are the challenging research issues. Many algorithms have been proposed to solve the said problem. However, they lack in finding the potential information about the resources and tasks and it may lead to the improper assignment of the tasks to the VMs. M. Adhikari et al., proposed a new load balancing algorithm for Infrastructure as a Service (IaaS) cloud. We devise an efficient strategy to configure the servers based on the number of incoming tasks and their sizes to find suitable VMs for assignment and maximize the utilization of computing resource. We test the proposed algorithm through simulation runs and compare the simulation results with the existing algorithms using various performance metrics. Through comparisons, we demonstrate that the proposed algorithm performs better than the existing ones.

Resource scheduling becomes the prominent issue in cloud computing due to rapid growth of on demand request and heterogeneous nature of cloud resources. Cloud provides dynamism, uncertainty and elasticity based services to users in pay-as-you-go fashion over the internet. In recent decade, increase in requests (diverse and complex applications) for cloud services is raising the workload in cloud environment. Inefficient scheduling techniques face the challenges of resources being over utilized and underutilized (imbalanced) which leads to either degradation in service performance (in case of over utilized) or wastage of cloud resources (in case of underutilized). The basic idea behind the scheduling is to distribute tasks (diverse and complex nature) among the cloud resources in such a manner that scheduling algorithm avoids the problem of imbalance. Scheduling algorithm should also optimize the key performance indicator parameters like response time, makespan time, reliability, availability, energy consumption, cost, resource utilization etc. To fulfill the above-mentioned objective, many state of art scheduling algorithms have been proposed by M. Kumar et al., based upon heuristic, meta-heuristic and hybrid, reported in the literature. This paper provides the systematic review as well as classification of proposed scheduling techniques along with their advantages and limitations. We hope that our systematic and comprehensive survey work as a stepping stone for new researchers in the field of cloud computing and will be helpful for further development of scheduling technique.

III. SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Resource scheduling becomes the prominent issue in cloud computing due to rapid growth of on demand request and heterogeneous nature of cloud resources. Cloud provides dynamism, uncertainty and elasticity based services to users in pay-as-you-go fashion over the internet. In recent decade, increase in requests (diverse and complex applications) for cloud services is raising the workload in cloud environment. Inefficient scheduling techniques face the challenges of resources being over utilized and underutilized (imbalanced) which leads to either degradation in service performance (in case of over utilized) or wastage of cloud resources (in case of underutilized). The basic idea behind the scheduling is to distribute tasks (diverse and complex nature) among the cloud resources in such a manner that scheduling algorithm avoids the problem of imbalance. Scheduling algorithm should also optimize the key performance indicator parameters like response time, makespan time, reliability, availability, energy consumption, cost, resource utilization etc. To fulfill the above-mentioned objective, many state of art scheduling algorithms have been proposed based upon heuristic, meta-heuristic and hybrid, reported in the literature. This paper provides the systematic review as well as classification of proposed scheduling techniques along with their advantages and limitations. We hope that our systematic and comprehensive survey work as a stepping stone for new researchers in the field of cloud computing and will be helpful for further development of scheduling technique.

3.1.1 Disadvantages of Existing System

- **Imbalanced Resource Utilization**: Existing scheduling algorithms often lead to either over-utilization or under-utilization of cloud resources, resulting in performance degradation or wastage of resources.
- **Lack of Adaptability to Dynamic Workloads**: Many current techniques struggle to adapt to the dynamic and uncertain nature of cloud workloads, especially when dealing with sudden spikes or drops in demand.
- **Inadequate Optimization of Key Parameters**: Existing methods may not simultaneously optimize all key performance indicators such as response time, makespan, energy consumption, and cost, leading to suboptimal system performance.

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||

DOI:10.15680/IJARETY.2025.1202044

- **Scalability Issues**: Some scheduling algorithms do not scale well with the growing size and complexity of cloud infrastructure and applications, which affects their efficiency in large-scale data centers.

3.2 PROPOSED SYSTEM

The proposed load balancing framework aims to enhance resource allocation efficiency in cloud environments by addressing issues like unbalanced workloads, task rejection, and workload migration. It is structured in two layers. The **Top Layer** handles requests from various clients (both mobile and desktop) accessing the cloud via the internet. These requests are managed using the Cloudlet Scheduler Time Shared algorithm, which schedules tasks based on their arrival time, deadline, and completion time. Data Centers receive these tasks and forward them to the active Load Balancer, which applies the proposed algorithm to detect and manage SLA violations specifically focusing on workload migration when completion times exceed deadlines.

