International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7
Article Received: *25 April 2023* ***Revised:*** *12 May 2023* ***Accepted:*** *30 June 2023*

Machine learning based Model for Cloud Load Prediction and Resource Allocation

Nanasaheb Bhausaheb Kadu
Department of Computer Science & Engineering
Dr. A.P.J. Abdul Kalam University
Indore, India
kadukamlesh@gmail.com

Dr. Pramod Jadhav
Department of Computer Science & Engineering
Dr. A.P.J. Abdul Kalam University
Indore, India
ppjadhav21@gmail.com

Abstract — Elasticity and the lack of upfront capital investment offered by cloud computing is appealing to many businesses. There is a lot of discussion on the benefits and costs of the cloud model and on how to move legacy applications onto the cloud platform. Here we study a different problem: how can a cloud service provider best multiplex its virtual resources onto the physical hardware? This is important because much of the touted gains in the cloud model come from such multiplexing. Studies have found that servers in many existing data centers are often severely under-utilized due to over-provisioning for the peak demand. The cloud model is expected to make such practice unnecessary by offering automatic scale up and down in response to load variation. Besides reducing the hardware cost, it also saves on electricity which contributes to a significant portion of the operational expenses in large data centers.

Proper resource allocation for various virtualized resources must be based on these cloud load predictions. The presence of heterogeneous applications, such as content delivery networks, web applications, and MapReduce tasks, complicates this process. Cloud workloads with conflicting resource allocation needs for communication and information processing further exacerbate the difficulty.

Keywords: *Cloud Computing, Load prediction, Resource allocation, Task allocation.*

I. INTRODUCTION

Three different resource services are offered by cloud computing as a commercial computing pattern: Platform as a Service (PaaS), Software as a Service (SaaS), and Infrastructure as a Service (IaaS). Cloud computing still has issues with scheduling tasks and resources even if it offers a variety of services and is focused on different application programs. Regarding the latter issue, user service quality plays a crucial role in determining the reliability of the cloud service, resource usage rate, and operating costs. As a result, the multi-objective task scheduling problem in cloud computing has important theoretical and practical implications. The resources and their loads in the cloud computing environment are subject to several dynamic and unpredictable circumstances. For example, the resource node's load varies with time, and resource requests differ according to the year, quarter, and holiday.

These elements could potentially result in resource waste and decreased service quality. Resources will be squandered if the cloud resource load is too low; on the other hand, the system's service performance would suffer if the cloud resource demand is extremely high. Scholars in the aforementioned domains have conducted study on cloud computing job scheduling in relation to the aforementioned issues. A job scheduling approach for ant colony optimization that is based on load balance, cost, and the shortest task completion time. The task scheduling elements of cloud computing specify the

load balancing function and the cost constraint function of task completion time, along with providing the initial pheromone. Next, it enhances the ant colony optimization algorithm's heuristic function and pheromone update technique. Using the ant colony optimization algorithm, it derives the objective constraint function and, ultimately, the global optimal solution. After that, it compares the ACO method with the Min-Min algorithm and runs a cloudsim simulation. This approach outperforms the other two algorithms in terms of job execution time, cost, or load balance, as the experiment demonstrates.

The actual implementation of a cloud environment has been hampered by several issues. Resource discovery, scheduling,

security, and privacy are a few of them. Among these, load balancing is one of the most important issues. It describes the distribution of the load across several machines. The delivery and distribution of the necessary workload across many computing platforms is referred to as load balancing. Methods for optimizing system output production, resource usage, and virtual machine (VM) performance factors are proposed by load balancing. In order to optimize resource use, the cloud system makes use of many load-balancing techniques. Recent surveys have presented a few of these algorithms. Reducing reaction times and increasing resource utilization are the goals of load balancing, which raises production while cutting expenses. Furthermore, load balancing strives to offer durability and flexibility for applications that expand in scope and require additional resources. It also gives equal division of complex tasks top priority. This study provides an in-depth analysis of the classification schemes used in the literature to describe load balancing strategies before going any further. Two types of load balancing exist in cloud-based systems:

(1) Static algorithms and (2) dynamic algorithms.

