DOI: http://dx.doi.org/10.26483/ijarcs.v14i3.7006

ISSN No. 0976-5697

Volume 14, No. 3, May-June 2023

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

MECHANISMS AND TOOLS USED FOR RESOURCE ALLOCATION IN THE CLOUD

Karmanbir Singh
Lecturer,Department of Computer Science,
MIMIT,Malout,Punjab,India

Jasmine Kaur
Ph.D. Scholar,Department of Computer Science,
Thapar Institute of Engineering & Technology,
Patiala,Punjab,India

Abstract: Pay-as-you-go access to computer resources is a major selling point of the cloud computing model. Cloud tenants demand complete networking of their dedicated resources to simply implement network functions and services, in addition to the conventional computer resources. The flexibility and convenience of on-demand resource provisioning make cloud computing a compelling computing platform. The key to meeting fluctuating needs and maximizing return on investment from Cloud-supporting infrastructure is dynamic resource allocation and reallocation. For traditional IaaS, we offer an energy-efficient resource allocation strategy based on bin packing. In this paper, we present an accurate energy-conscious method for initial resource allocation by casting the issue of energy-efficient resource allocation as a bin-packing model. The available VMs (virtual machines) employ a modified version of the max-min scheduling technique, which saves money and resources. The results of this study give a framework for comparing and contrasting the many different resource distribution approaches that have been proposed by other researchers. The importance of efficient data centers for the cloud is growing. Power consumption has been a major problem due to its expanding size and widespread usage. The overarching purpose of this effort is to create models and algorithms for resource allocation that are both energy-efficient and take into account a variety of relevant factors.

Keywords: resource allocation, Cloud, Energy, load balancing, VM

I. INTRODUCTION

The openness of cloud computing's underlying infrastructure location makes it an attractive option for developing applications and delivering content. Customers of the Cloud's services often get access to a fraction of the infrastructure's total computing capacity over a scalable network. Distributed elastically in response to consumer demand, the supplier makes these computer resources accessible. Cloud computing is a new approach to running software that differs significantly from the status quo, in which software is often run across infrastructures sized for worst-case and peak-use situations. Allocation and reallocation in Cloud Computing need to be flexible so that they can scale to meet the unpredictable demands placed on the underlying infrastructure. Moreover, making sure all applications' needs are addressed appropriately is another crucial aspect of the resource allocation processes in Cloud Computing. If performance deterioration is kept to a minimum within a certain range, then the resource allocation is considered resilient against perturbations in the stated system parameters.

Allocating resources is an issue that has been discussed in several subfields of computer science, encompassing data center management, grid computing, and operating systems. A Resource Allocation System (RAS) in Cloud Computing is any technique that attempts to guarantee that the requirements of the applications are satisfied by the infrastructure of the service provider. This assurance to the developer should be taken into consideration by the algorithms used to better assign physical and/or virtual resources to the apps to reduce the operating cost of the cloud environment.

Cloud computing is a relatively new technology that uses a service-oriented architecture to provide computer resources (including hardware, operating systems, and applications) via the Internet. Allocating resources in the cloud means assigning work to be done on various machines that make up the cloud's architecture. This cutting-edge technology was developed with the intent of providing customers with pay-as-they-go service. With the ultimate objective of encouraging the strategic development and usage of cloud computing in mind, it is important to clearly describe the activities and interactions involved in this emerging technology[1]. In terms of infrastructure, Among other things, it makes use of server virtualization software and other infrastructure components. Allocating resources in the cloud entails scheduling and providing them while taking into account the available infrastructure, Service Level Agreements, Money, and Power. For instance, cloud service providers use a pay-as-you-go strategy for resource management, all while guaranteeing high Quality of Service (QoS) and customer happiness. In a similar vein, the allocation of resources must be done such that all applications get the resources they need without surpassing the capabilities of the cloud. Similarly, resource allocation allows service providers to provide resources for each specific module, solving the problem of applications hungry for them [2].

