National College of Ireland

Evaluating the Impact of Energy efficient Task Consolidation Techniques on Service Latency Performance in Multi-cloud Environments

MSc Research Project
Cloud Computing

Haritha Raguram
Student ID: 22110003

School of Computing
National College of Ireland

Supervisor: Mr.Rashid Mijumbi

National College of Ireland
Project Submission Sheet
School of Computing

National College of Ireland

Student Name:	Haritha Raguram
Student ID:	22110003
Programme:	Cloud Computing
Year:	2023
Module:	MSc Research Project
Supervisor:	Mr.Rashid Mijumbi
Submission Due Date:	14/12/2023
Project Title:	Evaluating the Impact of Energy efficient Task Consolidation Techniques on Service Latency Performance in Multi-cloud Environments
Word Count:	7067
Page Count:	20

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Haritha Raguram
Date:	30th January 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	☐
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	☐
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	☐

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Evaluating the Impact of Energy efficient Task Consolidation Techniques on Service Latency Performance in Multi-cloud Environments

Haritha Raguram
22110003

Abstract

This research focuses on the challenges of task consolidation in cloud computing, by analyzing it into five main domains. Initially, it focuses on analyzing task consolidation algorithms like the Green Task Consolidation Algorithm (GTCA) and Intelligence Workload Prediction Algorithms (IWPA), examining their effects on energy efficiency and service latency in cloud-based environments. The study then shifts to distributed optimization approaches as alternatives to centralized scheduling, evaluating their performance in reducing delays while enhancing energy efficiency and maintaining service levels. This research introduces and examines new energy-aware task consolidation techniques that incorporate intelligent load balancer, comparing them with conventional methods to determine their impact on service latency. It also examines resource allocation and task consolidation techniques for improved efficiency and scalability, analyzing their impact on service latency. The final aspect involves exploring capacity management techniques to optimize resource utilization and energy efficiency, assessing their effect on service latency and energy consumption. As a whole, the study highlights the need for improved thermal performance in cloud data centers, reviewing energy-aware task consolidation techniques and relevant literature. It highlights the crucial balance between quality of service and energy savings in cloud computing, offering insights into emerging strategies for a more adaptable cloud infrastructure.

1 Introduction

Cloud computing's rapid growth has completely changed the face of IT services, and organizations are depending more and more on cloud-based services to run their businesses. However, this paradigm shift has given rise to a fundamental concern: the effective utilization of resources within the data centers that use cloud services. The increased need for computer resources and energy means that the way resources are managed needs a reconsideration. The traditional way of allocating resources based on instant demand doesn't work well for dealing with the rising demands on data centers caused by the rapid growth of cloud services.(Beloglazov and Buyya; 2010). This shortcoming has led to requests for new and innovative strategies that go beyond basic functionality and resource allocation. The research focuses on resource allocation strategies, which are methods for allocating or distributing resources in advance based on predictions, trends, or predicted requests. (Gmach et al.; 2013). This transition is necessary not just for

resolving current challenges but also for supporting long-term efficiency and sustainability in cloud computing systems. The exponential growth of cloud services, as seen by the widespread use of virtualization technologies, has increased the demand for energy and computational resources. (Dastjerdi and Buyya; 2016). This has led to an important point when the standard models fall short, forcing studies into proactive resource allocation methodologies. By understanding the context and problems, this study aims to contribute to a radical change in how cloud computing resources are handled. The importance of this research comes from its ability to impact the current state of cloud-based resource management. As customer demand for cloud services grows inevitably, the need to manage resource conflicts, reduce interruptions, and improve overall system performance becomes more essential. Existing Resource allocation techniques, although useful to a certain degree, fall short of delivering complete answers to the various issues provided by the constantly changing characteristics of cloud systems. Its purpose is to give practical ideas for strengthening the overall resource administration approach in cloud-based systems. The study attempts to dive into the complexities of predictive resource allocation, analyzing its techniques, benefits, and limitations. By doing so, it intends to deliver significant insights that might shape future practices and techniques in the dynamic field of cloud computing. This study builds upon the research of (Zhang et al.; 2010), which analyses moving virtual machines in federation clouds for networking energy reduction. The problems and possibilities described in this paper contribute to the greater knowledge of how to allocate resources dynamically. The constant challenges of centralized scheduling and inefficient resource usage have uncovered constant inadequacies in present techniques, leading to poor resource allocation and energy consumption. This research focuses on five important categories, applying innovative and analytical techniques to overcome these difficulties. Preemptive resource allocation appears as a viable route for study and innovation, with the preemptive deployment of resources having the possibility of changing the efficiency of cloud data centers (Ma et al.; 2013). This not only solves immediate problems relating to efficiency constraints and resource shortages but also prepares cloud computing systems for long-term sustainability. The study aims to contribute to the set of information that goes beyond traditional methodologies and takes a proactive approach to resource management. This trend towards active resource allocation matches efficiently with the developing characteristics of cloud computing, wherein flexibility and efficiency are crucial. This research intends to deliver practical insights and approaches that may be quickly utilized in real-world cloud computing systems. By doing so, it seeks to encourage an evolution towards effective resource management, impacting industry best practices. This research revolves around an important question—how can preemptive resource allocation algorithms be efficiently applied to minimize energy usage and improve system effectiveness in cloud computing environments? This challenge drives the study to conduct a thorough examination of existing techniques to identify gaps in current studies and propose alternate approaches to predictive resource allocation. The research investigates the Green Work Consolidation Algorithm (GTCA) and Intelligence Workload Forecasting Algorithms (IWPA) in the field of task consolidations. These well-known task consolidation techniques are investigated to establish their impact on energy efficiency and service latency, both of which are critical components of cloud-based customer support. The paper investigates distributed optimization strategies to overcome the limitations of centralized scheduling. The research compares the performance of these distributed alternatives in terms of delay to centralized systems, stating their ability to maintain acceptable service levels