- In a static method, Location is ignored by the available base stations in a static approach. All of the connections and their traits are known in advance. This type of method has predetermined execution. It doesn't rely on real-time data from the existing system and is easy to use.
- On the other hand, the dynamic balancing procedure takes into account the machine's current state. Its operation is driven by changes in the node design. Although dynamic algorithms are challenging to build, they more effectively balance load by distributing resources in an efficient manner.

In today's rapidly evolving digital environment, users have access to a vast array of resources on the internet, including technology, services, applications, and other relevant assets. A pioneering approach is introduced to enable users to access a diverse range of resources tailored to their budget, covering data, software, power infrastructure, connectivity, and more. This strategy empowers users to utilize their computer networks according to their individual needs, regardless of their internet connection status. The core principle underlying this approach is the freedom for individuals to utilize available resources in a manner that best suits their requirements. Users can initiate cloud computation by submitting requests, thereby simplifying processing, improving service quality, and enhancing customer satisfaction.

Cloud-based services offer a significant advantage over traditional technologies due to their lower initial costs, as users are only charged for the specific features they use. This reduction in the procurement of processing power enables programmers to focus on innovation rather than hardware acquisition. Accelerating software deployment can be achieved through a process scheduler that optimizes concurrent execution by identifying the most suitable resources for workflow tasks and allocating them to various processing units.

Cloud scheduling plays a crucial role in cloud infrastructure, involving three primary steps: resource determination, implementation of filtering rules based on the current state of the networked cloud service, and resource allocation based on specific objectives. Static scheduling, where temporal aspects such as task duration and start and finish times are predetermined, enables concurrent execution by outlining the methodology for each task in advance.

Scalability emerges as a key quality characteristic in cloud computing systems. Effective management of data and cloud infrastructure with optimal task distribution is essential for achieving higher performance and scalability. Large-scale distributed architectures require adaptive real-time resource management to meet evolving demands. The challenge lies in accommodating the substantial data requirements of data-intensive applications while establishing relevant scalability indicators and evaluation frameworks. Leveraging optimization techniques enables the integration of heterogeneous resources into a system that scales linearly and fulfills diverse optimization goals, thereby enhancing Quality of Service (QoS) measurements in the cloud. The adoption of software-defined resource management and artificial intelligence (AI) approaches facilitates quick and flexible adaptation to changing workflow requirements, ensuring optimal resource utilization.

[1] P. P. Chen. introduces a multi-objective task scheduling optimization for cloud computing based on a fuzzy self-defense algorithm. By formulating an objective function and solving it using the fuzzy self-defence algorithm, the model achieves optimal task scheduling. Comparative analysis demonstrates the superiority of this approach in terms of maximum completion time, deadline violation rate, and virtual machine resource utilization.

[2] Gennaro Costagliola, Vincenzo Deufemia, and Giuseppe Polese proposes a self-supervised point cloud upsampling technique, enabling the generation of dense and uniform point sets from sparse inputs without supervision from dense point clouds. The technique leverages point feature extraction, expansion, and self-projection optimization to significantly improve point cloud upsampling quality.

[3] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Unman, and J. Widom. develop the TAM load-balancing system to optimize performance for both long and short flows in data center environments. TAM effectively tracks the status of parallel lines to select the optimal transmission path for flows or flowlets, resulting in performance improvements of 20% to 60% compared to existing load balancing systems.

[4] Peter J. Clarke, Vagelis Hristidis, Yingbo Wang, Nagarajan Prabakar and Yi Deng.evaluate a rule-based framework for managing cloud-native applications, contributing to SLA compliance by establishing adaptable policies for cloud-native environments. Experimental evaluation demonstrates the effectiveness of the framework in achieving autonomic

management, even in dynamic and frequently changing environments.

