UDC 62-50
DOI: 10.20535/SRIT.2308-8893.2023.4.10

A COMPREHENSIVE SURVEY ON LOAD BALANCING TECHNIQUES FOR VIRTUAL MACHINES

SUMAN SANSANWAL, NITIN JAIN

Abstract. Cloud computing is an emerging technique with remarkable features such as scalability, high flexibility, and reliability. Since this field is growing exponentially, more users are attracted to fast and better service. Virtual Machine (VM) allocation plays a crucial role in cloud computing optimization; hence, resource distribution is not impacted by machine failure and is migrated with no downtime. Therefore, effective management of virtual machines is necessary for increasing profit, energy-saving, etc. However, it could utilize the virtual machine resources more efficiently because of the increased load, so load balancing is more concentrated. The predominant purpose of load balancing is to balance the available load equally among the nodes to avoid overloading or underloading problems. The present study conducted an extensive survey on virtual machine placement to describe the application of prediction algorithms and to provide more efficient, reliable, high response, and low overhead VM placement. Furthermore, the survey attempted to overview the challenges in load balancing in VM placement and various ideas of state-of-the-art techniques to resolve the issues..

Keywords: virtual machine allocation, load balancing, cloud computing, overloading, physical machine, data center.

INTRODUCTION

Cloud computing is now becoming vital for hosting several IT services that provide various on-demand VR (Virtual Resources). The cloud service providers used large-scale DC (data center) with more physical machines. Virtualization is more beneficial in data centers for providing the VM comprising a software layer known as a VVM Monitor. This VMM enables the controlling of shared physical machine resources, thereby increasing VM security but accommodating multiple VM in a single physical machine remains a challenging problem. Due to this problem, there is a chance of overutilizing PM, degrading it, or wasting high-cost resources.

Further, the power consumption of cloud DC mainly occurs by physical machines, so proper VM placement associated with dynamic management greatly mitigates DC power consumption, improving the profit and throughput and preventing SLA violations. However, VM placement employs a widespread and expensive VM migration process. If improper placement occurs, it will lead to the destruction of data centre performance. Furthermore, balancing the request's workload and allocating appropriate tasks to the appropriate VM is also considered challenging. Hence load balancing is a crucial factor to be considered with the increasing requests and impulsive arrival patterns. Load balancing is the even classification of the task processed in-between more CPUs, storage devices, and network links which deliver fast service with more efficiency. This is obtained

using hardware/software devices and multiple servers that appear as a computer clustering. In addition, load balancing improves the efficiency of distributed or parallel systems through load redistribution. Load balancing algorithms are classified into static and dynamic algorithms. The static algorithm is simple and needs minimized runtime overhead, whereas a dynamic system is utilized in most of the modern load-balancing approaches due to its flexibility and robustness. Likewise, there exist four kinds of load-balancing policies, which are location policies, transfer policies, selection policies, and information policies. Other such objectives of load balancing include reducing carbon emission and energy consumption, resource provisioning, avoiding bottlenecks, and achieving QoS requirements.

To overcome the prevailing limitations and obtain end-user satisfaction, high-quality and effective methodologies must be adopted to support the optimization of VM load balancing. Therefore, the study's main contribution is to provide a comprehensive survey of the existing methods dealing with virtual machine load balancing by the factors affecting the cloud computing process.

Objective:

- To analyze several virtual machine placement and load balancing techniques in the existing literature.
- To overview the prevailing challenges in load balancing in virtual machine processing and to provide a comprehensive outlook to rectify the issues.
- To outlook the recent trends for the optimization of load balancing in virtual machine allocation.

The paper is organized as follows: Section 2 deals with the predictive virtual machine placement methods with various algorithms, and Section 3 reviews prevailing load balancing techniques such as static and dynamic methods. Section 4 summarizes the advantages of load balancing in virtual machine allocation, and Section 5 overviews the performance metrics for evaluating load balancing in virtual machines. Section 6 provides the recent trends in this concept, and Section 7 deliberates on the challenges and research gaps of load balancing for virtual machines. Followed by Section 8 concludes the work.

PREDICTIVE VIRTUAL MACHINE PLACEMENT METHODS

Various predictive virtual machine placement methods are designed and suggested for the CC environment. Besides, implementing all these methods for enhancing the placement process of virtual machines by utilizing historical data. The predictive methods for the VM placement are classified as the following [1].