while enhancing energy efficiency. In the field of energy-aware task consolidation, various strategies are created and researched. These energy-aware task consolidation strategies, combining intelligent workload forecasting, are statistically tested for their impact on service latency performance in contrast to traditional methods. To improve efficiency and scalability, the study proposes distributed project schedulers. The primary focus is on understanding how different schedulers affect service latency in a distributed system capable of simultaneously processing several workloads. Capacity management methods are being researched to improve resource utilization and energy efficiency. The efficiency of these strategies is examined by considering their influence on service delay and energy usage. The study utilizes simulation, analytical, and experimental methodologies to examine trade-offs between performance and energy consumption, with a special focus on service latency. The ultimate objective is to contribute to a complete conceptual framework for energy-aware task consolidation, including practical implementation proposals in cloud computing contexts. Recognizing the requirement for a balance between service latency performance and energy economy, the suggested method intends to give significant insights for practical applications in cloud computing. The paper is divided into 6 sections, Literature Review, Methodology, Design Specification, Implementation, Evaluation, and Conclusion and Future Works.

2 Related Work

Cloud computing is now a fundamental part of modern computers. However, the fast growth of cloud-based services has made people worry about how efficiently data centers utilize their resources and the amount of energy they are using. This literature review is focused on finding solutions to these problems by looking at the most significant research that supports making systems energy efficient. The growth in cloud services has increased data center energy demand, leading to a need to research about energy efficiency and system performance in modern cloud computing. This research not only looks at what has already been done on this subject but also offers new ways to deal with and fix these important problems. Task consolidating is another essential component of the cloud computing ecosystem's energy efficiency optimization. It involves relocating and distributing virtual machines (VMs) to cut down on energy use and waste. Two recognized methods in this field are the Intelligence Workload Prediction Algorithm (IWPA) and the Green Task Consolidation Algorithm (GTCA). A complete examination of GTCA was presented by (Beloglazov and Buyya; 2010), highlighted the technology's use in handling resources in virtualized cloud data centers. By optimizing resource allocation depending on workload parameters, GTCA uses dynamic virtual machine consolidation, which results in considerable energy savings. The research stated how important it is to dynamically assign resources based on workload needs, to reduce energy consumption and underutilization. Gmach et al. (2013) examined how well IWPA could predict the characteristics of a workload and effectively manage cloud resources. IWPA employs predictive models to predict upcoming workloads and execute preemptive resource allocation choices. The research demonstrated how IWPA may reduce resource excessive provisioning and competition for resources to improve energy efficiency. Decentralized optimization methods have gathered attention due to the limitations of centralized programming. These methods offer the advantage of quicker response times and environmentally sustainable cloud environments. To reduce the impact of network latency and locality, distributed

scheduling techniques are used to distribute tasks across multiple servers. (Dastjerdi and Buyya; 2016) spoke about the idea of fog computing, which is a distributed method that takes cloud computing to the network's edge. By relocating data storage and processing closer to the end users, fog computing seeks to minimise latency. This enhances the consumer's experience when using latency-sensitive applications, such as Internet of Things (IoT) devices, while simultaneously increasing the effectiveness of energy consumption. The concept of dynamic internet resource allocation via simulated clusters was presented by (Ma et al.; 2013). This approach makes effective use of virtualization technologies to distribute resources. The study underlined that by preventing capacity overload, virtual clustering and dynamic allocation reduce resource fragmentation and improve the use of energy. From the comparative study of (Foster et al.; 2008) towards the load balancing work of (Cervone et al.; 2014) a group of fundamental works in cloud computing offers a complete overview of Computing environments industries. (Foster et al.; 2008) begin the study by contrasting cloud computing with grid computing in every aspect and addressing what distinguishes both. But the paper misses a thorough examination of security issues and practical scenarios. This gap indicates the need for additional research to address these critical concerns and enhance the results' practical implementation.

To further improve energy efficiency, improvements to energy-aware task consolidation techniques are essential. To make better allocation choices, these techniques combine deep learning, adaptive workload prediction, and recognition of patterns.Using the Mapper technique, (Verma et al.; 2010) developed the idea of power-aware program placement. The method automatically puts Virtual Machines (VM) on the greatest number of power-efficient servers while still fulfilling their performance needs by taking into account the power use of the servers. This method minimises energy waste and assures service quality by optimising electrical consumption and effectiveness. The "Lookahead" control, which makes choices regarding virtual machine migration and resource allocation using predictive models, was covered by (Kusic et al.; 2007) in the context of virtualized systems. This proactive strategy lowers energy waste, increases overall system adaptability, and enables optimum resource utilisation. Sustainability and resource efficiency in cloud settings depend on effectively handling huge numbers of concurrent processes. One approach to this problem is the emergence of federated job schedulers. In their assessment of network virtualization,(Zhang et al.; 2010) spoke about the advantages of federated task scheduling in a virtualized network context. The research emphasised how crucial it is to coordinate network resources so that energy efficiency and service quality are both maximised. In their 2010 paper,(Wang et al.; 2010) highlighted the importance of federated computing platforms and offered a viewpoint on cloud computing. The research emphasised the significance of federated computing resource administration in optimising resource growth while upholding the conservation of energy. In cloud computing settings, capacity management solutions are essential for optimising resource utilisation as well as energy usage. A perspective on cloud computing and the significance of managing capacity in cloud data centers were provided by (Armbrust; 2010) . According to the research, proper capacity planning may reduce energy loss by preventing over-provisioning and making sure that resources are distributed to satisfy user needs. Power-aware machine management for heterogeneity multi-core machines has been described by (Chen; 2012). This study looked at how power-aware scheduling may be used to maximise energy efficiency, especially in server setups with many cores. The research demonstrated how wise scheduling choices might result in considerable energy savings. (Marinos and Briscoe; 2009) study proposes a collaborative model and redirects attention on public cloud computing. In the absence

of case studies or actual information, the model's applicability and practical use cannot be completely evaluated. Further research into practical uses and success assessments would improve the research's value and guide future research projects. (Armbrust et al.; 2009) study lays out the key concepts and aspects of cloud computing, thus serving as an essential work. However, the paper may have gained from a more detailed examination of the potential drawbacks and problems associated with embracing the cloud. Taking these aspects into account would provide academics, business leaders, and legislators with a more complete picture. (Cervone et al.; 2014)Cervone et al.'s 2014 research on cloud load balancing gives interesting information on a significant component of cloud computing. Given its benefits, the study's absence of a full review of the usable performance of various load balancing algorithms limits our understanding of how effective these techniques are in actual cloud computing environments.