The primary focus of this paper is on maximizing centralized resource management through task scheduling while considering scalability in heterogeneous cloud systems. The research evaluates the performance and scalability of heterogeneous cloud infrastructure for resource-intensive tasks. In summary, this study offers insights into cloud load prediction and resource allocation models, providing details on the deployed approach, evaluation results, and conclusions.

II. LITERATURE SURVEY

[5] Xinliang Wei et al. delved into a challenge in edge computing related to optimizing both resource allocation and task dispatching across different timeframes. To address this dual optimization in a dynamic edge environment, the researchers proposed an approach based on deep reinforcement learning and a two-stage optimization technique. The simulation results demonstrated that: (1) both proposed methods surpassed greedy and random algorithms in performance; and (2) conducting resource placement and task dispatching at different time scales not only reduces placement costs but also minimizes the need for extensive task prediction in the future.

[6] K. Fraser, S. Hand, R. Neugebauer, I. Pratt, A. Warfield, and M. Williamson, introduced a ground breaking component for load balancing, named CHEETAH, which ensures PCC (Predictable, Consistent, and Controllable) and supports any feasible LB mechanism. CHEETAH was implemented using programmable ASIC Tofino switches and software switches. The authors consider this work as an initial step towards fully leveraging the potential of load balancing techniques. They leave open the question of whether new load balancing techniques tailored specifically for Layer 4 LBs can be developed and integrated with existing middle boxes in future research.

[7] E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J., discussed cooperative autonomous driving, aiming to maximize efficiency and safety through wireless exchange of sensor data among autonomous vehicles. The authors also explored a dynamic map system, serving as a platform for managing shared sensor data and running applications. However, scalability issues arise when the number of vehicles transmitting and receiving sensor data increases. To address this, the authors proposed dispersing numerous edge servers worldwide and managing their data in the cloud. Challenges remain in effectively managing applications like merging arbitration and intersection collision warning due to fluctuating radio wave conditions. The authors introduced a lane section ID system to assign each road segment to an edge server and built a dynamic map system linking cars and edge servers. Multicast communication between the edge server and the vehicle, along with anycast for vehicle interaction, was proposed to facilitate effective communication.

[8] Ramezani, Fahimeh, Jie Lu, and Farookh Khadeer Hussain., developed an optimal task offloading scheme for multi-cloud and multi-edge networks considering network topology and bandwidth limitations. Their cooperative multi-agent deep reinforcement learning (Coop-MADRL) technique forms the foundation of the task offloading approach. By utilizing deep reinforcement learning to learn the relationship between network-demand patterns and suitable task offloading, this strategy achieves efficient task offloading quickly. Additionally, the authors proposed a cooperative multi-agent mechanism to enhance task offloading effectiveness. Experimental results demonstrated that the suggested approach can reduce network usage and task latency while minimizing constraint violations in various network topologies. The authors intend to further evaluate the effectiveness of their method in more complex scenarios and enhance its scalability and interpretability.

[9] Sarker, Tusher Kumer, and Maolin. Tang., explored how evolutionary optimization algorithms can handle multitasking efficiently. The main challenge lies in translating diverse task design spaces into a single representation for optimization. To address this, the authors presented an MFEA framework leveraging a 3-D point cloud auto encoder to learn a domain-independent representation. This approach enables transferring designs from the unified space to the Cartesian space, fostering common geometric characteristics among designs. The authors demonstrated the effectiveness of the MFEA framework in optimizing shape designs for vehicle aerodynamics. They also highlighted the potential for enhancing manufacturing and maintenance efficiency by transferring latent elements representing automobile forms' underbody.

[10] Vakilinia Shahin, introduced Structured Allocation-based Consistent Hashing (SACH), a hashing algorithm incorporating consistency features to meet load balancing, consistency, memory utilization, fault-tolerance, and lookup time requirements in cloud architecture. SACH is designed for cloud infrastructure systems with low concurrent backend failure rates. The authors demonstrated through experimental results that SACH outperforms existing techniques in load balancing and lookup rate, particularly when concurrent backend failure rates are low. SACH is expected to benefit both academia and industry.