The Problem Statement

In cloud computing, everything is saved to an online repository. Each data file is broken down into sections. The overall task execution time (makespan) is long because the workload is unevenly distributed. By using virtualization technology, which generates several virtual computers on a single physical server, this energy inefficiency of the data center may be avoided. Live virtual machine migration may also be used to avoid this by moving VMs to fewer servers based on how they're currently being used, with the idle servers put into power-saving hibernation or sleep. In addition to improving existing infrastructure managers and schedulers like OpenNebula and OpenStack, the suggested methods may also be utilized as an energy usage-cognizant VM scheduler. Tools for estimating energy consumption, such as a joulemeter, may provide signs of power use.

Research Plan

Load balancing is the basis for a model for allocating cloud data resources. Load balancing and decreasing make-span in the cloud are discussed, along with how cloud load balancing for automatic dynamic defragmentation and Modified Max-Min Scheduling load balancing algorithms work. Multi-cloud storage load balancing based on a modified defragmentation model improves dependability, availability, and cloud data storage decision-making to the delight of customers. Virtual machines now in use employ a refined version of the max-min scheduling method, which helps cut down on expenses and power consumption. For traditional IaaS clouds, we suggest a bin-packing-based strategy for optimal resource allocation. We represent the challenge of allocating resources in the most effective use of energy as a bin-packing problem. This concept is Virtual Machine (VM) based and allows for dynamic allocation of resources.

Scope of the Project

This study's focus is on how the cloud computing environment satisfies requirements by allocating resources to apps by their resource needs. Service provisioning, social networking, website hosting, and so on are only some of the many examples of such applications. Therefore, making full use of all available resources is crucial for managing the cloud computing life cycle efficiently. Load balancing is an essential tactic in the cloud life cycle for efficient and effective application deployment. Cloud service providers provide services by the service level agreement (SLA) they negotiate with their customers; thus, the cloud data center employs extremely high-performance servers and other infrastructures to guarantee the SLA is met and the service is always available.

II. LITERATURE REVIEW

G. Bharanidharan and S. Jayalakshmi (2021) [3]System clusters are notoriously complex, and the sheer volume of data they process makes them much more so, the CC system is constantly confronted with new issues. One of the main arguments in favor of switching to a CC system is that it allows for more flexible resource acquisition. In the cloud, elasticity is most useful when it's used to expand or contract virtual resources on the fly in response to changing demand. When it comes to effectively managing cloud resources by user needs and cloud service provider capacity, elastic resource allocation plays a crucial role. This article discusses the research into optimizing load distribution and resource usage using elasticity to achieve optimal resource allocation in the cloud. While there are many articles about cloud computing and its advantages, there is a paucity of surveys that go into depth in examining the cloud's flexibility. This research introduces a new survey design, one that is based on the concept of elastic adaptation in CC, to fill a previously unfilled need. Several mechanisms of elasticity are discussed in this work, from their definition and measurement through their assessment and the current state of elastic solutions, as well as some of the challenges that have been encountered. Finally, several unanswered questions and fresh approaches have been provided in this study. This is the first research to our knowledge to use a systematic review approach to detect problems with elasticity model solutions.

P. Banerjee and S. Roy (2021) [4]The proliferation of the cloud computing environment has been a benefit to the fields of communication and the Internet as a whole to keep up with the ever-increasing need for computing power in today's society. One of the most in-demand infrastructures in the virtual world is cloud computing due to its vast pool of resources, on-demand services, and pay-as-you-go access. Efficient resource management is difficult because of the large volume of requests for both user access and tasks. As a result, increasing productivity and revenue may be achieved by better scheduling of jobs to achieve optimal use of available resources. In this study, we focused on heuristic and hybrid methods for allocating jobs. The two most useful criteria for work schedule—makespan and time flow—have been used to conduct a side-by-side comparison of several heuristic scheduling algorithms and hybrid scheduling algorithms. This paper provides a comprehensive analysis of current job allocation algorithms, highlighting their strengths and weaknesses.

O. Runsewe and N. Samaan (2019) [5]If you're looking to cut down on service costs and make better use of your cloud's resources, containerization may be a good option to consider. When hosting a heterogeneous cluster of hosts, it might be difficult to decide how to divide up container resources between competing streaming applications with wildly different requirements for quality of service. This research focuses on workload distribution for optimal resource allocation to meet the real-time requirements of competing containerized big data streaming applications. We overcome this problem by treating the interaction between the various containerized streaming applications as a competitive n-player environment. Our research leads us to the Nash Equilibrium, the best possible situation in which no player may increase their performance without hindering others. Unlike previous solutions, which may unjustly treat certain applications, our method aims to meet the request of every containerized streaming application equally.