(Zhang; 2010) help improve the knowledge of cloud computing by providing a current overview and outlining research challenges.(Buyya et al.; 2009) discover into the concept of the cloud as the "5th Utility," presenting an idea, resolving the hype, and looking into the realities of new IT platforms.(Kliazovich; 2012) make contributions to the area of energy-aware cloud computing.

2.1 What Gap is the research bridging?

This work addresses major gaps in current research on task consolidation and cloud computing, advancing the progress in understanding in several critical areas. Starting with a complete examination of the Green Task Consolidation Algorithm (GTCA) and Intelligence Workload Forecasting Algorithms (IWPA), the research intends to address a gap in the literature on task consolidation approaches. This investigation analyses the impact of these techniques on energy efficiency and service latency in cloud-based applications. By doing so, the research gives essential insights into the applications of these strategies, helping to the design of more efficient cloud-based systems (Beloglazov and Buyya; 2010; Gmach et al.; 2013). On the other hand, the research tackles a knowledge gap relating to networked optimization strategies, stressing the limitations of centralized scheduling. By examining and comparing distributed optimisation methods with centrally managed ones, the study attempts to deepen the understanding of scheduling strategies, shedding more light on their efficiency in reducing delays and enhancing energy consumption without compromising service quality (Ma et al.; 2013; Verma et al.; 2010). Additionally, the study intends to address the absence in the existing literature regarding workload predictions and their consequences on service latency. The work provides new strategies for energy-aware task consolidation, stressing the insufficient use of predictive capabilities in existing methodology. This gap is vitally addressed, setting the framework for more advanced and adaptable employment consolidation solutions (Dastjerdi and Buyya; 2016; Kusic et al.; 2007). Hence, the work adds to the field by addressing a research gap on distributed task planners, vital for boosting scalability and efficiency. The creation and evaluation of these planners get attention, revealing valuable insights that have been overlooked in the research thus far. The research also explores the manner in which various schedulers effect service latency, providing useful information (Zhang et al.; 2010). The papers by (Beloglazov and Buyya; 2010; Gmach et al.; 2013; Ma et al.; 2013; Verma et al.; 2010; Dastjerdi and Buyya; 2016; Kusic et al.; 2007; Zhang et al.; 2010), contribute foundational insights into energy-efficient resource management, workload prediction, fog computing, dynamic resource allocation, power-aware placement, virtual machine migra-

tion, and cloud computing perspectives.

2.2 Novelty and Innovation

With a focus on five key categories, this research seeks to discover innovative ways to overcoming challenges connected with cloud-based work aggregation. First, it will carry out a thorough examination of task accumulation computations, specifically the Intelligence Workload Forecasting Algorithms (IWPA) and the Green Task Consolidation Algorithm (GTCA), to assess their effects on assistance delay and energy efficiency within the framework of cloud-based user experiences. After, as a replacement to centralised scheduling, the research will look at dispersed optimisation techniques. It will evaluate how well these methods function in terms of latency and their capacity to preserve service levels while improving energy efficiency. Then, by combining intelligent workload prediction and creating and analysing new energy-aware task consolidation strategies, the study will compare the effectiveness of these approaches with conventional ones in terms of service latencies and effectiveness. To increase effectiveness and scalability, the research will also investigate the development of distributed application schedulers and assess how various planners affect service latency. Finally, in order to optimise resource utilisation and energy efficiency, the inquiry will examine capacity management strategies and evaluate their effects on energy consumption and service latency.
Addressed Research Question: How can Service Latency be minimized in multi-cloud environments using Task Consolidation Techniques?

3 Methodology

The research methodology consists of three steps namely Client to Server DNS Route Location System, Server to Server Communication system, Intelligent Load Balancer .

3.1 Client to Server DNS Route Location System:

The first part of our suggested system is aimed at streamlining cloud-based communication between clients and servers. The purpose of the Client to Server DNS Route Location System is to provide more effective communication between users and Virtual Computing Servers (VCS) instances. This innovative approach ensures that the communication channels are optimized based on proximity by taking into account the geographical position of both individuals and networking clusters. It gives expected locations for cluster installations, thereby minimizing consumer inclinations across continents. This strategy enhances the cloud-based user experience while reducing latency. A greedy technique has been suggested to improve DNS routing effectiveness by reducing the overall amount of communication hops to tackle these issues. By using an evolving method to choose the DNS resolver closest to the client, this method lowers latency and speeds up DNS query replies. To preserve the reliability of the service, the system takes into account the volume of traffic that the DNS servers are currently experiencing. This ensures that customers are routed to locations that aren't overburdened.

3.1.1 Customer-focused routing:

The innovation is in taking focused on customer strategy and acknowledging customers' varied network needs and regional dispersion. The suggested method seeks to optimize customer interaction by minimizing latencies and lowering the total number of communication hops by choosing DNS servers depending on how close they are to clients.

3.1.2 Load-Aware Routing:

The suggested method includes load-aware navigation in addition to closeness. The method automatically routes customers to servers with lesser load by taking into account the present demand on the DNS servers. This prevents server overloading and guarantees constant service quality. The constant adaptation to network circumstances helps make the DNS system adaptable and receptive.

3.1.3 Real-Time Networking Variation Adjustments:

The suggested approach's capacity to adapt in actual time to technological fluctuations is one of its main advantages. The process continually modifies DNS forwarding to ensure that clients are routed to the best servers when network circumstances change. This flexibility improves the domain name method's overall reactivity to changing customer needs and network conditions.

