ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/365103475

A Quality of Service Aware VM Placement for User Applications in Cloud Data Center

Conference Paper · October 2022

DOI: 10.1109/CCCI55352.2022.9926493

CITATIONS
0

READS
29

2 authors, including:

Alireza Shameli-Sendi
Shahid Beheshti University
46 PUBLICATIONS **892** CITATIONS

SEE PROFILE

All content following this page was uploaded by Alireza Shameli-Sendi on 07 December 2022.

The user has requested enhancement of the downloaded file.

A Quality of Service Aware VM Placement for User Applications in Cloud Data Center

Arash Hadadi and Alireza Shameli-Sendi
Faculty of Computer Science and Engineering, Shahid Beheshti University (SBU), Tehran, Iran
Email: {a.hadadi@mail.sbu.ac.ir, a_shameli@sbu.ac.ir}

Abstract—The placement of virtual machines in cloud data centers faces the challenge of multiple optimization goals. For example, the benefits of cloud providers are to reduce the cost of cloud resources and energy consumption, while the requirement of cloud users are to increase the efficiency and service quality of applications. Providing a solution that reduces the cost of cloud infrastructure to an optimal level and also satisfies users in terms of service quality, is always one of the challenges of virtual machines placement. Service quality criteria are broad and some of them are related to the quality of experience that has been considered in this paper. Many current placement techniques do not take into account the workload fluctuations, response times, and user preferences. In this paper, a cloud user application, which will be in the form of virtual machines, is placed in the data center based on the service quality parameters. By changing the service quality parameters at any time, replacement for that application is performed with the least overhead. Our simulation results show that we have been able to improve the quality of service by 37% compared to previous work for 30 applications that lead to the placement of 90 virtual machines in a data center. Our simulation results for 30 applications show that previous approaches that are not sensitive to service quality, experience a 60% reduction in service quality when the application load is increased by 50%.

Index Terms—VM placement, optimal placement, quality of service, quality of experience, data center.

1 Introduction

Due to the vast benefits of cloud services that are no longer hidden from anyone today, the migration of users or organizations to this platform has increased more than before. The needs of cloud users are in the form of a series of virtual machines. Optimal placement of these virtual machines in the cloud is an important challenge that can be examined from several dimensions in the cloud infrastructure (e.g., reducing energy consumption, resource management, security, quality of service, etc.) [1], [2], [4]–[9]. What is focused on in this paper is the issue of optimal placement with respect to maintaining the quality of customer service.

Service quality [36], which is a non-functional aspect of applications, includes a wide range of indicators that are classified based on different models and standards [10]. Coverage of all these indicators, which are qualitative, cannot be covered easily in the issue of virtual machines placement. Service quality alone cannot satisfy users in interacting with software. It varies from user to user and from software to software. For this purpose, the quality of experience [25], [37], [38] is considered and is a criterion for measuring the satisfaction of end users of the software. Quality of experience is defined based on linear relationships from a specific set of service quality criteria that are relevant to execution time [3], [10]. The main criterion in this category is the service response time, which is the processing time from the time the user requests. In other words, response time is the sum of processing time in virtual machines and the latency in the network [3].

• *Alireza Shameli-Sendi (Corresponding author).

In different software, the workload changes at different times and this change increases the processing load of resources. Various solutions have been proposed to properly manage workload fluctuations and reduce response time. The first solution is to increase the resources allocated to the software, which is called vertical scalability. In this solution, by allocating more resources to the software, the expected response time can be achieved [12]. In the second solution, if no more resources are available in the physical machine, we need to migrate the virtual machine to a physical machine with more powerful resources. Certainly, any migration will impose a traffic overhead and reduce the quality of service at the time of migration [40]–[42]. The third solution is in the field of software architecture decisions, which is called horizontal scalability. Our solution is not in the third category at all. Our assumption is that the software has an appropriate architecture and based on that, the desired virtual machines are selected. Our solution is in the second category, that is, after the first placement, based on traffic fluctuations, we will decide to migrate some virtual machines owned by a software to new locations.

In this work, three-tier software whose deployment model is the placement of layers in different virtual machines is considered and according to the quality of service criterion, the best location and routing are selected. Over time, by changing the software workload, to maintain the quality of service, replacement is performed according to the previous state of the related placement. In summary, in this study, to increase the quality of service, an optimal placement with the possibility of moving virtual machines that are already deployed in the network will be considered and implemented.