- Ensemble-based scheme.
- Hybrid scheme.
- Exponential smoothing predictor-based scheme.
- Dynamic programming-based scheme.
- Grey model-based scheme.
- Fractal based schemes.
- Bayesian-based scheme.
- Neural network-based scheme.
- SVM based scheme.
- Queuing based scheme.
- Markov based scheme.

- Hidden Markov Model based scheme.
- ARIMA model-based scheme.
- Regression-based scheme.

The following are a few existing virtual placement schemes that utilize the above-said algorithms. This study [2] attempted to predict resource requirements for virtual machines to improve the process of virtual machine placement. Besides, the study indicated that the time-reliable hidden Markov model replicated the properties of CPU utilization data and determined the CPU's future usage. Nevertheless, the study applied only univariate normal distribution to determine resources, whereas the multivariate normal distribution to determine multiple resources could be useful. Likewise, [3] provided a Markov predicting framework for forecasting the future under-utilized/over-utilized physical machines and preventing unnecessary and immediate migration of virtual machines. Besides, this study utilized the cloud sim toolkit evaluated using random forest and Planet Lab datasets. Finally, the current usage of the CPU of every physical machine has been compared with upper and lower thresholds for recognizing the status.

Moreover, determine the future state of physical machines by implementing the Markov model. This study [4] introduced a framework for identifying the relationship of resources amid virtual machines by utilizing ARIMA-based determinations. Further, the study analyzed resource utilization after the placements of two virtual machines on the same physical machine, and also the study named this system an affinity model. Similarly, this study [5] implemented automata for enhancing the usage of resources as well as mitigating the usage of power. Furthermore, this study considered the variations in user demands for estimating overloaded physical machines. Due to the prevention of physical machine overload, this system improves the utilization of resources, mitigates the amount of migration, and shuts the idle physical machines to mitigate the utilization of power. Finally, the study stated that this method was executed in the cloudsim toolkit by utilizing Planet Lab dataset. Nevertheless, this cannot detect underutilized physical machines.

REVIEW OF PREVAILING LOAD BALANCING METHODS

This study surveyed the literature on prevailing load-balancing methods and comprehensively reviewed certain studies. Besides, the load balancing methods are segregated into dynamic and static, based on the system's state. Load balancing is needed to improve resource utilization, reducing the completion and response time for the tasks on the cloud. This study [6] suggested a method in which it considered QoS, number of migrations as well as response time as the parameters of load balancing. Further, tasks with less priority have been transferred from one virtual machine to another when overloaded with virtual machines. This method can be improvised with other algorithms like ACO and PSO.

Static load balancing methods. The static load balancing method doesn't require knowledge of a system's current state; it requires knowledge of the system resources like processing power, storage capacity, memory, and execution time in advance. Besides, the static load balancing methods don't allow resource allocation at execution time. Also, these methods are easy to execute and implement, but they are beneficial to small networks or systems with a minimum amount of resources. On the other hand, as they don't consider the present state of the system, these methods aren't beneficial for computing systems that perform distributed computing. Moreover, they need to permit the detection of connected server machines at the execution time, thereby leading to uneven resource distribution.

Dynamic load balancing methods. Since the static load balancing techniques are not suitable for the distributed computing system, the dynamic load balancing methods are suitable in a cloud computing environment. The following are different load balancing methods, which rely on the criteria of the load balancer:

- Cluster-based load balancing.
- Task-based load balancing.
- Agent-based load balancing.
- Hybrid load balancing.
- Natural phenomena based on load balancing.
- General load balancing.
- Cluster-based load balancing.

This study [7] addressed a heuristic method for load balancing based on (LB-BC) Bayes and Clustering for overcoming the difficulties of prevailing load balancing techniques. This technique is based on Bayes' theory and has accomplished long-term load balancing. This computes the posterior probability of the physical hosts and integrates with clustering for picking an optimal host. Further, this considered the parameters like load balancing effect, standard deviation, and the number of requested tasks. Then, it was compared with the dynamic load balancing, leading to increased time and minimal standard deviation. This method only works in localized areas, but further enhancement can be made for working in a real-time environment and a wide area network. This study [8] presented a cluster-based method for improvising intercloud communication in real-time and dynamic multi-media for load balancing. This method has a two-step process. The first step is to develop the cluster to monitor the activities, handle platform difficulties, and meet the satisfactory quality of service and demands for hosts based on a hello-packet broadcast for all the servers. In the second step, it decides on transfer job requests. When this method was compared with HFA, WCAP, and ant colonies, the suggested method produced an improved response time. In addition, this method could be improvised for a real-time environment, in which the intermediate nodes are congested, and owing to reduce the data loss because of congestion by utilizing communication jobs instead of computation jobs.