These summaries offer insights into cutting-edge research across various domains, including edge computing, load balancing, autonomous driving, and optimization algorithms. Each study addresses specific challenges and proposes innovative solutions to advance the respective fields.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7
Article Received: *25 April 2023* ***Revised:*** *12 May 2023* ***Accepted:*** *30 June 2023*

III. PROPOSED MODEL METHODOLOGY

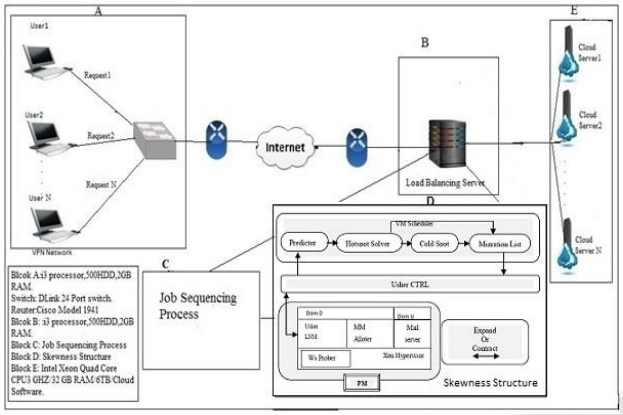

Figure 1: Proposed model for load prediction and Resource allocation

Figure 1 above illustrates the proposed method for resource allocation and load prediction at the cloud.

The proposed methodology is devised to simulate cloud load forecasting and resource allocation within a virtual environment. A novel resource allocation technique is formulated utilizing both the Hungarian neural network and Genetic algorithm, leveraging the generated load dataset. Several sequential steps are undertaken to execute the processes of load prediction and resource allocation.

Step 1: Data Generation: The proposed model initiates dataset generation for load prediction through a random number generation process based on threshold maintenance selection. The generator constructs datasets encompassing fields such as num_of_shards, num_of_samples, new_period, Shape, and Size, which are eventually stored in a workbook file.

Step 2: Pre-processing: Upon dataset generation, pre-processing procedures are employed. The dataset is read from the specified path into a double-dimensional list, utilizing Python's Pandas module. Early attributes' parameters are estimated to characterize the dataset features within their respective ranges. Moreover, the dataset's entropy across various data types is asserted, and oversampling of labelled classes is conducted to ensure balanced distribution.

Step 3: Data Imputation: Oversampled data is utilized to estimate heat maps for each characteristic, facilitating the identification and imputation of missing data. The Iterative Imputer () method, employing multiple imputation using chained equations (MICE), restores missing values under specific assumptions about data missing-ness mechanisms. Subsequently, the interquartile range (IQR) is estimated to fill in missing values.

Step 4: Pearson Correlation: A comprehensive dataset list is subjected to Pearson Correlation analysis to identify characteristics with the lowest correlation values. A correlation matrix is generated to aid in selecting attributes with optimal correlation, enhancing the dataset's predictive capacity.

Step 5: Attribute Selection: The MinMaxScaler function is applied to the imputed and pre-processed data to alter features within a specified range. This procedure generates two lists for cloud load prediction attributes, selecting critical characteristics based on their highest scores.

Step 6: Data Segregation: Data segregation involves dividing attributes into features (X) and labels (Y) using the train_test_split () function. Subsequently, the data is split into training and test sets, enabling model fitting and validation.

Step 7: Long Short-Term Memory (LSTM): The LSTM neural network is employed for load prediction, necessitating scalar normalization and parameters such as units, features, and return sequences. Additionally, a dense layer with "relu" activation function is added to efficiently learn new information.

The proposed methodology, detailed through sequential steps, aims to optimize cloud load forecasting and resource allocation within virtual environments, thereby enhancing system efficiency and performance.

```
Model: "sequential_1"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 10)                1280

dense_2 (Dense)              (None, 1)                 11

=================================================================
```

Table 1: Long Short-Term Memory MAPE %

Step 8: Decision Tree for Resource Allocation

The result of the Hungarian neural network from the previous stage is a sequence list for the upload process, obtained through innovative fusion with the Hungarian job allocation system in the H-net neural network. This sequence list aids in achieving better resource allocation results, facilitating real-time customization of scheduling techniques based on cloud resource characteristics. The decision-tree technique utilizes adaptive resource allocation schemes to maintain cloud properties and attain superior outcomes.