M. S. Quesada, et al. (2021) [6]The rapid growth of automobile ownership in cities has created significant societal issues and new obstacles. Vehicular ad hoc networks (VANETs) may be used to fine-tune the network as more and more vehicles and other mobile devices become linked to it. VANETs allow cars and infrastructures to communicate with one another to share data and services. Another idea called Vehicular Ad Hoc Networks (VANETs) adds Cloud principles to this situation and is named Vehicular Cloud

Computing (VCC). In this paper, we propose a system for vehicle cloud resource allocation that we call NAUTILUS and that takes its inspiration from the biology of bats. To define pseudo-optimal allocation decisions in a Vehicular Cloud, the method makes use of the metaheuristic search technique. To further aid the suggested mechanism's allocation, we investigate a fog-based paradigm. We allow the cars' storage, memory, runtime, and computing power. The NAUTILUS was compared against the Greedy and AHP algorithms, both of which use conventional search strategies. We count the number of blocked, attended, and rejected services and then compare those totals. The simulation findings suggest that the NAUTILUS strategy is more effective than the Greedy and AHP methods in terms of fewer blocked connections, higher attendance rates, and fewer denied service requests.

III. PROJECT JUSTIFICATION/DESCRIPTION AND GOALS OF THE PROJECT

The suggested dynamic defragmentation paradigm for cloud storage uses energy-efficient load balancing to improve cloud storage decision-making for all users. In small-scale distributed systems, Min-Min and Max-Min algorithms are often used. To get around this problem, we offer a modified version of the Max-Min method. In the next part, we will examine the dynamic defragmentation paradigm that is used automatically for load balancing.To accomplish this goal of lowering energy usage and VM migration costs, In this paper, we discuss both exact and approximate methods for performing initial resource allocation and subsequent dynamic reallocation.

Cloud Simulation Tool

The grid simulation toolbox could not separate the distinct tiers of cloud services (SaaS, PaaS, and IaaS). Thus, several different cloud simulation technologies have been explored as potential solutions to these problems. The CloudSim utility is used for simulation, and it may be used on the Java platform. The following are some of the most crucial components of CloudSim: Cloudlet, Host, Virtual Machine Allocation Policy, and Datacenter.

Automatic dynamic defragmentation using cloud-based load balancing

A Virtual Data Center (VDC) is a kind of cloud capacity that is used in the suggested paradigm. The data centers' storage servers are set up in clusters, each of which is made up of many individual servers. Instead of permanently keeping data in the accessible cloud data server when a user attempts to share or upload a large volume of data, the model briefly transfers data into the fragmented servers. To calculate how much storage space is available on the cloud server, an adaptive method is used. When data is spread out among numerous cloud servers in this manner, all of the available cloud space may be used, our proposed approach, as shown in Figure 1, provides better performance metrics and their calculation.

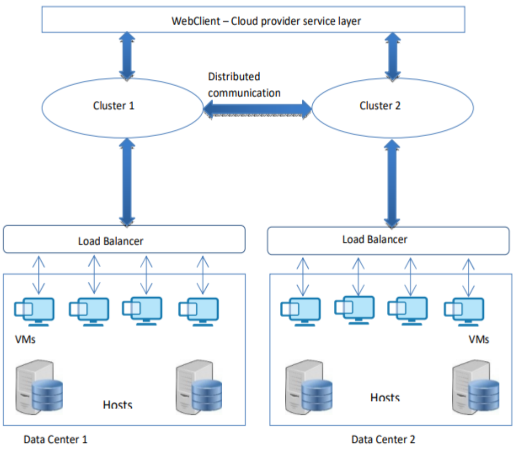
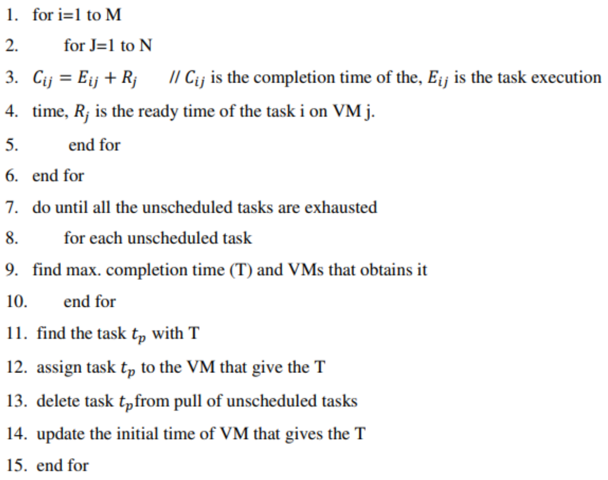