The **Bottom Layer** handles the allocation of tasks to Virtual Machines (VMs). If a VM violates the SLA by exceeding its deadline, the Load Balancer initiates a migration process, reallocating tasks to other VMs after reconfiguring their MIPS values. The allocation table is updated dynamically to reflect changes in VM status and task distribution. If there is no SLA violation (i.e., the time to complete a task is within the deadline), no migration is triggered. This framework supports dynamic scheduling and intelligent load balancing to optimize CPU usage and maximize cloud resource utilization.

3.2.1 Advantages of Proposed System

- **Improved Resource Utilization**: Ensures efficient distribution of tasks across VMs, reducing underutilization and overutilization of cloud resources.
- **Dynamic SLA Violation Handling:** Actively detects and mitigates SLA violations through intelligent workload migration and reallocation techniques.
- **Enhanced Scheduling Accuracy**: Considers key parameters like deadline and completion time to optimize task scheduling and avoid delays.
- **Supports Heterogeneous Environments**: Adapts to diverse client devices and workload types, making it robust for real-world cloud scenarios.

IV. MODULES

In this work we include three modules are used,

4.1 TASK SCHEDULAR

The Task Scheduler module is the core component responsible for managing and distributing incoming tasks (user requests) across available virtual machines (VMs) in the cloud environment. It plays a pivotal role in achieving efficient load balancing by ensuring that tasks are assigned to VMs in a manner that minimizes execution time, prevents resource overload or underutilization, and adheres to Service Level Agreement (SLA) parameters such as deadlines. In this project, the Task Scheduler implements the proposed Load Balancing Algorithm (LBA), which operates in a dynamic cloud environment where user requests arrive in random order and vary in size.

The Task Scheduler leverages the Cloudlet Scheduler Time Shared algorithm as its foundation, scheduling tasks based on two critical parameters: Deadline and Completion Time. It receives user requests from the User module via the top layer of the proposed framework and interacts with the Cloud module to allocate these tasks to VMs. The scheduler continuously monitors the status of VMs, identifying potential SLA violations (e.g., when a VM's completion time exceeds its deadline). In such cases, it triggers a migration technique, redistributing the workload to underutilized VMs by reconfiguring their MIPS (Million Instructions Per Second) to optimize resource usage. The Task Scheduler updates an allocation table to track VM status, workload distribution, and the number of assigned requests, ensuring a balanced and efficient system. This module directly contributes to reducing makespan and enhancing resource utilization, achieving an average utilization rate of 90% as demonstrated in the results.

4.2 USER

The User module represents the front-end interface of the cloud system, facilitating interaction between clients and the cloud infrastructure. It is designed to handle requests from multiple users accessing the system through various devices, such as mobile phones, desktops, or tablets, over the Internet. This module operates within the top layer of the proposed framework and serves as the entry point for submitting tasks to the cloud environment.

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||

DOI:10.15680/IJARETY.2025.1202044

Users generate requests that vary in complexity and resource requirements, such as computational tasks, data storage needs, or application-specific operations. These requests are forwarded to the Task Scheduler module for processing and allocation. The User module ensures seamless connectivity to the cloud services, enabling clients to access scalable, on-demand resources provided by the IaaS model. It is designed to support a diverse user base, reflecting real-world scenarios where requests arrive dynamically and unpredictably. By interfacing with the Task Scheduler, the User module ensures that client requests are efficiently queued and scheduled, contributing to the overall goal of enhancing system throughput and user satisfaction. Additionally, this module provides feedback to users regarding task completion, ensuring transparency and reliability in service delivery.

4.3 CLOUD

The Cloud module encompasses the backend infrastructure of the system, managing the physical and virtual resources necessary to execute user tasks. It operates within the bottom layer of the proposed framework and includes data centers, physical machines (servers), and virtual machines (VMs) that form the computational backbone of the IaaS model. This module is responsible for resource provisioning, virtualization, and workload execution, working in tandem with the Task Scheduler to achieve efficient load balancing.