In general, the main contributions of the paper are as follows:

978-1-6654-8617-0/22/$31.00 ©2022 IEEE

- In this work, the placement of virtual machines related to a software system with the aim of service quality has been performed. This goal has received less attention in previous work on virtual machine placement. This type of placement has usually been done in the past regarding two general goals of (i) the best physical machine and (ii) the best traffic route. In this research, we have added the goal of service quality or customer satisfaction (the best response time) to past work.
- In this study, fluctuations in software workload at different times lead to the replacement of virtual machines belonging to a software. The way it works is that a new initial placement is deployed for the first time, but over time and respecting workload fluctuations, replacements will be performed with the migration of one or more virtual machines belonging to the related software. There is a balance between the cost of migration and the provision of the desired quality of service to the user due to workload fluctuations.

The rest of paper is organized as follows. Section 2 reviews the previous works in the field of VM placement. Sections 3 and 4 explain the proposed solution and analyze the findings of this paper, respectively. In Section 5, conclusions are provided.

2 Related Work

Over the past decade, a lot of research has been done on the placement of virtual machines. These researches are different in many aspects: 1) *Objective*: single objective (such as the most optimal place or the most optimal path) or multiple objectives (which means the combination of single objective) [12], [13], 2) *Static vs. dynamic*: static means that the placement for a given application does not change over time [10], [11], [24], [25], [27], 3) *Data center indicators*: such as energy consumption [23], [28], [29], security [21], [30], [31], traffic [16], [17], [20], [32], topology [10], service quality or SLA [13], [23], [24], [26], [34]. The placement issue raised in this paper is a combination of multi-objective with maintaining service quality. In the following, we will review the most important works done in this field.

Based on the analysis of dependencies between virtual machines, Narantova et al. [26] investigated the problem of placement in software-defined networking based infrastructure. This work states that exchange traffic between virtual machines has not been much studied. If dependent virtual machines are placed in different physical machines, resource consumption is greatly increased in the cloud infrastructure layers. Liu et al. [32] raised the issue of placement in a traffic-aware environment and guaranteed reliability. First, an appropriate reliability criterion for virtual machines is determined and virtual machines are classified into similar groups. Then, based on that, the appropriate physical machine is selected.

Wei et al. [14] considered the time interval between requests as a function of cost, and have shown how requests with different time intervals can affect performance and quality of service criteria. In [15], dynamic placement was proposed due to the fact that after a while servers are saturated with virtual machines and the load balance in the network is disturbed. To balance the cloud data center in each iteration of the placement, if the load is exceeded the tolerance threshold, some virtual machines are forced to migrate.

[16], [17] used a multi-objective function to place virtual machines. The algorithms used in [16] and [17] are genetic and greedy algorithms, respectively. Both have tried to better distribute the load in the data center with multi-objective functions. Goals such as reducing the number of active servers, centralizing network traffic, and equal productivity rates among active servers have resulted in better load distribution and acceptable performance in both energy consumption and efficient use of resources.

Braiki and Youssef [18] proposed a virtual machine placement based on particle swarm optimization. The proposed work was able to perform a balance between energy reduction and efficient use of resources, simultaneously. This research has shown that with its multi-objective function, it has been able to reduce the number of services and active physical servers, simultaneously. Fatima et al. [19] addressed customer and service provider satisfaction in the placement issue based on the Grey Wolf Optimizer algorithm. They focused on the productivity rate of physical machines and have shown that high productivity rates in each virtual machine lead to a higher level of service quality and load distribution in the data center.

The VM placement problem in [20] has been addressed with the awareness of traffic and communications in the infrastructure. The objective function in this research was divided into two parts: 1) the objectives of the cloud service provider and 2) the customer. The cloud service provider pays attention to the efficient use of data center and energy reduction, while the quality of cloud services and their acceptance levels are important for the customer. In this research, an attempt has been made to create a balance between these two goals. Alresheedi et al. [21] proposed several goals for the placement function, such as flexibility, security, support, and even better maintenance.