Figure1: Proposed model for Automated dynamic defragmentation using cloud load balancing

Alternative load-balancing technique based on Max-Min Scheduling

Max-Min is a resource allocation and scheduling method utilized in many areas of cloud and grid computing to maximize profit and minimize makespan. To do this, we assign the resource with the fastest execution time to the job with the longest estimated completion time in the job list. The algorithm's description elucidates the likelihood of assigning priority to each work in a timely fashion that conserves completion time and by the first in, first out principle before determining the function of the other task that requires the least amount of time to complete.

Max-Min Algorithm's Pseudocode:



```
for i=1 to M
    for J=1 to N
$C_{ij} = E_{ij} + R_j$     // $C_{ij}$ is the completion time of the, $E_{ij}$ is the task execution
time, $R_j$ is the ready time of the task i on VM j.
    end for
end for
do until all the unscheduled tasks are exhausted
    for each unscheduled task
find max. completion time (T) and VMs that obtains it
    end for
find the task $t_p$ with T
assign task $t_p$ to the VM that give the T
delete task $t_p$from pull of unscheduled tasks
update the initial time of VM that gives the T
end for
```

The System Model

The model takes into account cloud service providers distributing instances of physical resources to house virtual machines (VMs) belonging to their customers and tenants. The hardware components are analogized to servers. Providers of the underlying infrastructure will presumably host apps that have been bundled into virtual machines. Cloud service providers save money and resources by migrating to virtual machine systems (VMS) that allow for greater server consolidation and use of sleep states. The components of the system model shown in Figure 2 are a cloud manager, an energy consumption estimator, and the proposed energy-efficient allocation and migration methods. To be ready for analytical modeling of the cloud's resource allocation problem with an emphasis on energy efficiency [7], we provide short descriptions of each module.

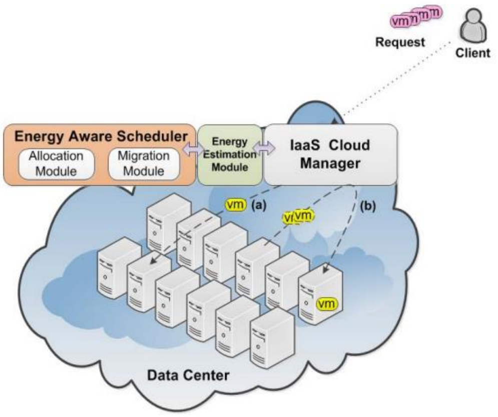


Figure 2: The system model

- Managers of cloud infrastructure as a service (IaaS) like OpenStack, OpenNebula, and Eucalyptus process requests from customers, schedule virtual machines and retrieve and store images in data centers.

- The Energy Estimation Module Sits Between Cloud Infrastructure Management and Energy-Aware Scheduling.

- Our energy consumption optimization approach focuses on the energy-aware virtual machine (VM) scheduler that is responsible for the energy-aware placement of VMs in the data center.

IV. RESULTS

An Energy-Efficient Load Balancing Model for Cloud Data Resources

The model is written in Java, a widely used programming language. Each component has been optimized for use in running simulations of the cloud defragmentation model. To implement the suggested concept, a Servlet-based application is built. Tomcat is used for all of the web development and server backend, and it makes use of open-source development environments such as Eclipse JEE. To test our hypotheses, we use Eclipse JEE, an open-source IDE for creating enterprise-grade, web-based applications, to create a prototype.

Evaluation of Results

Processing speed, throughput, instruction volume, and data volume are only a few of the assessment criteria used to gauge the efficacy of the suggested energy-efficient load-balancing approach in the cloud storage model for automatic defragmentation. Consider a schedule manager faced with the difficulty of allocating two resources (R1 and R2) to four tasks (T1, T2, T3, and T4). Table 2 shows the number of rule predictions and information from T1 to T4 at various rates of speed and bandwidth (Table 1).