The Cloud module hosts a pool of VMs, each with configurable resources such as CPU capacity (measured in MIPS), memory, and storage. It receives scheduled tasks from the Task Scheduler and allocates them to appropriate VMs based on the load balancing algorithm's decisions. The module employs virtualization techniques to abstract physical resources into virtual entities, enabling dynamic scaling and efficient resource sharing among multiple tasks. When the Task Scheduler detects an SLA violation (e.g., an overloaded VM), the Cloud module facilitates workload migration by reallocating tasks to available VMs and adjusting resource configurations as needed. It maintains a high level of resource utilization (up to 90%, as per the project results) by ensuring that no VM remains idle or overburdened for extended periods. Additionally, the Cloud module interacts with the server health monitor and traffic monitor components to provide real-time data on VM performance and system status, enabling proactive load balancing and fault tolerance.

V. SYSTEM DESIGN

The load balancing algorithm is designed to distribute incoming traffic across multiple servers to ensure efficient use of resources, improve responsiveness, and maximize throughput. The algorithm aims to provide a highly available, scalable, and reliable solution for cloud computing tasks. The system consists of several components, including a load balancer, server cluster, traffic monitor, and server health monitor. The load balancer is the core component of the system, responsible for receiving incoming traffic and distributing it across multiple servers in the cluster based on the chosen load balancing algorithm. The traffic monitor is responsible for monitoring incoming traffic and providing feedback to the load balancer, while the server health monitor is responsible for monitoring the health and performance of each server in the cluster.

The load balancing algorithm used in this system is designed to provide a highly available, scalable, and reliable solution for cloud computing tasks. The algorithm takes into account factors such as server load, response time, and availability to select the most suitable server to handle incoming traffic. The system is designed to meet several requirements, including scalability, availability, performance, and security. The system should be able to scale horizontally to support increased traffic and server additions, and be available 99.99% of the time, with a maximum downtime of 5 minutes per year. The system should also be able to handle at least 10,000 concurrent connections with a response time of less than 1 second, and ensure the security and integrity of all traffic, encrypting all data using SSL/TLS and supporting secure authentication methods.

5.1 SYSTEM ARCHITECTURE

The system architecture consists of a load balancer, server cluster, traffic monitor, server health monitor, database, and API gateway. The load balancer receives incoming traffic and distributes it across multiple servers in the cluster based on the chosen load balancing algorithm. The traffic monitor provides feedback to the load balancer, while the server health monitor tracks the performance of each server. The database stores server information, and the API gateway provides a single entry point for incoming traffic, routing it to the load balancer. This architecture enables scalability, high availability, improved performance, and security, with features like encryption and secure authentication methods to protect against unauthorized access and data breaches. The architecture diagram is shown in figure 1a and 1b.

International Journal of Advanced Research in Education and TechnologY(IJARETY)

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||

DOI:10.15680/IJARETY.2025.1202044

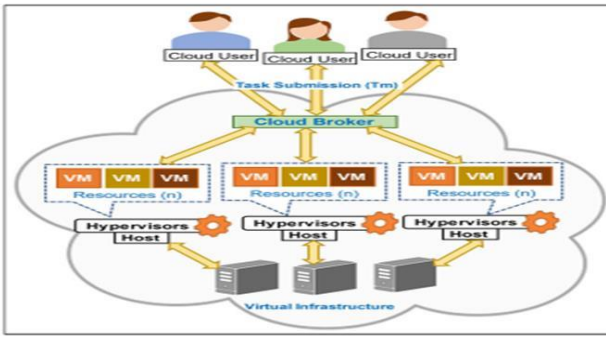

Fig 1a: The architecture diagram

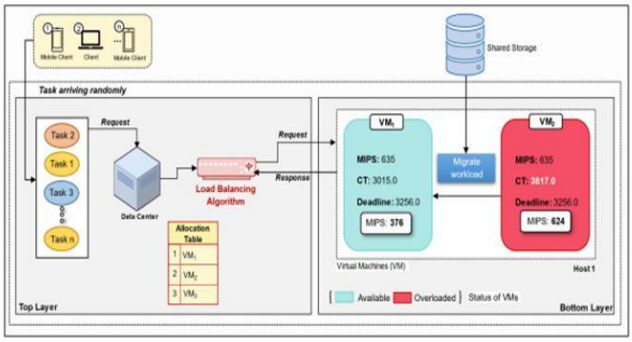

Fig 1b: The system architecture diagram

5.2 ALGORITHMS

The load balancing algorithm proposed in this project is designed to optimize resource allocation and task scheduling in cloud data centers, specifically within the Infrastructure as a Service (IaaS) model. It leverages dynamic scheduling techniques and migration strategies to enhance the efficiency of cloud computing tasks, addressing challenges like workload imbalance and Service Level Agreement (SLA) violations.