This study [9] presented a cluster-based load-balancing method for overcoming load distribution issues. Besides, this integrated the concept of KUHN and genetic algorithm and created a task allocation strategy by grouping the tasks into clusters and distributing them in a cooperating node. As a result, this method provided improvised task distribution and response time among data center nodes. Similarly, this study [10] created a hierarchical model to self-schedule the schemes for improving the scalability and load balancing of the cloud system. Besides, this method can extract in a heterogeneous and homogeneous environment. In addition, this study has implemented the schemes on a large scale by utilizing various computation applications. Finally, the outcomes of the study depicted improvised scalability and overall performance, as well as decreased communication overhead. The further analysis deals with a testing algorithm for large-scale loops and clusters with dependencies.

Task-based load balancing. This study [11] presented a network-aware task placement method for reducing task completion time, data cost, and transmission time. The study stated that the three challenges faced by tasks are the availability of resources dynamically changes resulting in access over time; data fetching time relies on the task's location and size; the load on the path significantly impacts the data access latency. Therefore, the study must consider loading over the path dur-

ing scheduling to minimize this latency. The study's outcomes depicted that the suggested method has significantly reduced the task's completion time and increased resource utilization.

This study [12] suggested a scheduling technique for reducing the resource competition between high device load and tasks based on the weighted random scheduling method. The tasks are assigned by considering parameters such as communication delay, time, and cost. Besides, the study was analyzed with MATLAB software by utilizing workflow for generating the dataset. Also, the study analyzed the dataset, which included a large set of tasks with transmission delay, cost, and time. Moreover, the study considered device dependency, task arrival time, and task structure. The study's outcomes depicted that multiple schedules have seen improvement in parameters like execution cost and task completion for the devices. Nevertheless, it still needs to calculate the optimal value for parameters that could be improvised in further analysis.

Agent-based load balancing. The multi-agent-based load-balancing framework helps increase resource utilization [13]. This executed both the receiver originate method, as well as the sender, originated method for reducing the waiting time of tasks and also for assuring SLA. This method incorporated the agents like NA (Negotiator Ant) agent, DCM (Datacenter Monitor) Agent, as well as VMM (Virtual Machine Monitor) Agent. Among these, the virtual machine monitor agent supports every virtual machine in the system and retains the information on bandwidth, CPU, and memory by utilizing virtual machines for monitoring the load. Besides, the datacentre monitor agent executes information policy by utilizing the available information from the virtual machine monitoring agent and categorizing the virtual machines relying upon various characteristics. Also, this initiates the negotiator and agent that moves to various other data centers for identifying the available virtual machines' status. From the experimental analysis, the study stated that the suggested method was more effective, improving the response time and reducing the makespan time.

The (SVLL) selection of virtual machines with the least load balancing technique for the distribution of tasks increased the cloud computing performance [14]. This model computes a load of every virtual machine and assigns tasks to evaluate based on the virtual machine's load rather than the number of tasks assigned to virtual machines. Besides, the study implemented the SVLL method with various task scheduling methods like shortest job first and first came first serve methods, in which the outcomes of the study denoted that the suggested method has improvised in total finishing time and total waiting time. In addition, this method was employed with basic task scheduling methods for better results.

This study [15] developed a load-balancing method by integrating round-robin features and shortest-job-first scheduling algorithms. This method stores long and short tasks in separate queues and utilizes dynamic task scheduling quantum to balance waiting time among the tasks. Besides, this study has taken into account the issues of starvation as well as throughput. Also, they executed the experiment on the cloud tool. As a result, the experimental analysis showed that response time, waiting time, and the turnaround time was reduced. In addition to that, long-task starvation was also minimized. Nevertheless, the task quantum was not efficient in balancing the tasks, but it could be improvised in further analysis.

The hybrid load-balancing method. This study [16] employed a hybrid algorithm for optimizing the system's performance by integrating throttled and round-robin load balancing methods with a service-proximity broker and performance-optimized service broker algorithm. Besides, the study suggested one

load balancing and three service broker methods. The study denoted them as CA (Cost Aware) and LA (Load Aware) algorithms for high utilization of resources. However, although the LA algorithm offers low processing time, it can generate high costs, whereas CA reduces cost. Moreover, the service broker algorithm decides on the server to users' requirements, which might increase cost or processing time.