Table1:Comparing the Effectiveness of the Max-Min with a Modified Max-Min Method

Problemsample	Resources	ProcessingSpeed(MIPS)		Throughput(MBPS)	
		Max-Min	Modified Max-Min	Max-Min	ModifiedMax-Min
P1	R1	29	50	87	100
	R2	73	100	300	500
P2	R1	128	150	279	300
	R2	283	300	130	150
P3	R1	272	300	282	300
	R2	17	30	130	150

The results of the original max-min algorithm and the new Modified max-min method are shown in Table 1. Here, R1 and R2 are the available resources, while p1, p2, and p3 are the corresponding problem sets. Figure 3 compares the processing times of the standard Max-Min algorithm to the new, improved algorithm.

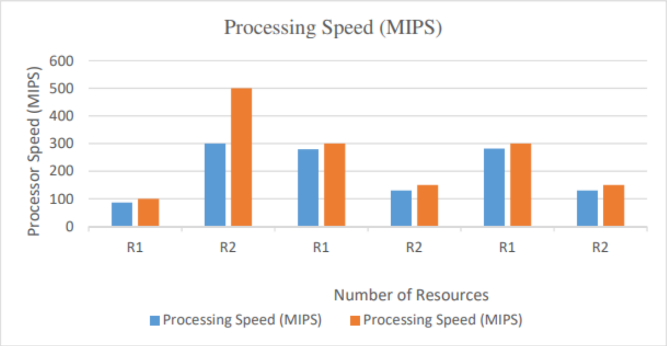


Figure 3: Processing Speed (MIPS)

As can be seen in Figure 3, the processing speed of the modified Max-Min methodology is faster than that of the traditional Max-Min method. In terms of computational efficiency, the modified Max-Min approach that was presented yields superior results. Since it transmits a lot of data quickly, its speed is quite high.

Table 2:Instructional Effectiveness and the Quantity of Data

Problem sample	Task	InstructionVol. (MI)		DataVol. (MB)	
		Max-Min	Modified Max-Min	Max-Min	Modified Max-Min
P1	T1	112	128	39	44
	T2	52	69	53	62
	T3	201	218	83	94
	T4	11	21	43	59
P2	T1	212	256	73	88
	T2	21	35	25	31
	T3	298	327	89	96
	T4	201	210	432	590
P3	T1	11	20	63	88
	T2	321	350	27	31
	T3	192	207	83	100
	T4	17	21	37	50

Table 2 shows how much information and directions will be needed to complete the project. P1, P2, and P3 denote three examples of problem sets. Each set of problems has four parts, labeled T1 through T4. The modified Max-Min approach outperforms the original Max-Min technique. There is a rapid exchange of data between individual nodes. The big quantity of instruction and data may be sent in a modified Max-Min amount of time [8].

Energy-efficient allocation of static resources

Algorithm for Exact Allocation:

By adding valid requirements in the form of constraints or inequalities, the proposed precise VM allocation process expands on the original Bin-Packing method. The idea is to keep items together in storage units. We also provide the data center's server count (m) alongside n. It is assumed that each server has a similar maximum power consumption: $P_{j,Max}$, {j = 1, 2, ..., m}. Each running virtual machine (VM) on the server j's current power usage is characterized at runtime by $P_{j,current}$. To deploy all requirements (or VMs) on as few servers as possible, the goal function may be stated as:

$$\min Z = \sum_{j=1}^{m} e_j \quad (1)$$

When hosting virtual machines, a server's total power consumption cannot exceed a predetermined limit known as $P_{j,Max}$.

$$\sum_{i=1}^{n} p_i x_{ij} \leq P_{j,Max} e_j - P_{j,Current}, \forall j = 1, \ldots, m \qquad (2)$$

Each virtual machine (VM) request shall be fulfilled by one and only one server and the cloud provider will do so within the bounds of any applicable service level agreement (SLA) or limit.

$$\sum_{j=1}^{m} x_{ij} = 1, \forall i = 1, \ldots, n \qquad (3)$$

The precise and expanded Bin-Packing VM allocation model may be summed up by the following set of equations, which include the goal function and all the restrictions and conditions.