Mosa and Paton [23] presented virtual machine placement with the goal of reducing energy consumption and maintaining SLA. Unpredictable workload has also been considered in this work. In addition to the initial placement, VMs are dynamically replaced due to the load of physical machines. Calcavecchia et al. [22] considered the nature of dynamics and the constant change of workload to locate the virtual machine. In this work, the *Backward Speculative Placement* technique was introduced, which selects the appropriate destination physical machine according to the previous behavior of a virtual machine.

Compared to previous work, in this paper, our main focus is on maintaining the quality of service from the user's perspective. Of course, after the initial placement, the load from the users will change and we may need to reposition the virtual machines, but this is very costly for all services. Therefore, in each placement request, we try to provide a better location from a service quality perspective by migrating previous services and collocating with new needs.

3 Proposed Solution

In Formula 1, the placement objectives of the problem described are modeled, which includes six objectives: 1) *The most optimal path*: the first goal is to select paths that have more bandwidth. Therefore, the paths traversed ($Y_{i,j,p}$) for application p and the current bandwidth of the links ($Lbandwidth_{i,j}$) are considered, 2) *Nodes with high CPU number*: in this goal, nodes ($X_{p,vm,i}$) are selected for placement of vm belongs to application p that have a high number of CPUs, 3) *Nodes with high memory capacity*: it is the same as the previous goal, except that the high amount of memory in the nodes should be considered, 4)

TABLE 1: Software application placement input parameters.

Index	Parameter	Description
1	$N = \{1..n\}$	Set of data center nodes
2	$L = \{1..l\}$	Set of physical links in data center
3	$Lbandwidth_{i,j}$	Bandwidth between node *i* and node *j*
4	$Llatency_{i,j}$	Link latency between node *i* and node *j*
5	$Ncpu_i$	CPU count of node *i*
6	$Nmips_i$	CPU MIPS of node *i*
7	$Nmem_i$	Memory capacity of node *i*
8	$P = \{1..p\}$	Set of new software applications to be deployed
9	$P' = \{1..p'\}$	Set of current placed software applications in the data center
10	S_p	Source node of software application *p*, $S_p \in N$
11	D_p	Destination node of software application *p*, $D_p \in N$
12	$BDmnd_p$	Bandwidth demand of software application *p*
13	$R = \{1..Layer\}$	Number of layers of software application
14	$VMofP_{p,vm}$	Binary variable set to 1 if a *vm* belongs to software application *p*
15	$RTDmnd_p$	Maximum acceptable response time for software application *p*
16	$CPUDmnd_{p,vm}$	CPU demand of a *vm* in application *p*
17	$MIPSDmnd_{p,vm}$	MIPS demand of a *vm* in application *p*
18	$MEMDmnd_{p,vm}$	Memory demand of a *vm* in application *p*
19	$X'_{p',vm,i}$	Set to 1 if a *vm* belongs to software application p' has been placed in node *i* in previous placement

TABLE 2: Software application placement output parameters.

Index	Parameter	Description
1	$X_{p,vm,i}$	Binary variable set to 1 if a *vm* of software application is placed at node i, otherwise set to 0
2	$Y_{i,j,p}$	Binary variable set to 1 if link between *i* and *j* is selected for software application request *p*, otherwise set to 0
3	$MGRT_{p',vm,i}$	Binary variable set to 1 if node *i* is selected as migration destination of a *vm* belongs to software application p', otherwise set to 0

High processing power nodes: in reducing the response time of the user request, nodes with high MIPS power should be considered, 5) *Low latency paths*: reducing the user response time is strongly dependent on the selection of low-delay edges, and 6) *Reduction the number of migrations*: the migration of previous virtual machines to new locations is of great importance for optimizing the current placement. This is not only to improve the quality of service to deal with the new workloads on the previously placed services, but also to perform collocation operation between different requests over time and eliminate the need for new virtual machines creation. It should be noted that migration has its own costs. In this goal, if migrations has a favorable profit, it is considered.