3.1.4 Effective use of Work station clusters:

Effective Use of Workstation Clusters because The suggested greedy approach optimizes the grouping of DNS resolutions within the center. The program make sure that machine clusters are used efficiently by choosing the best host depending on demand and vicinity factors. This minimizes latency and improves system efficiency by supporting the dependability and efficient response of DNS queries.

3.2 Server to Server Communication System

The next phase examines the Server to Server Communication System, which seeks to minimize latency and enhance the exchange of information between servers. This system introduces geo-acyclic graph movements and price functions designed to improve the flow of data and reduce the cost constraint on the infrastructure. The anticipated outcome is an advanced cloud service framework that delivers reduced latency and cost-effective operational efficiency.

3.3 Intelligent Load Balancer

The final component of the methodology is the Intelligent Load Balancer, which employs deep generative neuronal algorithms for resource distribution. By simulating various account privileges, it aims to extend processing times and minimize idle times, thus ensuring efficient deployment of resources. This strategic component is anticipated to directly contribute to increased revenue growth and heightened operational efficiency. Its main objective is to optimize job scheduling using artificial intelligence projections and previous process durations. By taking into account the nature of each activity and

its predicted length, the load balancer aims to reduce time spent processing, minimize resource conflicts, and improve system efficiency.

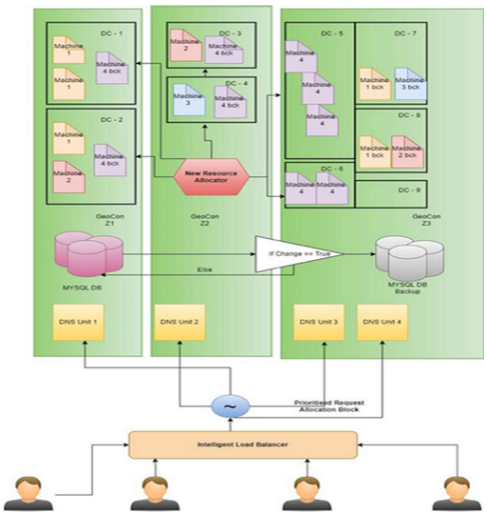

Figure 1: Research Methodology

The presented methodology diagram Figure 1 states that an intelligent load balancer receives the data that is provided by the user. With the help of the prioritization function, the Intelligent Load Balancer assigns priorities to the tasks. It pushes the data to DNS and sorts it according to the proposed prioritization rule. Both Domain Name System (DNS) and Database are organized into zones. Each zone has a resource allocator and an intelligent resource allocator. To reduce latency, it maintains the resource in such a way that users who have machines with configurations that are similar to one another are grouped in the same geographic location. The backup is kept in a different area to improve the backup and rack policy. This ensures that even in the event of an emergency, the data are safe and secure in the backup, which is kept in a location that is completely distinct from the original database. The method improves cloud computing by intelligently allocating work, structuring resources according to location, and securely storing data across several areas. This method provides effective resource utilization, prioritizes critical data processing, and protects data integrity.

4 Design Specification

The Hardware and Software specification needed for the research is stated in below Figure 2

4.1 Fast DNS Routing System

In the Cloud Network Sustainability, the Domain Name System (DNS) is essential because it functions as the foundation for converting machine-readable Internet Protocol (IP) addresses into human-readable domains. For consumers to effortlessly navigate websites and offerings, translating them is essential. In this setting, the structure that responds to client DNS queries is made up of the DNS servers and other network devices, which are usually kept in data centers. These machines, which are arranged in computer

Hardware Specification	- Operating System: Windows/Linux - CPU: Quad-core 2.5 GHz - RAM: 8 GB - RAM: 16 GB - Disk Space: 20 GB
Software Specification	- Python 3.8 - GeoPy Library (v1.22.0)
Network Specification	- DNS server bandwidth: 1 Gbps - Minimum 100 Mbps for user-server communication

Figure 2: Hardware and Software specifications

clusters inside the data center, receive DNS requests from clients, who stand in for users or gadgets. To provide a low-latency connection for a globally distributed user base, data centers are positioned carefully. There are issues with the DNS and the system's speed and effectiveness. Clients that vary in location and have distinct network needs may encounter different latencies based on how close the DNS resolving services are. Furthermore, DNS server demand varies, which may cause servers to become overloaded and result in a decline in service quality. It is possible that the conventional DNS routing techniques don't always account for these variables, which may lead to less-than-ideal customer service, higher latency, and poor performance. The suggested algorithm solves

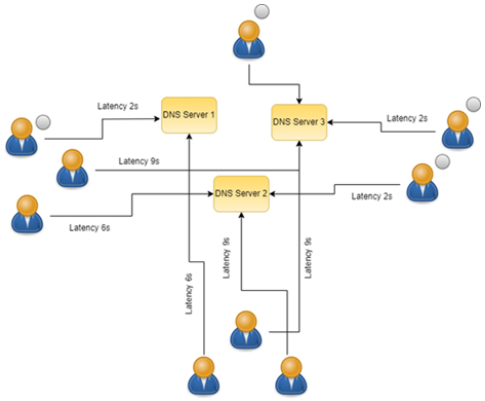

Figure 3: Proposed DNS Prioritization Architecture

the problems caused by a worldwide audience and changing server load. It focuses on customers and takes workload into account when sending DNS requests. A better DNS system has fewer delays and faster connections. This is possible because machine groups work well together and can adapt to changes in the network in real-time. The study's goal is to find a solution that will work for the DNS route issues the cloud ecosystem is currently having.

4.2 Intelligent Load Balancer

This intelligent load balancer's main objective is to optimize job scheduling using pattern recognition techniques. The load balancer seeks to decrease time spent processing, minimize resource disagreement, and enhance system efficiency by taking into account the nature of each activity and its anticipated length. They Compile previously collected information on resource utilization, system status indicators, and work durations.