$$\min Z = \sum_{j=1}^{m} e_j \qquad (4)$$

The linear solver CPLEX and a Java implementation of our suggested techniques are used for evaluation. The evaluations and comparisons of performance are carried out with the use of a custom-built simulator. The numerical evaluation's goal is to determine the estimated proportion of energy savings or power consumption savings that result from integrating our suggested precise migration algorithm into both the exact allocation algorithm and the consolidation procedure. The maximum power draw for each server is limited to 200 watts (a typical server's peak power is 250 watts). We used a suggested power estimate approach to calculate power consumption for each virtual machine. High, medium, and low power consumption workloads on the SP ECcpu2006 were tested. In terms of wattage, they need around 13, 11, and 10 respectively, correspondingly [9].

Results of a comparison between our accurate extended Bin-Packing allocation technique and the modified Best-Fit heuristic are shown in Figure 4. The virtual machines (VMs) and server counts in the [1, 200] range are used in the simulations. The VMs all have the same lifespan of [30s, 180s]. In other words, the duration of a VM task is at least 30 seconds and will be shorter than 180 seconds. Figure 4 shows that in the simulated 1000s-time period, the precise allocation technique performs better than the Best-Fit heuristic.

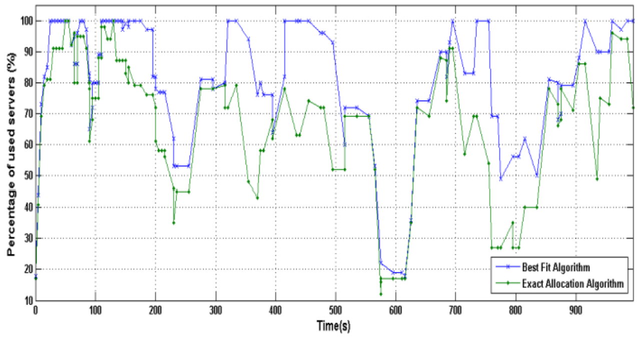


Figure 4: A comparison of the heuristic and accurate allocation methods

The accurate and expanded Bin-Packing allocation technique is further analyzed in Figure 5, which compares the algorithm's performance with and without consolidation. Combining the exact algorithm with the migration algorithm may considerably increase the former's efficiency and the latter's ability to cut down on energy usage. Ten percent to twenty percent more servers might be idled on average. The typical exact algorithm line utilizes around 80% of available servers, whereas the typical exact algorithm with migration uses closer to 60%.

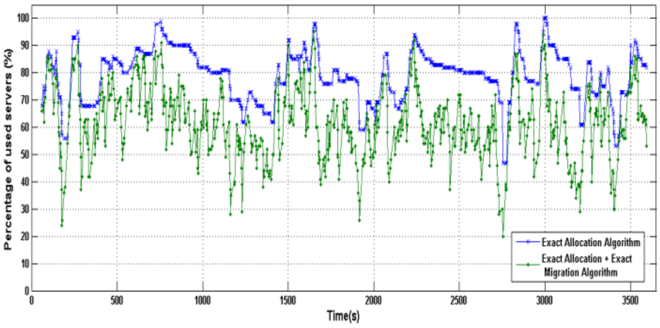


Figure 5: The precise allocation algorithm's performance with and without migration

V. CONCLUSION

A new processing paradigm, cloud computing offers users "everything as a service." The emergence of cloud computing has made it possible to efficiently distribute resources like memory, storage, and processor for the execution of tasks and services for a geographically dispersed supply chain. The workloads may be balanced and the makespan can be shortened with the help of a load-balancing technology that uses less energy. Energy efficiency is greatly aided by the outcomes of effective work scheduling methods. The suggested method of task consolidation for VM scheduling is compared to the state-of-the-art methods, and the resulting data demonstrates the method's superiority. Compared to the best fit heuristic, experimental findings indicate that combining the allocation and migration algorithms yields considerable energy reductions at manageable runtimes [10].