$$
\begin{aligned}
Min \quad & \sum_{i\in N}\sum_{j\in N}\sum_{p\in P}\frac{1}{Lbandwidth_{i,j}} \times BDmnd_p \times Y_{i,j,p} \\
& + \sum_{i\in N}\sum_{vm\in R}\sum_{p\in P}\frac{1}{Ncpu_i} \times CPUDmnd_{p,vm} \times X_{p,vm,i} \\
& + \sum_{i\in N}\sum_{vm\in R}\sum_{p\in P}\frac{1}{Nmem_i} \times MEMDmnd_{p,vm} \times X_{p,vm,i} \\
& + \sum_{i\in N}\sum_{vm\in R}\sum_{p\in P}\frac{1}{Nmips_i} \times MIPSDmnd_{p,vm} \times X_{p,vm,i} \\
& + \sum_{i\in N}\sum_{j\in N}\sum_{p\in P} Llatency_{i,j} \times Y_{i,j,p} \\
& + \sum_{i\in N}\sum_{vm\in R}\sum_{p'\in P'} MGRT_{p',vm,i} \\
& - \sum_{i\in N}\sum_{vm\in R}\sum_{p'\in P'} X'_{p',vm,i} \times X_{p',vm,i}
\end{aligned}
\tag{1}
$$

In the following, we examine the constraints related to the above goals. Path continuity is important in placement for any application. In order to maintain the continuity of the request, the sum of the input ($\sum_{j\in N} Y_{j,i,p}$) and output ($\sum_{z\in N} Y_{i,z,p}$) edges must be equal in each node (i). Nodes of origin and destination are excluded.

$$
\begin{aligned}
\forall p \in P, i \in N : & \sum_{j\in N} Y_{j,i,p} + (1 \; if \; i = S_p) \\
& = \sum_{z\in N} Y_{i,z,p} + (1 \; if \; i = D_p)
\end{aligned}
\tag{2}
$$

The next two constraints are to prevent the cycle from the origin of a request p to its destination.

$$\forall p \in P, j \in N : \sum_{i\in N} Y_{i,j,p} \leq 1 \tag{3}$$

$$\forall p \in P, i \in N : \sum_{j\in N} Y_{i,j,p} \leq 1 \tag{4}$$

The next constraint verifies that the total bandwidth requested ($\sum_{p\in P} BDmnd_p$) by all applications traveling through a link ($Y_{i,j,p}$) does not exceed its current capacity ($Lbandwidth_{i,j}$).

$$\forall i, j \in N : \sum_{p\in P} BDmnd_p \times Y_{i,j,p} \leq Lbandwidth_{i,j} \tag{5}$$

The next constraint controls the exact number of virtual machines requested for each application.

$$\forall p \in P, vm \in R : \sum_{i\in N} X_{p,vm,i} = VMofP_{p,vm} \tag{6}$$

The next constraint ensures that each virtual machine is in the selected path of origin to destination. In other words, vm belongs to the application p, is placed in node i, if node i is a member of one of the path nodes.

$$\forall p \in P, vm \in R, i \in N : \sum_{j\in N} Y_{i,j,p} + \sum_{z\in N} Y_{z,i,p} \geq X_{p,vm,i} \tag{7}$$

The next two constraints check the node capacity in terms of the number of processors and the amount of memory, and ensure

that the sum of the requested resources of all the virtual machines in the selected node does not exceed its resources.

$$\forall i \in N : \sum_{p \in P} \sum_{vm \in R} X_{p,vm,i} \times CPUDmnd_{p,vm} \leq Ncpu_i \tag{8}$$

$$\forall i \in N : \sum_{p \in P} \sum_{vm \in R} X_{p,vm,i} \times MEMDmnd_{p,vm} \leq Nmem_i \tag{9}$$

Finally, the last constraint avoids placing the virtual machine on the switch nodes. The valid area would be S.

$$\forall p \in P, vm \in R : \sum_{i \in N, i \notin S} X_{p,vm,i} = 0 \tag{10}$$

4 Experimental Results

4.1 Setup

The fat-tree network topology (k=4) was selected to perform the simulation. Each of the nodes in this network has between 12 to 96 processing cores and memory volume between 12GB and 96GB. The processing power of the nodes was also considered between 15MIPS and 90MIPS. The latency between the links varies between 6 and 30 milliseconds and the bandwidth of the links varies between 50Gbps and 100Gbps. All numbers are generated randomly between the specified ranges. 30 three-layer software were considered for placement. Each software needs three virtual machines and each of the them requires different resources to perform related services.