In contrast, the service proximity algorithm decides on the data center near to client's region. Finally, the study integrated all the algorithms, and the outcomes of the study denoted that response time and processing time have significantly reduced. Nevertheless, further analysis deals with the improvisation of system performance. The development of an efficient CLB (Cloud Load Balancing) framework is needed to overcome the server failure response in the event of several user requests. Several studies have developed a framework that considers the loading and server processing for minimizing the server problems for handling various computation requests. Also, they presented a load-balancing method for virtual and physical web servers to preserve the information regarding computing power, priority, and server loading. Even though this framework provides high scalable performance, it can increase response time.

COMPARATIVE ANALYSIS OF STATIC AND DYNAMIC LOAD BALANCING TECHNIQUES

Table provides a comprehensive comparative analysis of the existing load-balancing algorithms.

Comparative Analysis between The Existing Load Balancing Algorithms

S. No	Type of Load Balancing algorithm in VM	Load Balancing Algorithm	Parameters enhanced	Merits	Demerits
1	Static	Weighted-round robin algorithm [17]	Waiting and response time	Utilize all resources in a balanced manner. Ensuring fairness in every allocation	Execution time Prediction is not possible. High Migration time
2		Opportunistic load-balancing algorithm [18]	User discomfort cost and reduction	The end-user achieves better accuracy and comfort maximization	Comfort maximization might lead to raised costs and energy
3		Software-Defined Networking based load-balancing algorithm [19]	Cost, response time, and scalability	Effective user request processing	Increased energy consumption
4	Dynamic	Ant colony algorithm [20]	Makespan, response time, scalability	Good scalability, Fault tolerance, and obtaining load balancing for Complex networks.	High power consumption. Less throughput
5		Deadline-constrained based dynamic load-balancing algorithm [21]	Task rejection ratio, makespan	Increases the utilization ratio	Increased consumption of cost
6		Honey-bee foraging algorithm [22]	Response time, throughput	Less waiting time and Increased system diversity	High response time Less throughput

BENEFITS OF LOAD BALANCING IN VIRTUAL MACHINES

Ideally, these solutions can be implemented when performing the placement of virtual machines. Decreasing the number of physical machines as well as consolidating virtual machines could be utilized for solving cloud-spot issues. Reducing the migrations of virtual machines by predicting future workloads will prevent unnecessary migrations of virtual machines. Future pages could be identified by mitigating transmitted pages by properly predicting the workload of applications. Consequently, the number of transmitted pages could be diminished in the pre-copy approach.

The load balancer offers flexibility for balancing the server's workload by traffic distribution across multiple servers. Further, load-balancing targets mimic a software infrastructure via Virtualization. This runs physical load-balancing software on VM. In addition, availability, performance, scalability, and reliability are the major metrics of load balancing.

Availability. The mechanism of load balancing assures an efficient offer of service. Moreover, the loads will be effectively distributed in terms of server unavailability.

Performance. An effective load balancing provides cloud applications as well as cloud services for responding faster when compared to the average completion time. In addition, execution time is also decreased via effective compression methods and catching mechanisms.

Scalability. The major benefit of the load-balancing technique is that some servers can be easily included without any disturbance, and the applications can be smoothly performed via the load-balancing servers.

Reliability. The reliability of cloud services was secured by the redundancy of servers in which the applications could be hosted. Even in failure cases, the cloud-serving resources will function, and its services will be redirected to other locations in the cloud.

PERFORMANCE METRICS IN THE EVALUATION OF LOAD BALANCING FOR VIRTUAL MACHINES

Various virtual machine load balancing metrics are present for assessing load balance performance. These metrics were reflected in diverse task scheduling behavior. The following are the load balancing metrics.

Load variance. Consider that there exists n number of hosts in the data center. The usage of host i can be expressed as $U(host_i)$, whereas the average usage of every host can be calculated as

$$avg\,(U_t) = \frac{1}{2}\sum_{i=1}^{n} host_i .$$

Makespan time. Makespan time is known as the longest-processing time on every host. Also, it is a normal criterion for accessing scheduling algorithms. Retaining load balance is for shortening the makespan time.

Overloaded hosts. The overload threshold can be denoted as $T(U_t)$, for n number of hosts, the host utilization can be expressed as $U(host_i)$, and the overload hosts is expressed as the following, $Num(T(U_t)) \le U(host_i)$.

Throughput. Throughput deals with system performance. A maximum number of tasks are executed to accomplish high performance within the minimal completion time.