This data will be used to train the machine learning model and sort jobs into different groups based on their characteristics, such as memory constraints, CPU constraints, or I/O constraints. Also to create an algorithm that uses supervised methods to predict task execution durations based on the kind of task, prior information, and the day's length. To create a precise predictive framework, this chooses a suitable machine learning approach, which includes neural network modeling. The Sorting of the jobs according to the estimated time of completion so that Longer jobs should be dispersed during off-peak hours, while shorter tasks should be planned during high-load hours. When setting priorities, dependency issues, deadlines, and job criticality are taken into account. The "Resource Allocation Algorithm with ML Predictions" is an advanced theoretical structure that combines real-time resource tracking, previous information evaluation, and machine learning (ML) forecasts to optimize the distribution of storage and processing power (CPU). By evaluating previous utilization behaviors and constantly modifying to real-time resource availability, this approach aims to reduce resource disputes, optimize system efficiency, and improve the user experience. This is accomplished by anticipating future resource requirements, allocating resources to projects as they arise, and ensuring that resources are rapidly reallocated after work is completed. This technology has the potential to significantly increase the responsiveness of computer environments, reduce operating costs, and maximize resource use. It demands precise neural network model forecasts, careful consideration of information privacy, and machine learning expertise for practical use.

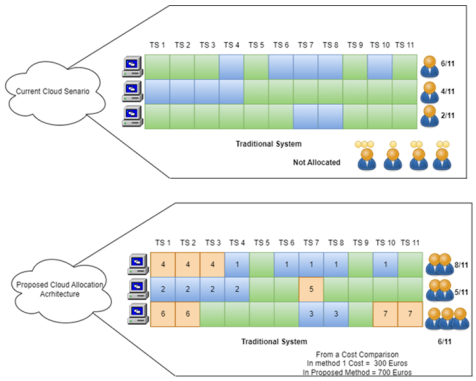

Figure 4: Proposed Intelligent Load Balancer Mechanism

Even though the stated method from Figure 4 has a lot of benefits, it also has certain drawbacks. The quality of the artificial intelligence algorithm has a big impact on how accurately it predicts resources, thus it must be trained and tested regularly. Data protection and user behavior control challenges must be resolved to ensure ethical execution. Additionally, using this technique might require a certain understanding, particularly in machine learning, which may be problematic for organizations with limited resources or experience. Despite these shortcomings, the "Resource Allocation Algorithm with ML Predictions" promises a more effective and adaptable resource allocation procedure by offering a theoretical framework for the development of workable resource management approaches across a wide range of systems.

5 Implementation

5.1 Fast DNS Routing System

The Fast DNS routing system simulates a scenario in which data centers (DCs), DNS servers, and user locations are produced at random. The main goal is to compare the costs and effectiveness of two different server selection strategies: aggressive and randomized. The calculation of the geographic separation between two points is done by the key 'calculate distance' that uses the Haversine formula. Following that, randomized user and DNS server locations are generated, together with the associated costs. A cost matrix is built to keep track of the total costs for each user and DNS server combination. The simulation then analyzes the cost of generation and assigns users priority passes at random. There is a total number of premium clients. Furthermore, a random number of DCs are generated. The random technique determines the mean delay, total income, and expenses sustained by the firm by randomly selecting DNS servers for each user. The pricing includes both user and server-related costs. In contrast, the new technique repeatedly selects the DNS server with the lowest overall cost for each user. The average delay, total revenue, and total cost spent by the firm are also computed for this technique. According to the training's theoretical analysis, the new strategy optimizes server selection by choosing the one with the lowest total cost for each user, whereas the random method incurs charges based on arbitrary server selections. The findings shed insight into the choices between the two strategies, taking variables such as income, latency, and total cost for the simulated circumstance into account.

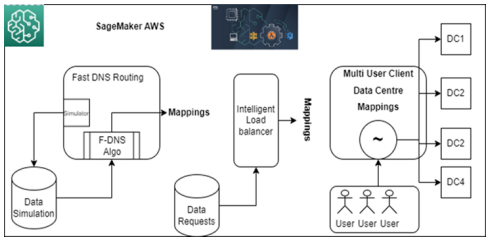

Figure 5: Proposed implementation Architecture

5.2 Intelligent Load Balancer

The following Programming environment is used to simulate a load balancing simulation including machines, processes, and users, as well as associated costs and time constraints. The architecture of the simulation balances the distribution of tasks across computers based on resource availability, taking machine utilization and execution durations into account. The code defines three classes: Machine, Process, and User. The Machine class represents a computational machine with initial and current resource capabilities. The Process class represents a computing job with predefined resource requirements and execution duration. The User class represents a user with related operations and an annual charge. The "generaterandomprocesses" method generates an arbitrary number of processes for a user, each with a randomized execution length and resource requirements. The "generatemachines" method returns a list of devices with randomly generated starting capacities. The "loadbalancewithcostandtimev2" method is the primary

load distribution method. Users with more processes are prioritized, and every activity is assigned to a suitable computer depending on time constraints and resource availability. The method generates allocation statistics and tracks machine usage. The modeling process begins with the "mainwithcostandtime" function, which requests user input for the number of people and computers. It publishes user data, builds machines, and assigns random processes to users. After that, the load balancing function is called, and an allocation breakdown and total cost achieved are reported. The simulation considers user charges, resource allocation, and machine utilization. By looping over users and activities, the load-balancing algorithm attempts to assign processes to available computers. It refreshes machine availability and utilization while accounting for execution time and resource restrictions.