5.2.1 Proposed Load Balancing Algorithm (LBA)

The core of this project is a novel Load Balancing Algorithm (LBA) that operates within a two-layer framework to manage tasks in a dynamic cloud environment. It is tailored to handle random task arrivals, varying task sizes, and SLA constraints, ensuring efficient resource utilization and reduced execution times.

Key Components:

- **Cloudlet Scheduler Time Shared**: This scheduling mechanism assigns tasks (cloudlets) to virtual machines (VMs) in a time-shared manner, considering their arrival time, size, and SLA-defined deadlines. It forms the foundation for initial task distribution.
- **Load Balancer**: Acts as the primary decision-making entity in the top layer, monitoring VM performance and initiating workload migration when SLA violations occur (e.g., when completion time exceeds deadline).
- **Migration Mechanism**: Dynamically reallocates tasks from overloaded VMs to underutilized ones by reconfiguring their MIPS (Million Instructions Per Second), ensuring balanced resource usage.
- **Allocation Table**: A data structure that tracks VM status (e.g., "violated" or "normal"), the number of assigned tasks, and resource allocation details, updated after each migration or task completion.

The algorithm excels in dynamic environments by adapting to unpredictable workloads and prioritizing SLA compliance, achieving an average resource utilization of 90% compared to 78% for the existing Dynamic LBA.

5.2.2 Task Scheduling and SLA Monitoring

Before load balancing occurs, tasks are scheduled based on critical parameters derived from the SLA and VM performance.

- **Deadline**: The maximum allowable time for task completion, as specified in the SLA.
- **Completion Time**: Estimated time to finish a task on a VM, calculated based on task size and VM capacity (MIPS).
- **Monitoring Process**: The algorithm continuously compares the Time to Complete (TTC) with the deadline. If TTC ≤ Deadline, no action is taken; if TTC > Deadline, the VM is flagged as "violated," triggering load balancing actions.

This step ensures that the system proactively addresses potential SLA breaches, a key differentiator from static scheduling methods.

5.2.3 Workload Migration and Resource Reconfiguration

When an SLA violation is detected, the algorithm employs a migration strategy to redistribute the workload:

- **Violation Detection**: Identifies VMs where TTC exceeds the deadline (e.g., VM2 in the project example).
- **Target Selection**: Chooses an underutilized VM capable of handling additional load without violating its own SLA

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||

DOI:10.15680/IJARETY.2025.1202044

constraints.

- **MIPS Reconfiguration**: Adjusts the processing capacity of both the overloaded and target VMs (e.g., reducing MIPS on the source VM and increasing it on the target) to balance the load.
- **Task Transfer**: Moves excess workload to the target VM, ensuring seamless execution continuity.
- This process minimizes makespan (total task completion time) and prevents resource wastage, enhancing overall system efficiency.

5.2.4 Model Implementation and Environment

The algorithm is implemented using a robust technical stack to simulate and test its performance in a cloud environment:

- **Programming Language**: Python 3.10, leveraging its data processing and algorithmic capabilities.
- **Framework**: Django, enabling a web-based interface for user interaction (e.g., file uploads as tasks) and backend logic for scheduling and balancing.
- **Database**: MySQL, storing task metadata, VM states, and allocation tables.
- **Simulation**: Likely tested in a tool like CloudSim (common in cloud research), though not explicitly stated, to model VM behavior and workload dynamics.

The system supports a practical application where users upload files (tasks), and the algorithm schedules them across VMs, reflecting real-world cloud usage scenarios.