SLA violations. Similarly, this also deals with the performance of the system. The virtual machines can't fetch adequate resources from the host, so the host isn't well-balanced. Thus, SLA violations must be reduced.

Turnaround time. Turnaround time is defined as the time systems take from the request submission to a response from the server. And turnaround time can be calculated as

$$\text{Turnaround time} = C_t - C_T .$$

From the above equation C_t refers to completion time, and GT refers to generation time.

Overhead. Generally, overhead occurs because it increases the communication cost or takes more time to migrate from one virtual machine to another. Good load-balancing algorithms will decrease the overhead.

Resource usage. Good performance usually deals with the proper resource usage among nodes. This will be beneficial for measuring if the nodes are underloaded or overloaded.

Fault tolerance. This enhances the systems such that the single failure point doesn't impact the entire system. Besides, the load balancing algorithm must be designed in a way where the failure of one node must not affect the system.

Response time. Generally, response time is the time taken by load balancing techniques to users. Lesser response time indicates better system performance. Therefore, load balancing will be more beneficial for the entire cloud by decreasing the response time of cloud servers and task scheduling issues; the following articles discuss the response time in virtual machines.

This study [23] suggested TMA (Throttled Modified Algorithm) improves the response time of virtual machines on CC (Cloud Computing) to improvise a performance. Besides, this study simulated the suggested method with the clouds tool; the evaluated outcomes showed improved processing time and response time.

In this study [24], a firefly load balancing technique was utilized to solve the load imbalance problems in a cloud server to enhance the learners' user experience. The suggested method needs a cloud-server mapping method for various virtual machine methods, ensuring the users receive the content without delay. From the experimental analysis, the study stated that, compared to the existing method, the suggested method showed less response time.

RECENT TRENDS OF LOAD BALANCING IN VIRTUAL MACHINE ALLOCATION

This study [25] suggested that response time was similar to execution time in every task, and this parameter should be minimized. This determines the virtual machine status based on the current load. Later, the tasks are eliminated from the machine with additive load, which depends on the virtual machine's condition. Finally, it will be transferred to the appropriate VM, which is the criteria to assign

tasks to virtual machines based on the least distance. The outcomes of the cloudsim tool evaluation showed that response time was improved compared to existing algorithms. Additionally, the degree of load imbalance has also seen some improvements.

The main aim of task scheduling incorporates scheduling resources and reducing the schedule's objective. This study [26] suggested a mean grey-wolf optimization technique to enhance the system's performance and reduce scheduling problems. The primary objective of this study is to reduce energy consumption and makespan time. This was evaluated by utilizing the cloudsim tool. The study showed that the suggested algorithm had better results than the prevailing methods.

This suggested method in this study [27] attempted to avoid SLA violations via power optimization and optimal cloudlet by reducing the migrations of virtual machines. Besides, the SLA reduction system incorporated three parts a scheduling algorithm, a MinVM scheduling algorithm, and a credit-based virtual machine migration algorithm. When considering the scheduling algorithm, it efficiently schedules the cloudlets to VMs based on the host's processing time. Likewise, the MinVM scheduling algorithm schedules the cloudlets to VMs based on counts of cloudlet allocation to every virtual machine. And the credit-based algorithm utilizes the virtual machine's credit to take virtual machine migration.

CHALLENGES AND RESEARCH GAP

The most challenging task in virtual technology is virtual machine placement on the physical machine under optimal conditions in cloud-data centers. Further, the virtual machine placement can result in managing resources and preventing the wastage of resources. Minimizing energy consumption, cost reduction, utilization of resources, and presentation of best QoS are significant challenges in the cloud computing environment. Since only a few studies focus on privacy and security issues, in further analysis, security is a crucial factor that must be focussed on. Besides, attackers can steal the secrets from other tenants by utilizing side-channel attacks based on shared resources since the virtual machines from various tenants might be located at one physical machine, thereby threatening data security in a cloud computing environment. The following are certain limitations that should be considered,

- The forecasting approaches employed in predictive virtual machine placement schemes could be enhanced to better deal with non-linear and linear loads.
- Moreover, the predictive virtual machine placements in the multi-cloud and multi-site cloud environments must be studied for further analysis.
- Even though dynamic power management can be implemented to improvising DCs energy efficiency, only a few studies have suggested this approach in their literature.
- One of the significant problems avoided by various studies is DDoS attacks that could be originated from malicious virtual machines by uplifting the resource demands and introducing several unnecessary virtual machine migrations.
- The integration of predictive virtual machine placement methods with prevention and intrusion detection systems must be investigated to recognize the true demands and increase the DC's security.