5.3 Multi User Client Data center Mappings

A greedy provisioning mechanism is implemented, which generates a delay matrix and provides DataCenter subclasses. The DataCenter class represents a data center with a random machine capacity. The "generatelatencymatrix" function generates an asymmetrical latencies matrix that illustrates the transmission delay between many data centers. The "generatecapacityneeds" function generates random capacity requirements for a given user count. The "randomallocation" function assigns computers to users at random, taking into account the data center's capacity restrictions. The "printdatacentermatrix" function prints the allocation data as well as the total number of computers granted to each user. The function for greedy provisioning implements a greedy algorithm that prioritizes client allocation to data centers with low latency. When fully provisioning users in a particular data center is not possible, it investigates additional data centers. After the initial allocation, it reserves the remaining machines in data centers with available space for consumers. The "calculatelatencyloss" function calculates total latency and related losses using the resource allocation and latency matrix as input. The outcome demonstrates how the greedy provisioning engine distributes resources across users and decreases latency, providing insights into its effectiveness. The latency-related metrics and allocation information provide a comprehensive view of the simulation results.

6 Evaluation

6.1 DNS Mapping System

This research simulates an instance in which a company selects DNS servers in order to minimize latency, maximize profits, and minimize total expenses. The experiment generates random user and DNS server addresses, computes geographical distances, and compares two server selection strategies that are a greedy technique and a random strategy.

6.1.1 Configuration for Simulation:

The user accounts and DNS servers are first placed at random places in the experiment, each with unique latitudes, longitudes, and expenses. The initialization of the cost matrix records the expenses and distance among users and DNS providers. Furthermore, random profiles of consumers are created that show the cost of the pass and if a user has a priority pass. Data centers (DCs) with arbitrary locations are also included in the simulation.

6.1.2 Greedy Server Selection:

The DNS server is chosen by the simulation using a greedy method. The method calculates the overall cost for every DNS host by iterating over each user and taking into account both customer-specific prices and geographic distance. For every user, the DNS servers with the lowest overall cost is chosen. Metrics like average delay, overall revenue, and the total amount paid by the business due to the excessive strategy are included in the findings.

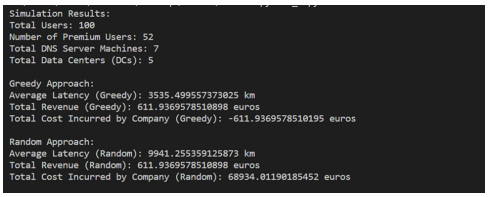

Figure 6: DNS Simulation Results

6.1.3 Random Server Selection:

In an arbitrary client-choosing technique, users are given servers at random. For this random technique, the algorithm computes the average delay, total revenue, and total cost borne by the firm. The simulation results are shown, highlighting important metrics for the random and greedy methods. There are reports on the overall number of users, premium users, DNS servers, and data centers. Median delay, total income, and total costs paid by the business are presented in the aggressive strategy. Similar information is provided for the random technique, including average delay, total revenue, and total costs spent by the business. The two server selection procedures may be compared thanks to the findings that have been supplied. The standard deviation latency statistic sheds light on how well servers are chosen in terms of reducing the distance between them. The business's overall cost includes both server-related and customer-specific expenses, while its total income represents the possible revenues from priority pass subscriptions.

6.2 Intelligent Load Balancer

This research simulates an instance of spreading workload in a computer environment where machines are assigned activities from multiple users. Throughout the simulation, random machines, users, and processes are generated, each with unique properties such as cores, RAM, memory, and execution time. This load-balancing technique's purpose is to assign jobs to machines as efficiently as possible while keeping resource and time constraints in mind. The key points and conclusions of the simulations could serve as a framework for the discussion.

6.2.1 Classes and Data Generation:

The Machines, procedure, and User classes are defined in the source code. These class examples stand in for the computers, different procedures, and users, in that order. The respective functions in the code, build a given quantity of machines with randomised beginning characteristics and produce processes that are random for a particular user.

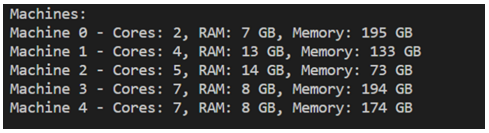

Figure 7: Machine Configuration for Simulation

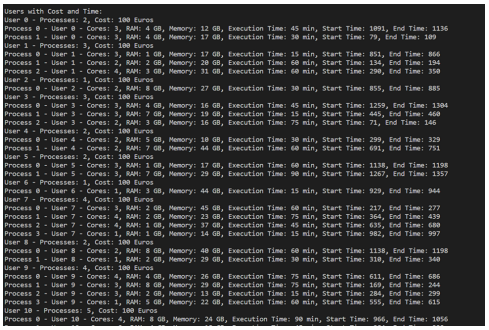

Figure 8: User Requirements and Processes

6.2.2 Load Balancing with Cost and Time:

The respectuve function carries out the load matching method, is the central component of the simulation. The amount of operations that each user has determines how they are arranged in decreasing order, and the accessibility of computers is monitored minute by minute. a combination of the availability of resources and time limitations, the function repeatedly runs all user tasks and tries to assign them to available servers. The resources and availability of the related machine are updated when a task is effectively assigned. A notification suggesting inadequate resources is displayed if an operation cannot be assigned.

6.2.3 Simulation Results:

After initialising users and machines and printing their characteristics, the main cost and time function invokes the workload balance algorithm. The printed findings provide an understanding of the structure's initial condition by giving details about the devices, users, and operations. Detailed notifications are issued for each allotment or sign of insufficient assets throughout the load levelling procedure.