IV RESULTS AND DISCUSSIONS

The proposed model for optimizing cloud load prediction and resource allocation using deep learning is developed on a Windows-based computer system, implemented with Python programming language. The development environment utilized Spyder IDE from the Anaconda distribution. The testing system features a 500GB SSD, 16GB RAM, and an Intel Core i5 processor.

To ensure effective cloud load prediction and resource allocation, comprehensive performance evaluation is crucial. In this approach, a dataset is generated using an adaptive generator scheme for cloud load prediction and fed into a Transformer model to predict load categories. The Transformer model's output is then used by a Genetic Algorithm to devise a resource allocation strategy. Both the Transformer's predictions and the Genetic Algorithm's resource allocations undergo rigorous testing to validate their effectiveness, employing the evaluation methods outlined below.

Estimating Convergence Trend of Loss Functions for Resource Allocation with Deep Learning Models.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7
Article Received: *25 April 2023* ***Revised:*** *12 May 2023* ***Accepted:*** *30 June 2023*

Algorithm complexity is a critical aspect of the resource allocation mechanism in device-to-device communication. To demonstrate the reliability and efficiency of the proposed Genetic Algorithm (GA) paradigm, a comparative analysis of model loss functions is conducted. This study compares the GA model against traditional models such as the deep Monte Carlo Tree Search (MCTS), single-chain deep MCTS, and double-chain deep MCTS.

From the perspective of convergence value and speed, the GA model is tested over 2500 iterations. Figures 6 and 7 present a comparative graph showing the convergence trends of the loss functions. The trend indicates that the GA model converges efficiently and achieves a superior final convergence value compared to the single-chain deep MCTS, double-chain deep MCTS, and deep MCTS, thereby validating the effectiveness of the GA model proposed in this study.

The performance evaluation of the cloud load forecasting system was conducted using various metrics, including Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and Mean Absolute Percentage Error (MAPE). These metrics allow for the quantification of the error generated by the forecasting system. The accuracy of the proposed approach was demonstrated by comparing it with a previously proposed method, which showed its superiority.

The RMSE metric simplifies the evaluation of error between two continuously correlated metrics, measuring the accuracy and inaccuracy of load predictions.Additionally, an ensemble approach combining LSTM and HNet outperformed a previously proposed model that used a hybrid deep learning and beluga whale optimization technique (LFS-HDLBWO). The LSTM-HNet ensemble model achieved better results in terms of MAPE, as it leveraged resource allocation information during the load forecasting process, thereby reducing the chance of error. The results of the comparison are summarized in Table 2 and Figure 9.

	MAPE %	
Data Size (samples)	LFS-HDLBWO	LSTM-HNet
300	0.228	0.199
480	0.266	0.191
600	0.254	0.188
720	0.059	0.031

Table 2**:** Performance results of models for MAPE %

V. CONCLUSION AND FUTURE SCOPE

In this research, the proposed model utilizes a synthetic data generation engine to create cloud load data based on parameters such as num_of_shards, num_of_samples, new_period, shape, and size attributes. During data generation, the model applies data segmentation and pre-processing. This prepared dataset is then used by a Random Forest model to predict cloud workloads. The prediction yields a sequence of parameters, which is subsequently used with a Genetic Algorithm to determine the optimal sequence for resource allocation. This sequence aids in distributing resources across multiple virtual machines.

Using the Random Forest-Genetic Algorithm (RF-GA) model to forecast load and allocate resources simultaneously, the load prediction model iteratively triggers resource allocation sequences. The information from the allocated resources improves the learning process of load forecasting, significantly reducing the error rate. The results demonstrate that the proposed model outperforms the previous model in [21] in terms of the convergence of the loss function due to the effective deployment of the Genetic Algorithm. The RF-GA ensemble strategy surpasses the model described in [22] by approximately 26.79% in MAE and around 37.76% in RMSE. Additionally, the proposed model outperforms the model in [23] by approximately 24.75% in MAPE. This demonstrates the efficiency of the RF-GA model in handling the load forecasting and resource allocation paradigm in cloud environments.