5.2.5 Performance Evaluation

The LBA's performance is evaluated against the existing Dynamic Load Balancing Algorithm using key metrics:

- **Resource Utilization**: Achieved 90% on average, a significant improvement over Dynamic LBA's 78%, indicating better resource exploitation.
- **Execution Time**: Reduced due to efficient task allocation and migration, though exact figures are not detailed in the provided excerpt.
- **Makespan**: Minimized by preventing VM overload and optimizing workload distribution.
- **SLA Compliance**: Fewer violations, as the algorithm dynamically adjusts to meet deadlines.

These metrics were assessed in a dynamic setting with random task arrivals and varying sizes, demonstrating the algorithm's robustness and adaptability.

5.2.6 Data Visualization and User Interface

To aid understanding and system management, the project includes visualization and interaction components:

- **Visualization**: While not explicitly detailed, similar projects often use tools like Matplotlib or Plotly to graph resource utilization, task completion times, and VM load distribution.
- **User Interface**: Built with HTML5, CSS3, and JavaScript via Django, allowing users to upload tasks (e.g., files) and view their processing status, enhancing usability for cloud service providers and end-users.

This feature bridges the technical algorithm with practical application, making it accessible for real-world deployment.

VI. RESULTS

The following figures Fig 2a to Fig 2j present the sequence of screenshots of the results.