The resources required for virtual machines are generated randomly between 1GB and 8GB for processing, between 2GB and 8GB for memory, between 1Gbps and 12Gbps for bandwidth, and between 0.2 and 1.4 for MIPS. On the other hand, the maximum acceptable response time for applications is considered between 100 and 200 milliseconds. GLPK LP/MIP Solver v4.65 was used to simulate and solve the problem. We have executed our simulation on a machine with an Intel Core ™ i7-2670QM processor and 6GB of memory.

In the next section, three different algorithms have been implemented to compare the innovations of this paper, which are described below:

1) *PL*: It is a placement algorithm without considering the criteria of service quality and it is not sensitive to service workload fluctuations. The objective function of this algorithm includes two general objectives: 1) the best node(s) that has more CPU and memory, and 2) the best path(s) that has the most available bandwidth.
2) *PLQoS-UM*: It is a placement algorithm with service quality considerations that, in addition to having the objectives of the *PL* algorithm, also takes into account quality considerations. In other words, in case of service workload fluctuation, it begins to examine the migration of virtual machines to have a better service quality.
3) *PLQoS-LM*: It is the same as the *PLQoS-UM* algorithm, except that it controls the migration cost of virtual machines.

We finally compared the simulation results of these three algorithms from two aspects: 1) the number of service quality violations and 2) the number of virtual machine migration. In the next section, we will analyze the simulation results.

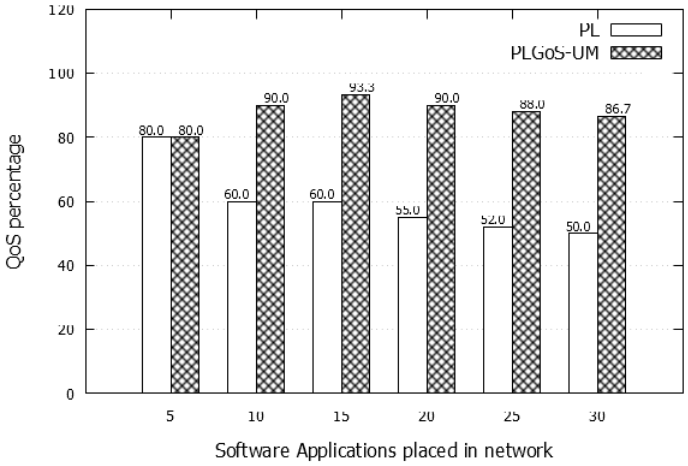

Fig. 1: The effect of increasing the number of application placements on the quality of user service

4.2 Result

In the first simulation, it is assumed that we first have a data center, in which there is no running virtual machine on it. In each placement, five applications are placed in the network, so that each application requires three virtual machines to perform the related services. The number of placements is repeated up to six times (see Figure 1).

The service quality evaluation criterion is considered as follows: *If the total execution time of a request in virtual machines plus their total communication delay is less than the maximum response time, the service will be in accordance with the service quality criterion, otherwise, it has been violated*. The execution time in a virtual machine to perform the relevant task is also calculated by dividing the required MIPS to the existing MIPS in the node in which it is placed. The simulation results were performed for two different algorithms: *PL* and *PLQoS-UM*. As can be seen in Figure 1, in the last placement, we will have a network with 30 three-layer applications, or 90 virtual machines. If we do not consider the service quality goals in the placement, only 50% of the services will meet the service quality criteria, and according to *PLQoS-UM* algorithm, if we apply the service quality goals, we will have a more effective placement. That is, 86.7% of the services will meet the service quality criteria and response time will be in an acceptable time.

In the second simulation, we answer the question of how much the quality of service will deteriorate if in *PL* algorithm, after initial placement, the workload is increased by 50%. We are also looking for an answer to the question that if in any placement the strategy is changed and *PLQoS-UM* algorithm is applied instead of *PL*, how much the service quality will be improved.