6.3 Multi User Client Data center Mappings

The simulation encompasses two distinct strategies for load balancing in a simulated environment: a Greedy Approach and a Hybrid Proposition of Multi-User Center Provisioning. The code is organized into classes representing machines, processes, and users, with each entity characterized by specific attributes such as cores, RAM, memory, and execution time. The discussion will focus on each strategy separately. Different needs are generated at random for each user, and these operations are assigned to machines using the load balancing algorithm. This allocation approach requires a rapid allocation technique that overlooks collective optimization goals. This is accomplished by providing priority to the first machine that is available and meets the resource constraints. The

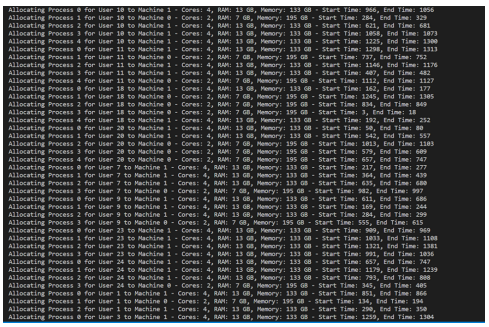

Figure 9: Allocation

```
Enter the number of data centers: 4
Enter the number of users: 20
```

Figure 10: Simulation Inputs

```
Data Center Information:
Data Center - 0: Total Machines - 42
Data Center - 1: Total Machines - 32
Data Center - 2: Total Machines - 47
Data Center - 3: Total Machines - 31

User Capacity Needs:
User 1: Needs 15 machines
User 2: Needs 7 machines
User 3: Needs 4 machines
User 4: Needs 8 machines
User 5: Needs 11 machines
User 6: Needs 12 machines
User 7: Needs 15 machines
User 8: Needs 8 machines
User 9: Needs 11 machines
User 10: Needs 22 machines
User 11: Needs 1 machines
User 12: Needs 5 machines
User 13: Needs 6 machines
User 14: Needs 15 machines
User 15: Needs 19 machines
User 16: Needs 4 machines
User 17: Needs 19 machines
User 18: Needs 16 machines
User 19: Needs 16 machines
User 20: Needs 4 machines
```

Figure 11: User and Datacenter mappings

simulation results include machine and user statistics that describe the distribution options made while the excessive provision method was running. The final presentation of loss metrics and overall delay provides a comprehensive grasp of the implications of this distribution approach.

The Greedy Approach highlights the distribution of resource options by revealing both examples of resource shortage and effective allocations. Although not as complicated as global optimisation, this strategy is effective in cases when allocation must occur rapidly and a more complex solution may not be required. The load balancing procedure

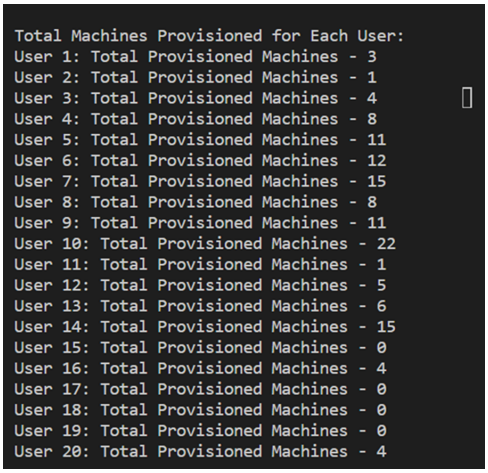

Figure 12: Greedy Provisioning

incorporates more complex factors according to the Hybrid Proposition . Two different approaches are investigated that are the load distribution with price and load balanced using expense and utilization limit. The load balance with cost function assigns computers to customers depending on the amount many tasks they own, taking into account their overall resource needs and related expenses. The load balance with cost and utilization threshold function, on the other hand, expands on this strategy by adding a utilization limit, guaranteeing that computers are not overworked during the allocation procedure. The Hybrid Proposition's model findings include machine visualization, the creation of consumers with arbitrary activities and related expenses, and load-balancing strategy implementation. Iteratively adding users continues until computers are operating at least 85By taking utilisation levels and cost into account, the Hybrid Proposal offers a more thorough method of load balancing. This makes it possible to allocate resources in a more intelligent and efficient manner, which strengthens the load balancing solution in situations when considerations other than speedy allocation are essential.

6.4 Discussion

The DNS Server Greedy approach sets an example in which users with different interests must select a DNS server to show how the cloud computing simulator works. This approach seeks for the most effective service by taking into consideration customer preferences, server costs, and geographical location. When cloud services are set up, this simulation illustrates how consumer preferences and location affect server choice. This method reduces the cost constraint by calculating the two metrics such as cost and geographical location. The benefits of this approach are a better user experience due to lower

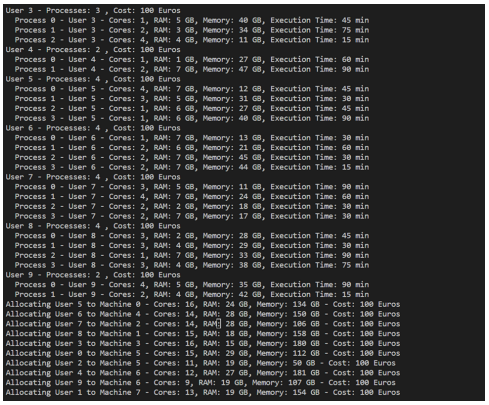

Figure 13: Multi-User Hybrid Allocation system

latency, improved server utilization, and possibly reduced expenses for the cloud service provider. When exploring cloud computing scenarios with global interdependence, the simulation's realistic and adaptable consideration of consumer preferences and geographical distances is essential. Intelligent Load Balancer simulates the load balancing of cloud computing systems while focusing on economic concerns. Its distinctive characteristics include costs for users, machine utilization, and dynamic resource allocation. The simulated scenario illustrates scenarios in which cloud service providers attempt to lower operating costs while managing the use of resources. The approach involves managing various user demands and tasks across many computers globally. Increased resource efficiency, reduced expenses, and more adaptable load-balancing solutions are among the advantages. This approach is suitable for real-world cloud computing environments due to its ability to continually manage prices and the use of resources, providing insights into effective cloud infrastructure management. Multi User Client Data Center mappings provide a cloud computing data center deployment simulation that focuses on latency optimization and related expenses. The addition of a greedy provisioning mechanism that gives lower-latency data centers priority is what makes this new. The difficulties cloud providers have in reducing communication latency and allocating resources as efficiently as possible are mirrored in this simulation. The use case entails distributing users between data centers in an effective manner while taking resource and connectivity limitations into account. Reduced communication latency, optimal resource use, and possible cost savings are among the advantages. As a useful tool for studies on latency-aware cloud design, the simulation's capacity to dynamically assign resources based on latency concerns highlights its use in analyzing cloud computing situations where latency is a crucial element. The three scenarios that the research recommends help us understand more about the various elements of cloud computing. The unique aspects of each are found in their particular areas of concentration, which include latency-aware provisioning, cost-effective load balancing, and geographic distance optimization. The approaches in research talks about optimizing user experience, managing diverse workloads, and minimizing latencies. The benefits include improved efficiency, cost-effectiveness, and adaptability in cloud infrastructure management. The capabilities demonstrated in these simulations make them valuable tools for researchers exploring various facets of cloud computing scenarios.