- Studies must design and apply low-overhead placement methods in developing technologies like mobile and cloudlets in the future.
- In future studies, context-aware virtual machine placement must be designed for the environment, like predicting mobile patterns, vehicular CC, and connectivity problems.

In recent years, cloud computing has seen rapid growth and advanced research in computation and data based on practical and theoretical aspects. Nevertheless, cloud computing researchers face several problems in which load balancing is more challenging and needs special attention. Besides, issues like user QoS (Quality of Service) satisfaction, virtual machine security, resource usage, and virtual machine migration must be considered to find a feasible solution to improve resource utilization. Additionally, various problems like the migration of virtual machines, resource utilization, QoS satisfaction, and migration of virtual machines need equal attention for finding the optimal solution to enhance the optimal solution to improve the utilization of resources.

The following are certain load-balancing problems.

Geographically distributed nodes. Generally, the data centers in the cloud are geographically-distributed. In these data centers, for effective system execution according to the request of users, the spatially distributed nodes were treated as a single location system. Besides, certain load balancing methods were designed for a small area. For example, they don't consider communication delay, network delay, and the distance between distributed resources, users, and computing nodes. Nevertheless, the nodes situated at various locations are challenging since these algorithms are unsuitable for these environments. Therefore, load-balancing techniques for distantly located nodes must be considered [28].

Migration of Virtual Machines. Virtualization allows for the creation of numerous virtual machines on one physical machine. As a result, virtual machines are generally independent and possess various configurations. Besides, if the physical machine is overloaded, certain virtual machines must migrate to a distant location using the virtual machine migration load balancing method [29].

Heterogeneous Nodes. During earlier research in load balancing, several studies have theorized about homogeneous nodes. But, usually, in cloud computing, users' requirements dynamically change, which needs executing time for efficient resource utilization and decreasing response time. Thus, introducing an effective load-balancing method for the heterogeneous environment is more challenging [30].

Storage management. Cloud storage solved the issues of the conventional storage system, which required higher hardware costs and personnel management. Further, the cloud allows users to store data heterogeneously without access issues as there is a rapid increase in cloud storage, data replication for data consistency, and effective access. However, because of duplicate storage policies, full data replication is ineffective. The partial replication could be adequate. However, there are certain issues in the dataset's availability, and there might be increased complexities in load balancing methods [31].

Scalability of the load balancer. The scalability and on-demand availability of cloud services allow users to access the services for rapidly scaling up or scaling down. Therefore, a load balancer must consider rapid variations by system topology, storage, and computing power to efficiently facilitate these variations [32].

Complexity of algorithms. In a cloud computing environment, usually, the algorithms must be easier and simple to implement. Besides, complex algorithms may diminish the efficiency and performance of cloud systems [33].

CONCLUSION

In general, a cloud indicates a distinct IT environment designed for the proper functioning of remotely providing scalable and measurable IT resources. The paper's main objective is to consolidate the prevailing VM placement and load-balancing methodologies. Further various challenges to the enhancement of effective VM and load-balancing algorithms are also discussed. This survey lets the users look at recent trends in VM placement and load balancing, enabling them to frame an effective research methodology with maximum profit and minimum cost.

Declaration. I confirm that this work is original and has not been published elsewhere, nor is it currently under consideration for publication elsewhere.

Acknowledgments

- None.

Funding

- Any organization/institute/agency did not fund this research work.

Competing Interests

- None of the authors have any competing interests in the manuscript.

Availability of data and material

- Not Available.

Code availability

- Not Available.

Compliance with ethical standards

Ethical statement

- No human participants or animals are involved in this research.

Consent statement. I confirm that any participants (or their guardians if unable to give informed consent, or next of kin, if deceased) who may be identifiable through the manuscript (such as a case report) have been allowed to review the final manuscript and have provided written consent to publish.