To simulate the increase in workload, the processing load (MIPS only) of virtual machines is added by 50%. As can be seen in Figure 2, if we compare the results of service quality for *PL* algorithm in two modes, initial placement and then 50% increase in workload, there is a very sharp drop in service quality. As can be seen, after the placement of 30 applications, there is a 30% reduction in service quality and a total of 20% service

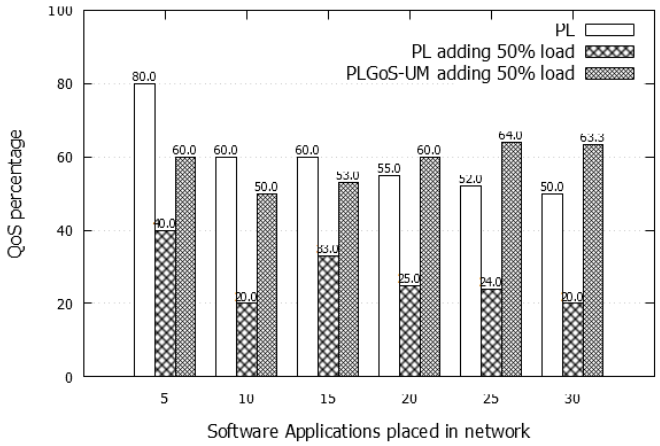

Fig. 2: Comparison of service quality sensitive and non-service quality sensitive placements due to 50% increase in workload

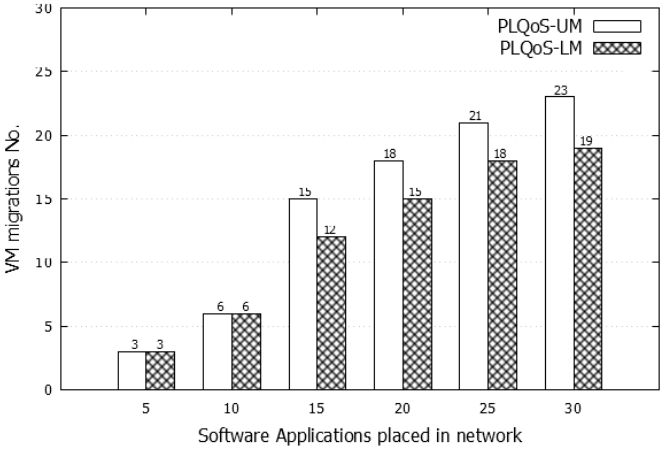

Fig. 3: Comparison of the number of virtual machine migration to improve the quality of service in the service quality sensitive placement algorithm in both unlimited and limited migration modes

quality for all applications is being met. If we change our decision in any placement and execute *PLQoS-UM* algorithm instead of *PL* algorithm, the quality of service is improved significantly. If we consider at the placement of 20 applications in the figure, we see that although the workload has increased by 50% and even *PLQoS-UM* algorithm was not executed from the beginning, the quality of service is improved by 35%.

In this study, we have implemented two different algorithms for the purpose of service quality, the difference between these two algorithms is in controlling or unlimited migration of virtual machines. Our simulation results show that their output does not differ much in terms of service quality criteria, so controlling the number of migrations will be in our favor. To control the number of migrations, we have implemented it as a goal in our optimization function, which force the placement solver to choose a solution that has the lowest migration cost. As shown in Figure 3, in the first placement, which is for five applications and leads to the creation of 15 virtual machines, if the workload is increased by 50% and replacement by these two algorithms is performed (*PLQoS-UM* and *PLQoS-LM*), three virtual machines out of 15 migrates to new places. In other cases, we had almost three fewer migrations. As

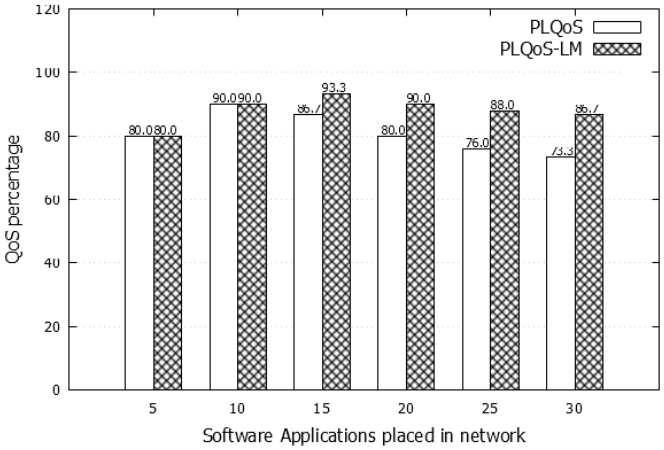

Fig. 4: Measuring the quality of service for placement sensitive to service quality in two modes of not relocating previously located services in the new placement and the possibility of their limited migration

can be seen, the number of migrations in the 90 virtual machines, which is for the last case, is decreased from 23 to 19, which means a 17% reduction in the number of migrations in *PLQoS-LM* algorithm.