7 Conclusion and Future Work

The research findings offer significant clarity on the merits of different server deployment strategies. These techniques show higher efficiency compared with traditional methods. The technique is validated by key performance measures such as the organization's total expenses, median delay, and overall income. This approach improves on conventional approaches by reducing the distance between chosen servers, which improves the assessment of latency variation. A careful comparison of total income to total cost provides important information about the financial effects of various server selection techniques. In conclusion, compared to other techniques, the recommended server selection strategy is both more effective and financially feasible. The computational paradigm that has been described offers a novel approach to load balancing that takes into account restrictions on system resources, such as RAM, memory, and CPUs, in addition to production time constraints. Allocating resources efficiently is ensured by ranking clients according to their needs for processing operations. To improve the system's allocation choices' transparency, temporary expendable messages are produced that illustrate efficient transfers and point out possible instances of resource shortage. Since a program can simulate how a load is spread across a system, more complicated load-balancing techniques and optimizations can be implemented on a simple and user-friendly platform. The model described above serves as the foundation for the development of elaborate load-balancing algorithms that take into account elements such as computational resources, user-specific activities, and time constraints.

7.1 Future Works:

The future research can be carried out to make the suggested approaches effective by considering aspects like the amount of energy they use and the way they affect the environment. This research might also be able to find out further on how load balancing works in cloud computer systems. It is possible that machine learning methods could make load balancing in the systems simpler by being utilized for predicting and making decisions. Load-balancing methods that are more adaptable and responsive could result from this approach. Additional research needs to be done in these domains. For instance, load balancing techniques should concentrate on retaining dynamic and adaptable parts, as well as sustainability optimization and developing simulation models that can handle an increased number of computing environments. These changes will help the continued growth of cloud computer methods that are both efficient and good for the environment.

References

Armbrust, M., F. A. G. R. J. A. D. K. R. H. K. A. L. G. P. D. A. R. A. . S. I. (2010). A view of cloud computing. communications of the acm, *53(4), 50–58* .

Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R. H., Konwinski, A., Lee, G. R., Patterson, D. A., Rabkin, A., Stoica, I. and Zaharia, M. (2009). Above the clouds: A berkeley view of cloud computing, *Technical Report UCB/EECS-2009-28*, EECS Department, University of California, Berkeley.

Beloglazov, A. and Buyya, R. (2010). Energy efficient resource management in virtualized

cloud data centers, *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing* pp. 826–831.

Buyya, R., Zhang, H., Broberg, J. and Venugopal, S. (2009). Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility, *Future Generation Computer Systems* **25**(6): 599–616.
URL: *https://doi.org/10.1016/j.future.2008.12.001*

Cervone, H. F., Kumar, A. and Nayak, A. (2014). Load balancing in the cloud: A comparative study, *Journal of Cloud Computing: Advances, Systems and Applications* **3**(1): 20.
URL: *https://www.hindawi.com/journals/jccis/2014/345104/*

Chen, Y., Z. X. . Y. J. (2012). Power-aware management for heterogeneous multi-core systems. journal of parallel and distributed computing, 72(2), 183–196.

Dastjerdi, A. V. and Buyya, R. (2016). Fog computing: Helping the internet of things realize its potential, *Computer* **49**: 112–116.
URL: *https://doi.org/10.1109/MC.2016.245*

Foster, I., Zhao, Y., Raicu, I. and Lu, S. (2008). Cloud computing and grid computing 360-degree compared, *Proceedings of the 2008 Grid Computing Environments Workshop (GCE'08)*.

Gmach, D., Rolia, J. and Cherkasova, L. (2013). On the use of workload forecasting for dynamic provisioning of multi-tier applications, *Performance Evaluation* **70**(10): 794–810.

Kliazovich, D., B. P. . K. S. U. (2012). 'greencloud: A packet-level simulator of energy-aware cloud computing data centers', *The Journal of Supercomputing* **vol. 62, no. 3, pp. 1263–1283.**

Kusic, D., Kephart, J. O. and Hanson, J. E. (2007). Power and performance management of virtualized computing environments via lookahead control, *Cluster Computing* **10**(3): 297–312.

Ma, W., Zang, W. and Zhang, Y. (2013). Dynamic internet resource allocation via simulated clusters, *Proceedings of the 2013 42nd International Conference on Parallel Processing*, pp. 437–444.
URL: *https://doi.org/10.1109/ICPP.2013.58*

Marinos, A. and Briscoe, G. (2009). Community cloud computing, *CoRR* **abs/0911.5158**.

Verma, A., Ahuja, P. and Neogi, A. (2010). Power-aware dynamic placement of hpc applications, *Proceedings of the 2010 39th International Conference on Parallel Processing*, pp. 400–405.

Wang, L., von Laszewski, G., Younge, A., He, X., Kunze, M., Tao, J. and Fu, C. (2010). Cloud computing: A perspective study, *New Generation Computing* **28**(2): 137–146.

Zhang, Q., C. L. . B. R. (2010). Cloud computing: State-of-the-art and research challenges', *Journal of Internet Services and Applications* **vol. 1, no. 1, pp. 7–18**.

Zhang, Q., Zhu, Q. and Zhani, M.-F. (2010). Virtual machine migration in federated clouds for network energy saving, *Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing* pp. 826–831.