REFERENCE

1. M. Masdari and M. Zangakani, "Green cloud computing using proactive virtual machine placement: challenges and issues," *Journal of Grid Computing*, pp. 1–33, 2019.
2. H.L. Hammer, A. Yazidi, and K. Begnum, "An inhomogeneous hidden Markov model for efficient virtual machine placement in cloud computing environments," *Journal of Forecasting*, vol. 36, pp. 407–420, 2017.
3. S.B. Melhem, A. Agarwal, N. Goel, and M. Zaman, "Markov prediction model for host load detection and VM placement in live migration," *IEEE Access*, vol. 6, pp. 7190–7205, 2017.
4. X. Fu and C. Zhou, "Predicted affinity based virtual machine placement in cloud computing environments," *IEEE Transactions on Cloud Computing*, vol. 8, pp. 246–255, 2017.
5. M. Ranjbari and J.A. Torkestani, "A learning automata-based algorithm for energy and SLA efficient consolidation of virtual machines in cloud data centers," *Journal of Parallel and Distributed Computing*, vol. 113, pp. 55–62, 2018.
6. K.R. Babu and P. Samuel, "Enhanced bee colony algorithm for efficient load balancing and scheduling in cloud," in *Innovations in bio-inspired computing and applications*. Springer, 2016, pp. 67–78.
7. J. Zhao, K. Yang, X. Wei, Y. Ding, L. Hu, and G. Xu, "A heuristic clustering-based task deployment approach for load balancing using Bayes theorem in cloud environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, pp. 305–316, 2015.

8. B. Kang and H. Choo, "A cluster-based decentralized job dispatching for the large-scale cloud," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, pp. 1–8, 2016.
9. F. Zegrari, A. Idrissi, and H. Rehioui, "Resource allocation with efficient load balancing in cloud environment," in *Proceedings of the International Conference on Big Data and Advanced Wireless Technologies*, 2016, pp. 1–7.
10. Y. Han and A. T. Chronopoulos, "Scalable loop self-scheduling schemes for large-scale clusters and cloud systems," *International Journal of Parallel Programming*, vol. 45, pp. 595–611, 2017.
11. H. Shen, A. Sarker, L. Yu, and F. Deng, "Probabilistic network-aware task placement for mapreduce scheduling," in *2016 IEEE International Conference on Cluster Computing (CLUSTER)*, pp. 241–250.
12. Y. Xin, Z.-Q. Xie, and J. Yang, "A load balance oriented cost efficient scheduling method for parallel tasks," *Journal of Network and Computer Applications*, vol. 81, pp. 37–46, 2017.
13. S. Keshvadi and B. Faghih, "A multi-agent based load balancing system in IaaS cloud environment," *International Robotics & Automation Journal*, vol. 1, pp. 1–6, 2016.
14. T. Aladwani, "Impact of selecting virtual machine with least load on tasks scheduling algorithms in cloud computing," in *Proceedings of the 2nd International Conference on Big Data, Cloud and Applications*, 2017, pp. 1-7.
15. S. Elmougy, S. Sarhan, and M. Joundy, "A novel hybrid of Shortest job first and round Robin with dynamic variable quantum time task scheduling technique," *Journal of Cloud computing*, vol. 6, pp. 1–12, 2017.
16. R.K. Naha and M. Othman, "Cost-aware service brokering and performance sentient load balancing algorithms in the cloud," *Journal of Network and Computer Applications*, vol. 75, pp. 47–57, 2016.
17. K. Manojkumar and P. Kanagaraju, *Enhanced load balancing algorithm to reduce response time and waiting time by incorporating weighted round robin and honey bee behaviour algorithm in cloud computing.*
18. M.B. Rasheed, N. Javaid, M.S.A. Malik, M. Asif, M.K. Hanif, and M.H. Chaudary, "Intelligent multi-agent based multilayered control system for opportunistic load scheduling in smart buildings," *IEEE Access*, vol. 7, pp. 23990–24006, 2019.
19. H. Zhong, Y. Fang, and J. Cui, "Reprint of "LBBSRT: An efficient SDN load balancing scheme based on server response time," *Future Generation Computer Systems*, vol. 80, pp. 409–416, 2018.
20. S. Dam, G. Mandal, K. Dasgupta, and P. Dutta, "An ant-colony-based meta-heuristic approach for load balancing in cloud computing," in *Applied Computational Intelligence and Soft Computing in Engineering*, IGI Global, 2018, pp. 204–232.
21. M. Kumar and S. Sharma, "Deadline constrained based dynamic load balancing algorithm with elasticity in cloud environment," *Computers & Electrical Engineering*, vol. 69, pp. 395–411, 2018.
22. V. Bhavya, K. Rejina, and A. Mahesh, "An Intensification of Honey Bee Foraging Load Balancing Algorithm in Cloud Computing," *International Journal of Pure and Applied Mathematics*, vol. 114, pp. 127–136, 2017.
23. N.X. Phi, C.T. Tin, L.N.K. Thu, and T.C. Hung, "Proposed load balancing algorithm to reduce response time and processing time on cloud computing," *Int. J. Comput. Networks Commun.*, vol. 10, pp. 87–98, 2018.
24. K. Sekaran, M.S. Khan, R. Patan, A.H. Gandomi, P.V. Krishna, and S. Kallam, "Improving the response time of m-learning and cloud computing environments using a dominant firefly approach," *IEEE Access*, vol. 7, pp. 30203, 2019.
25. L. Xingjun, S. Zhiwei, C. Hongpong, and B.O. Mohammed, " A new fuzzy-based method for load balancing in the cloud based Internet of thing usingagrey wolf optiization algorithm,'' *International Journal of Communication Systems*, vol. 33, p. e4370, 2020.