Future research will aim to incorporate more workload and resource variables to enhance the robustness of the Random Forest model, leading to improved solutions. Future studies will also explore the development of a proactive task scheduling algorithm based on Evolution Strategies, balancing load balancing, job assignment time, task completion time, and cost by combining the Genetic Algorithm with other techniques such as Multi-party computation in cloud-based virtual servers.

REFERENCES

[1] P. P. Chen, “The entity-relationship model: Toward a unified view of data”, ACM Trans. Database Syst. 1, 1, 9–36, 1976.

[2] Gennaro Costagliola, Vincenzo Deufemia, and Giuseppe Polese.A framework for modeling and implementing visual notations with applications to software engineering. ACM Transactions on Software Engineering and Methodology (TOSEM), 13(4):431–487, 2004.

[3] S. Chawathe, H. Garcia-Molina, J. Hammer, K. Ireland, Y. Papakonstantinou, J. Unman, and J. Widom. The TSIMMIS Project: Integration of Heterogeneous Information Sources. In Proceedings of IPSJ Conference, Tokyo, Japan, October 1994.

[4] Peter J. Clarke, Vagelis Hristidis, Yingbo Wang, Nagarajan Prabakar and Yi Deng. A Declarative Approach for Specifying User-Centric Communication. Symposium on Collaborative Technologies and Systems (CTS), 2006.

[5] Department of Health. Health Insurance Portability and Accountability Act (HIPPA) ttp://dchealth.dc.gov/hipaa/hipaaoverview.shtm (June 2005).

[6] K. Fraser, S. Hand, R. Neugebauer, I. Pratt, A. Warfield, and M. Williamson. Reconstructing I/O. Technical Report UCAM-CL-TR-596, University of Cambridge, UK, 2004.

[7] E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R. H. Castain, D. J. Daniel, R. L.

Graham, and T. S. Woodall. Open MPI: Goals, concept, and design of a next generation MPI implementation. In Proceedings, 11th European PVM/MPI Users' Group Meeting, pages 97–104, Budapest, Hungary, September 2004.

[8] Ramezani, Fahimeh, Jie Lu, and Farookh Khadeer Hussain, "Task-based system load balancing in cloud computing using particle swarm optimization," International Journal of Parallel Programming 42, no. 5 pp. 739-754,2014.

[9] Sarker, Tusher Kumer, and Maolin Tang, "Performance-driven live migration of multiple virtual machines in data centers," In Granular Computing (GrC), 2013 IEEE International Conference on, pp. 253-258. IEEE, 2013.

[10] Vakilinia Shahin, "Energy-Efficient Resource Allocation in Cloud Computing Environments" IEEE Access, special section on future networks: architectures, protocols, and applications, Vol 4, 2016.

[11] Vakilinia Shahin behdad heidar pound Mohamed cheriet "Energy-Efficient Resource Allocation in Cloud Computing Environments" IEEE Access, special section on future networks: architectures, protocols, and applications, Vol 4, 2016.

[12] Wen, Wei-Tao, Chang-Dong Wang, De-Shen Wu, and Ying-Yan Xie, "An ACO-Based Scheduling Strategy on Load Balancing in Cloud Computing Environment," In 2015 Ninth International Conference on Frontier of Computer Science and Technology, pp. 364-369. IEEE, 2015.

[13] Zhu, Changpeng, Bo Han, Yinliang Zhao, and Bin Liu, "A Queueing-Theory-Based Bandwidth Allocation Algorithm for Live Virtual Machine Migration," In2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), pp. 1065-1072. IEEE, 2015.

[14] Zhen Xiao, Senior Member, IEEE, Weijia Song, and Qi Chen, "Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment," IEEE transactions on parallel and distributed systems, vol. 24, no. 6, pp. 1107-1117 IEEE, JUNE 2013.