In the third simulation, our goal is to measure the quality of service for placement sensitive to service quality in the two modes of not relocating previously placed services in the new request (*PLQoS*) and the possibility of their limited migration *PLQoS-LM*. The way it works is that the placement is performed for five applications in each step, after the placement we start to change the workload of the applications by 50%. Then, we start measuring the quality of service to know how much the change in workload reduces the quality of service. As can be seen in Figure 4, with the increase in the number of applications, traffic and processing in nodes, the VMs needs to be relocated. As seen, the difference in service quality from the placement of 20 applications onwards is significant and is more than 10%. These results can help cloud providers to make appropriate decisions for maintaining SLA.

5 Conclusion

The issue of placement of virtual machines has been one of the most important researches in the last decade, which is performed for various goals. Examples of these goals include reducing energy consumption in the data center, balancing resources on servers, maintaining SLAs, establishing security, and so on. One of the goals that has received less attention is to improve the quality of service for users in case of workload fluctuations. In this paper, we present a new placement approach considering service quality considerations. The results show that considering the criteria for optimizing the quality of service in dynamic placement can properly manage the unforeseen workload by users. One of the concerns in this type of placement is the high number of migration of virtual machines. We addressed this problem by controlling the number of migrations in the placement objective function. Using the solution presented in this paper, cloud service providers can significantly reduce the payment of service level violations.

REFERENCES

[1] Gupta, R. K., & Pateriya, R. K. (2014). Survey on virtual machine placement techniques in cloud computing environment. International Journal on Cloud Computing: Services and Architecture (IJCCSA), 4(4), 1-7.

[2] Ihara, D., López-Pires, F., & Barán, B. (2015). Many-objective virtual machine placement for dynamic environments. In 2015 IEEE/ACM 8th International Conference on Utility and Cloud Computing (UCC) (pp. 75-79). IEEE.

[3] Padhy, S., & Chou, J. (2021). Reconfiguration Aware Orchestration for Network Function Virtualization With Time-Varied Workload in Virtualized Datacenters. IEEE Access, 9, 48413-48428.

[4] Gohil, B., Shah, S., Golechha, Y., & Patel, D. (2016). A comparative analysis of virtual machine placement techniques in the cloud environment. International Journal of Computer Applications, 156(14).

[5] Shigeta, S., Yamashima, H., Doi, T., Kawai, T., & Fukui, K. (2012). Design and implementation of a multi-objective optimization mechanism for virtual machine placement in cloud computing data center. In International conference on cloud computing (pp. 21-31). Springer, Cham.

[6] Ghasemi, A., & Haghighat, A. T. (2020). A multi-objective load balancing algorithm for virtual machine placement in cloud data centers based on machine learning. Computing, 102(9), 2049-2072.

[7] Masdari, M., Nabavi, S. S., & Ahmadi, V. (2016). An overview of virtual machine placement schemes in cloud computing. Journal of Network and Computer Applications, 66, 106-127.

[8] Kim, S., & Choi, Y. R. (2020). Constraint-aware VM placement in heterogeneous computing clusters. Cluster Computing, 23(1), 71-85.

[9] Zheng, Q., Li, R., Li, X., & Wu, J. (2015). A multi-objective biogeography-based optimization for virtual machine placement. In 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (pp. 687-696). IEEE.

[10] Tighe, M., & Bauer, M. (2017). Topology and application aware dynamic vm management in the cloud. Journal of Grid Computing, 15(2), 273-294.

[11] Li, B., Cheng, B., Liu, X., Wang, M., Yue, Y., & Chen, J. (2021). Joint Resource Optimization and Delay-aware Virtual Network Function Migration in Data Center Networks. IEEE Transactions on Network and Service Management, 18(3), 2960-2974.

[12] Pires, F. L., & Barán, B. (2015). A virtual machine placement taxonomy. In 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (pp. 159-168). IEEE.

[13] López-Pires, F., & Barán, B. (2017). Many-objective virtual machine placement. Journal of Grid Computing, 15(2), 161-176.