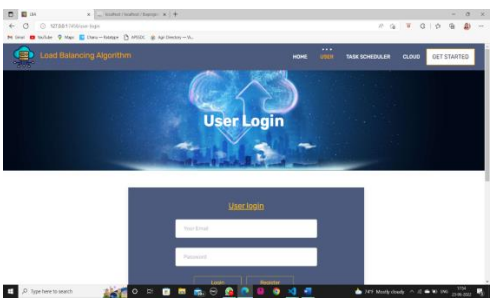

Fig 2a: Login page

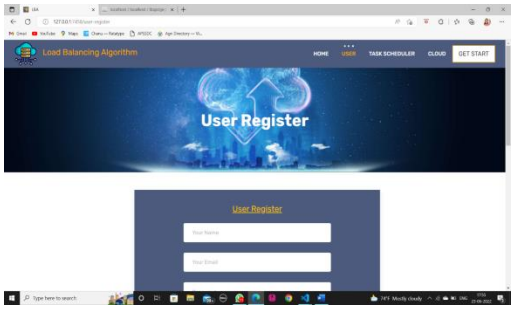

Fig 2b: Registration page

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||

DOI:10.15680/IJARETY.2025.1202044

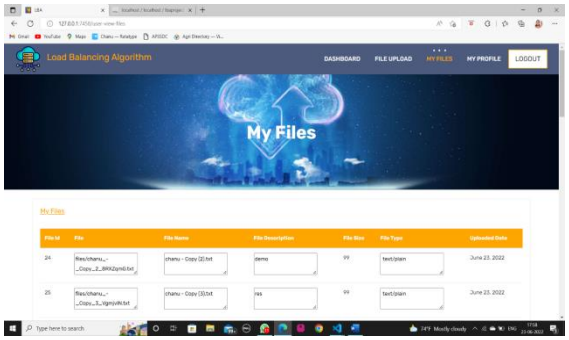

Fig 2c: Upload page

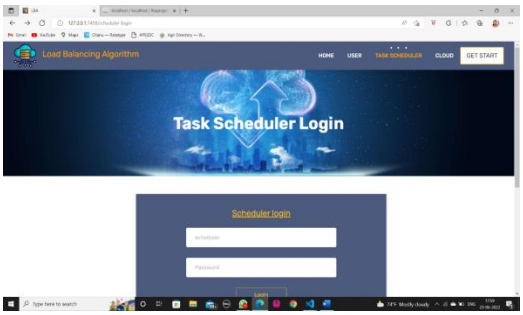

Fig 2d: Scheduler login

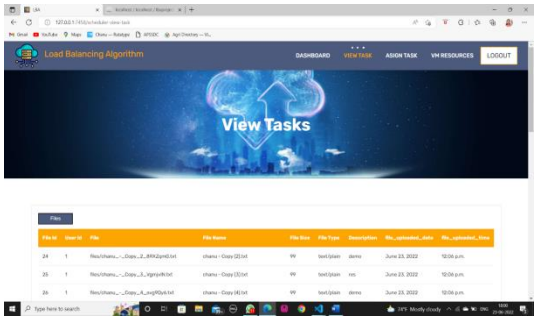

Fig 2e: View tasks

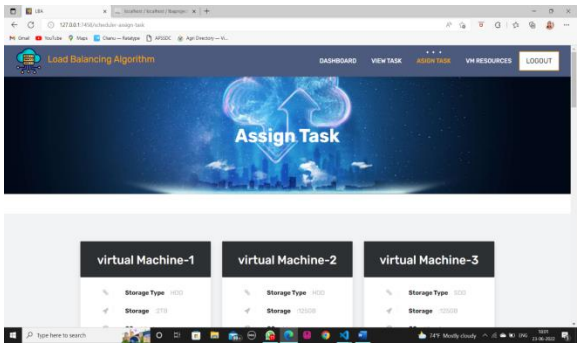

Fig 2f: Assign tasks

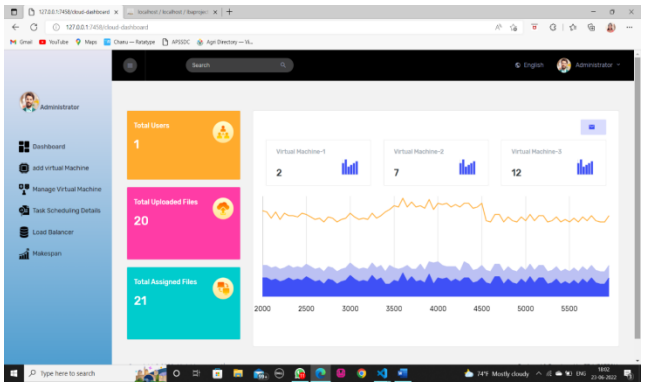

Fig 2g: Cloud dashboard

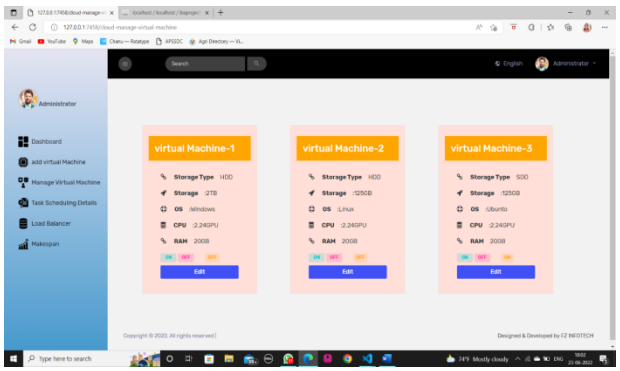

Fig 2h: Cloud manage virtual machine

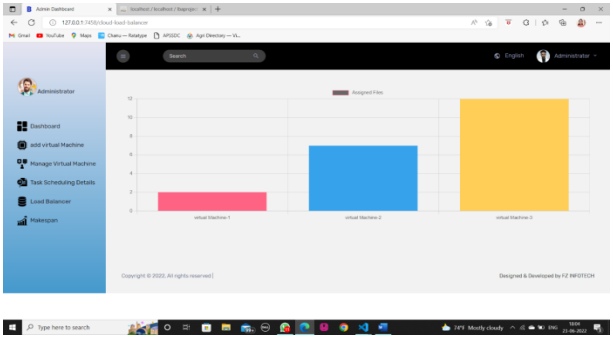

Fig 2i: Cloud Load Balancer

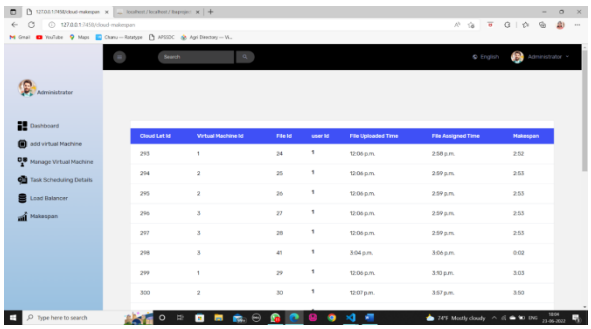

Fig 2j: Cloud make Span

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||

DOI:10.15680/IJARETY.2025.1202044

VII. CONCLUSIONS AND FUTUREWORK

7.1. CONCLUSION

This section concludes the paper by highlighting the findings and obtained results from the proposed LB algorithm. As we saw from the literature, task scheduling highly contributes to balancing the load in a cloud environment. Improving the Load Balancing process through Task Scheduling can result in efficient utilization of cloud resources. The objective of this paper was to provide an enhanced Load Balancing algorithm. Results proved that our algorithm reduces Makespan and provide efficient resource utilization of 90% compared to existing Dynamic LBA. It also shows that the proposed algorithm can function in a dynamic cloud environment where user requests arrive in random order and where there are many changes in the length of the user requests. The algorithm is also able to handle large size requests compared to the existing approach. The algorithm address SLA violation of VMs by reallocating resources to execute tasks efficiently. In the future, authors will work to optimize the cloud resources further and enhance cloud-based application performance, such as considering more SLA parameters. For example, the algorithm will be tested based on the number of violations and the migration count for better performance. Also, the algorithm will be comprehensively compared to other existing algorithms in the literature.

7.2 FUTURE WORK

While this project has successfully designed and evaluated a load balancing algorithm for enhancing the efficiency of cloud computing tasks in data centers, there are several avenues for future research and development. To further improve the algorithm's performance and adaptability, future work could focus on integrating machine learning techniques to dynamically adjust load balancing decisions based on real-time workload patterns and data center conditions. Exploring the application of this algorithm in