[15] How VMware virtualization right-sizes IT infrastructure to reduce power consumption, 2009

[16] Energy-Aware Dynamic Virtual Machine Consolidation for Cloud Datacenters, IEEE transactions, volume 6, 2018.

[17] M. M. Asiri, G. Aldehim, F. A. Alotaibi, M. M. Alnfiai, M. Assiri and A. Mahmud, "Short-Term Load Forecasting in Smart Grids Using Hybrid Deep Learning," in IEEE Access, vol. 12, pp. 23504-23513, 2024, doi: 10.1109/ACCESS.2024.3358182.

[18] Martin Randles, David Lamb and A. Taleb-Bendiab, "A Comparative Study into Distributed Load Balancing Algorithms for Cloud Computing", Proceedings of IEEE 24th International Conference on Advanced Information Networking and Applications Workshops, pp. 551-556, 2010.

[19] Zenon Chaczko, Venkatesh Mahadevan, Shahrzad Aslanzadeh and Christopher Mcdermid, "Availability and Load Balancing in Cloud Computing", International Conference on Computer and Software Modeling, Vol. 14, pp. 134-140, 2011.

[20] Jaspreet Kaur, "Comparison of load balancing algorithms in a cloud", International Journal of Engineering Research and Applications, Vol. 2, No. 3, pp. 1669-1673, 2012.

[21] L Shakkeera et al., "Improving Resource Utilization Using Qos Based Load Balancing Algorithm for Multiple Workflows in Iaas Cloud Computing Environment", ictactjournals/paper/IJCTV4I2,pp.750 to757.

[22]"VPN vs. Cloud Computing ", http://www.examiner.com/article/vpn-vs-cloud-computing.

[23] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the art of virtualization," in Proc. of the ACM Symposium on Operating Systems Principles (SOSP'03), Oct. 2003.

[24] M. McNett, D. Gupta, A. Vahdat, and G. M. Voelker, "Usher: An extensible framework for managing clusters of virtual machines," in Proc. of the Large Installation System Administration Conference (LISA'07), Nov. 2007.

[25] Zhen Xiao, Senior Member, IEEE, Qi Chen, and Haipeng Luo," Automatic Scaling of Internet Applications for Cloud Computing Services", IEEE Transactions On Computers, vol. 63, no. 5, pp.1111-1123, May 2014.

[26] En-Hao Chang, Chen-Chieh Wang, Chien-Te Liu, Kuan-Chung Chen, Student Member, IEEE, and Chung-Ho Chen, Member, IEEE," Virtualization Technology for TCP/IP Offload Engine", IEEE Transactions on Cloud computing, Vol. 2, no. 2, pp.117-129, April-June 2014.

[27] Zhen Xiao, Senior Member, IEEE, Weijia Song, and Qi Chen, "Dynamic resource allocation using virtual machine in cloud computing environment", IEEE Transaction on Parallel and Distributed Systems, vol.24, no.6, pp.1107-1117, June 2013.

[28] N. Bobroff, A. Kochut, and K. Beaty, "Dynamic placement of virtual machines for managing sla violations," in Proc. of the IFIP/IEEE International Symposium on Integrated Network Management (IM'07), 2007.

[29] Bo Yin, Ying Wang, Louming Meng, and X.Qui, "A Multi-dimensional Resource Allocation Algorithm in Cloud Computing", Journal of Information & Computational science Vol.9 ,no.11,pp.3021-3028, 2012.

[30] Sukhpal Singh and Inderveer Channa,"Energy based Efficient Resource Scheduling: A step Towards Green Computing", International Journal of Energy, Information and Communications", Vol.5, Issue 2, pp.35-52, 2014

[31] Yi-Ju Chiang,Yen-Chieh Ouyang and Ching-Hsien," An Efficient Green Control Algorithm in Cloud Computing for Cost Optimization", IEEE Transaction on Cloud Computing,vol3,issue2,pp.145-155,2015.