Using the suggested defragmentation approach, cloud service providers may more easily manage, distribute, and store their customers' data across several hosts. Among the previously evaluated techniques, an efficient planned load balancing method has been recommended to save energy and money while the operation is being carried out and the needs are being met. Load balancing adds a buffer between the virtual machines (VMs) and the client, allowing the latter to provide requests more quickly. Scheduling load distribution across available VMs helps save money and resources. We represent the challenge of allocating resources in the most effective use of energy as a bin-packing problem. This concept is a virtual machine (VM) based and allows for dynamic allocation of resources. To begin allocating resources, we suggest an ILP-based algorithm that is both precise and energy-conscious. In addition, a dynamic VM reallocation technique based on an accurate ILP was developed as a means of coping with the issue of resource consolidation on the fly. Using virtual machine migration, it seeks to continuously enhance energy efficiency following service terminations.

Recommendations Future Research

As a fast-changing field, cloud computing offers exciting new opportunities for researchers and developers. Also, in the context of Cloud Computing, resource allocation optimization is a large field of study. Task consolidation models are used in cloud environments to improve CPU usage and reduce power consumption. However, there is still some catching up to do in terms of security when it comes to cloud storage, so future studies should concentrate on addressing these issues while also making the most of virtualization and the cloud's architecture. The whole system load may be predicted with the use of load prediction algorithms. In the future, we want to include prediction algorithms in our solutions to make the resource allocation techniques we provide even more reliable and effective.

REFERENCES

[1] S. Gong, B. Yin, Z. Zheng, and K.-Y. Cai, “Adaptive multivariable control for multiple resource allocation of service-based systems in cloud computing”, IEEE Access, vol. 7, pp. 13817–13831, 2019. DOI: 10.1109/ACCESS.2019.2894188.

[2] Q. Qi and F. Tao, “A smart manufacturing service system based on edge computing, fog computing, and cloud computing”, IEEE Access, vol. 7, pp. 86769–86777, 2019. DOI: 10.1109/ACCESS.2019.2923610.

[3] G. Bharanidharan and S. Jayalakshmi, "Elastic Resource Allocation, Provisioning and Models Classification on Cloud Computing A Literature Review," 2021 7th International Conference on Advanced Computing and Communication Systems (ICACCS), Coimbatore, India, 2021, pp. 1909-1915, doi: 10.1109/ICACCS51430.2021.9442018.

[4] P. Banerjee and S. Roy, "An Investigation of Various Task Allocating Mechanism in Cloud," 2021 5th International Conference on Information Systems and Computer Networks (ISCON), Mathura, India, 2021, pp. 1-6, doi: 10.1109/ISCON52037.2021.9702358.

[5] O. Runsewe and N. Samaan, "CRAM: a Container Resource Allocation Mechanism for Big Data Streaming Applications," 2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID), Larnaca, Cyprus, 2019, pp. 312-320, doi: 10.1109/CCGRID.2019.00045.

[6] M. S. Quessada, D. D. Lieira, R. S. Pereira, R. E. De Grande, and R. I. Meneguette, "A Bat Bio-inspired Mechanism for Resource Allocation in Vehicular Clouds," 2021 17th International Conference on Distributed Computing in Sensor Systems (DCOSS), Pafos, Cyprus, 2021, pp. 197-204, doi: 10.1109/DCOSS52077.2021.00042.

[7] Xiaozhou Zhang, Tsung-Hui Chang, Ya-Feng Liu, Chao Shen, and Gang Zhu, “Max-Min Fairness User Scheduling and Power Allocation in Full-Duplex OFDMA Systems,” IEEE,2019.

[8] Chaima Ghribi, Makhlouf Hadji, and DjamalZeghlache. Energy efficient vm scheduling for cloud data centers: Exact allocation and migration algorithms. In CCGRID, pages 671–678. IEEE Computer Society, 2013. ISBN 978-1-4673-6465-2.

[9] Middya, Asif & Ray, Benay& Roy, Sarbani. (2019). Auction-Based Resource Allocation Mechanism in Federated Cloud Environment: TARA. IEEE Transactions on Services Computing. PP. 1-1. 10.1109/TSC.2019.2952772.

[10] O. Abdul Wahab; J. Bentahar; H. Otrok; A. Mourad, “Towards Trustworthy Multi-Cloud Services Communities: A Trust-based Hedonic Coalitional Game, in IEEE Transactions on Services Computing, vol.PP, no.99, pp.1-1, 2016.