edge computing environments and emerging cloud computing paradigms, such as serverless computing and containerization, could provide valuable insights into its versatility and scalability. By pursuing these research directions, we can continue to refine and expand the capabilities of our load balancing algorithm, ultimately contributing to more efficient, resilient, and sustainable cloud computing infrastructures.

REFERENCES

[1] H. Shukur, S. Zeebaree, R. Zebari, D. Zeebaree, O. Ahmed, and A. Salih, ‘‘Cloud computing virtualization of resources allocation for distributed systems,’’ J. Appl. Sci. Technol. Trends, vol. 1, no. 3, pp. 98–105, Jun. 2020, doi: 10.38094/jastt1331.

[2] M. Agarwal and G. M. Saran Srivastava, ‘‘Cloud computing: A paradigm shift in the way of computing,’’ Int. J. Mod. Educ. Comput. Sci., vol. 9, no. 12, pp. 38–48, Dec. 2017, doi: 10.5815/ijmecs.2017.12.05.

[3] N. Zanoon, ‘‘Toward cloud computing: Security and performance,’’ Int. J. Cloud Comput.: Services Archit., vol. 5, no. vol. 5, nos. 5–6, pp. 17–26, Dec. 2015, doi: 10.5121/ijccsa.2015.5602.

[4] C. T. S. Xue and F. T. W. Xin, ‘‘Benefits and challenges of the adoption of cloud computing in business,’’ Int. J. Cloud Comput.: Services Archit., vol. 6, no. 6, pp. 1–15, Dec. 2016, doi: 10.5121/ijccsa.2016.6601.

[5] D. A. Shafiq, N. Jhanjhi, and A. Abdullah, ‘‘Proposing a load balancing algorithm for the optimization of cloud computing applications,’’ in Proc. 13th Int. Conf. Math., Actuarial Sci., Comput. Sci. Statist. (MACS), Dec. 2019, pp. 1–6, doi: 10.1109/MACS48846.2019.9024785.

[6] S. K. Mishra, B. Sahoo, and P. P. Parida, ‘‘Load balancing in cloud computing: A big picture,’’ J. King Saud Univ.–Comput. Inf. Sci., vol. 32, no. 2, pp. 149–158, 2020, doi: 10.1016/j.jksuci.2018.01.003.

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

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

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

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

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

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

| ISSN: 2394-2975 | www.ijarety.in| | Impact Factor: 8.152| A Bi-Monthly, Double-Blind Peer Reviewed & Refereed Journal |

|| Volume 12, Issue 2, March-April 2025 ||

DOI:10.15680/IJARETY.2025.1202044

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

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

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

[16] M. A. Alworafi, A. Dhari, A. A. Al-Hashmi, and A. B. Darem, ‘‘An improved SJF scheduling algorithm in cloud computing environment,’’ in Proc. Int. Conf. Electr., Electron., Commun., Comput. Optim. Techn. (ICEECCOT), Dec. 2016, pp. 208–212, doi: 10.1109/ICEECCOT.2016.7955216.