26. G. Natesan and A. Chokkalingam, "Task scheduling in heterogeneous cloud environment using mean grey wolf optimization algorithm," *ICT Express*, vol. 5, pp. 110–114, 2019.
27. M.K. Halili and B. Cico, "SLA management for comprehensive virtual machine migration considering scheduling and load balancing algorithm in cloud data centers," *International Journal on Information Technologies & Security*, vol. 12, 2020.
28. A.K. Kiani and N. Ansari, "On the fundamental energy trade-offs of geographical load balancing," *IEEE Communications Magazine*, vol. 55, pp. 170–175, 2017.
29. N.H. Shahapure and P. Jayarekha, "Virtual machine migration based load balancing for resource management and scalability in cloud environment," *International Journal of Information Technology*, pp. 1–12, 2018.
30. X. Shao, M. Jibiki, Y. Teranishi, and N. Nishinaga, "An efficient load-balancing mechanism for heterogeneous range-queriable cloud storage," *Future Generation Computer Systems*, vol. 78, pp. 920–930, 2018.
31. S. Subalakshmi and N. Malarvizhi, "Enhanced hybrid approach for load balancing algorithms in cloud computing," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 2, pp. 136–142, 2017.
32. T.G. Rodrigues, K. Suto, H. Nishiyama, N. Kato, and K. Temma, "Cloudlets activation scheme for scalable mobile edge computing with transmission power control and virtual machine migration," *IEEE Transactions on Computers*, vol. 67, pp. 1287–1300, 2018.
33. M. Xu, W. Tian, and R. Buyya, "A survey on load balancing algorithms for virtual machines placement in cloud computing," *Concurrency and Computation: Practice and Experience*, vol. 29, p. e4123, 2017.

Received 21.11.2022

INFORMATION ON THE ARTICLE

Suman Sansanwal, ORCID: 0000-0001-8485-0931, Chandigarh University, Punjab, India, e-mail: sumanphd27@gmail.com

Nitin Jain, Chandigarh University, Punjab, India, e-mail: nitin@gmail.com

КОМПЛЕКСНИЙ ОГЛЯД ТЕХНІК БАЛАНСУВАННЯ НАВАНТАЖЕННЯ ДЛЯ ВІРТУАЛЬНИХ МАШИН / Суман Сансанвал, Нітін Джайн

Анотація. Хмарні обчислення — це нова техніка з чудовими характеристиками, такими як масштабованість, висока гнучкість і надійність. Оскільки ця сфера експоненціально зростає, швидке та якісне обслуговування приваблює більше користувачів. Розподіл віртуальної машини (VM) відіграє вирішальну роль в оптимізації хмарних обчислень; на розподіл ресурсів не впливає збій машини та перенесення відбувається без простоїв. Ефективне керування віртуальними машинами необхідне для збільшення прибутку, енергозбереження тощо. Однак воно може більш ефективно використовувати ресурси віртуальної машини через збільшення навантаження, тому балансування навантаження є більш концентрованим. Переважна мета балансування навантаження — рівномірно збалансувати доступне навантаження між вузлами, щоб уникнути проблем із перевантаженням або недовантаженням. У дослідженні виконано розширений огляд щодо розміщення віртуальних машин, щоб описати застосування алгоритмів прогнозування та забезпечити більш ефективне, надійне розміщення віртуальної машини з високою відповіддю та низькими накладними витратами. Крім того, у ході роботи зроблено спробу оглянути проблеми балансування навантаження у розміщення віртуальної машини, а також різні ідеї щодо сучасних методів вирішення цих проблем.

Ключові слова: розподіл віртуальної машини, балансування навантаження, хмарні обчислення, перевантаження, фізична машина, центр оброблення даних.