[17] V. Lakra and D. K. Yadav, ‘‘Multi-objective tasks scheduling algorithm for cloud computing throughput optimization,’’ Procedia Comput. Sci., vol. 48, pp. 107–113, 2015, doi: 10.1016/j.procs.2015.04.158.

[18] J. H. Shanthan and L. Arockiam, ‘‘Resource based load balanced min min algorithm (RBLMM) for static meta task scheduling in cloud,’’ in Proc. IC-ACT, 2018, pp. 1–5.

[19] Thomas, G. Krishnalal, and V. P. Jagathy Raj, ‘‘Credit based scheduling algorithm in cloud computing environment,’’ Procedia Comput. Sci., vol. 46, pp. 913–920, 2015, doi: 10.1016/j.procs.2015.02.162.

[20] H. Gamal El Din Hassan Ali, I. A. Saroit, and A. M. Kotb, ‘‘Grouped tasks scheduling algorithm based on QoS in cloud computing network,’’ Egyptian Informat. J., vol. 18, no. 1, pp. 11–19, Mar. 2017, doi: 10.1016/j.eij.2016.07.002.

[21] S. Banerjee, M. Adhikari, S. Kar, and U. Biswas, ‘‘Development and analysis of a new cloudlet allocation strategy for QoS improvement in cloud,’’ Arabian J. Sci. Eng., vol. 40, no. 5, pp. 1409–1425, May 2015, doi: 10.1007/s13369-015-1626-9.

[22] R. Kaur and P. Luthra, ‘‘Load balancing in cloud system using max min and min min algorithm,’’ in Proc. Nat. Conf. Emerg. Trends Comput. Technol. NCETCT, vol. 1, 2014, pp. 31–34.

[23] Arunarani, D. Manjula, and V. Sugumaran, ‘‘Task scheduling techniques in cloud computing: A literature survey,’’ Future Gener. Comput. Syst., vol. 91, pp. 407–415, Feb. 2019, doi: 10.1016/j.future.2018.09.014.

[24] P. Kathalkar, ‘‘Challenges & issues in load balancing in cloud computing,’’ Int. J. Res. Appl. Sci. Eng. Technol., vol. 6, no. 4, pp. 963–968, Apr. 2018, doi: 10.22214/ijraset.2018.4163.

[25] S. Afzal and K. Ganesh, ‘‘A taxonomic classification of load balancing metrics: A systematic review,’’ in Proc. 33rd Indian Eng. Congr., Jan. 2019, pp. 85–90.

[26] Kodi, D. (2024). Data Transformation and Integration: Leveraging Talend for Enterprise Solutions. International Journal of Innovative Research in Science, Engineering and Technology, 13(9), 16876–16886. https://doi.org/10.15680/IJIRSET.2024.1309124

[27] S. Afzal and G. Kavitha, ‘‘Load balancing in cloud computing–A hierarchical taxonomical classification,’’ J. Cloud Comput., vol. 8, no. 1, p. 22, 2019, doi: 10.1186/s13677-019-0146-7.

[28] R. K. Naha and M. Othman, ‘‘Cost-aware service brokering and performance sentient load balancing algorithms in the cloud,’’ J. Netw. Comput. Appl., vol. 75, pp. 47–57, Nov. 2016, doi: 10.1016/j.jnca.2016.08.018.

[29] P. Kumar and R. Kumar, ‘‘Issues and challenges of load balancing techniques in cloud computing: A survey,’’ ACM Comput. Surv., vol. 51, no. 6, pp. 1–35, Feb. 2019, doi: 10.1145/3281010.

[30] Jindal, ‘‘Optimization of task scheduling algorithm through QoS parameters for cloud computing,’’ in Proc. ICAET, vol. 57, 2016, pp. 1–4, doi: 10.1051/matecconf/20165702009.

[31] Semmoud, M. Hakem, B. Benmammar, and J. Charr, ‘‘Load balancing in cloud computing environments based on adaptive starvation threshold,’’ Concurrency Comput., Pract. Exper., vol. 32, no. 11, pp. 1–14, Jun. 2020, doi: 10.1002/cpe.5652.

International Journal of Advanced Research in Education and Technology

ISSN: 2394-2975 Impact Factor: 8.152

www.ijarety.in editor.ijarety@gmail.com