[14] Wei, W., Wei, X., Chen, T., Gao, X., & Chen, G. (2013). Dynamic correlative VM placement for quality-assured cloud service. In 2013 IEEE International Conference on Communications (ICC) (pp. 2573-2577). IEEE.

[15] Patel, K. K., Desai, M. R., & Soni, D. R. (2017). Dynamic priority based load balancing technique for VM placement in cloud computing. In 2017 International Conference on Computing Methodologies and Communication (ICCMC) (pp. 78-83). IEEE.

[16] Riahi, M., & Krichen, S. (2018). A multi-objective decision support framework for virtual machine placement in cloud data centers: A real case study. The Journal of Supercomputing, 74(7), 2984-3015.

[17] Qin, Y., Wang, H., Yi, S., Li, X., & Zhai, L. (2020). Virtual machine placement based on multi-objective reinforcement learning. Applied Intelligence, 50(8), 2370-2383.

[18] Braiki, K., & Youssef, H. (2018). Multi-objective virtual machine placement algorithm based on particle swarm optimization. In 2018 14th International Wireless Communications & Mobile Computing Conference (IWCMC) (pp. 279-284). IEEE.

[19] Fatima, A., Javaid, N., Anjum Butt, A., Sultana, T., Hussain, W., Bilal, M., & Ilahi, M. (2019). An enhanced multi-objective gray wolf optimization for virtual machine placement in cloud data centers. Electronics, 8(2), 218.

[20] Farzai, S., Shirvani, M. H., & Rabbani, M. (2020). Multi-objective communication-aware optimization for virtual machine placement in cloud datacenters. Sustainable Computing: Informatics and Systems, 28, 100374.

[21] Alresheedi, S. S., Lu, S., Abd Elaziz, M., & Ewees, A. A. (2019). Improved multiobjective salp swarm optimization for virtual machine placement in cloud computing. Human-centric Computing and Information Sciences, 9(1), 1-24.

[22] Calcavecchia, N. M., Biran, O., Hadad, E., & Moatti, Y. (2012). VM placement strategies for cloud scenarios. In 2012 IEEE Fifth International Conference on Cloud Computing (pp. 852-859). IEEE.

[23] Mosa, A., & Paton, N. W. (2016). Optimizing virtual machine placement for energy and SLA in clouds using utility functions. Journal of Cloud Computing, 5(1), 1-17.

[24] Prodan, R., Torre, E., Durillo, J. J., Aujla, G. S., Kummar, N., Fard, H. M., & Benedikt, S. (2019). Dynamic multi-objective virtual machine placement in cloud data centers. In 2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (pp. 92-99). IEEE.

[25] Roy, P., Tahsin, A., Sarker, S., Adhikary, T., Razzaque, M. A., & Hassan, M. M. (2020). User mobility and quality-of-experience aware placement of virtual network functions in 5g. Computer Communications, 150, 367-377.

[26] Wang, S. H., Huang, P. P. W., Wen, C. H. P., & Wang, L. C. (2014). EQVMP: Energy-efficient and QoS-aware virtual machine placement for software defined datacenter networks. In The International Conference on Information Networking 2014 (ICOIN2014) (pp. 220-225). IEEE.

[27] Zheng, X., & Cai, Y. (2014). Dynamic virtual machine placement for cloud computing environments. In 2014 43rd International Conference on Parallel Processing Workshops (pp. 121-128). IEEE.

[28] Choudhary, A., Rana, S., & Matahai, K. J. (2016). A critical analysis of energy efficient virtual machine placement techniques and its optimization in a cloud computing environment. Procedia Computer Science, 78, 132-138.

[29] Beloglazov, A., Buyya, R., Lee, Y. C., & Zomaya, A. (2011). A taxonomy and survey of energy-efficient data centers and cloud computing systems. Advances in computers, 82, 47-111.

[30] Demirci, S., Demirci, M., & Sagiroglu, S. (2019). Virtual security functions and their placement in software defined networks: A survey. Gazi University Journal of Science, 32(3), 833-851.

[31] Moradi, N., Shameli-Sendi, A., & Khajouei, A. (2021). A Scalable Stateful Approach for Virtual Security Functions Orchestration. IEEE Transactions on Parallel and Distributed Systems, 32(6), 1383-1394.

[32] Liu, X., Cheng, B., Yue, Y., Wang, M., Li, B., & Chen, J. (2019). Traffic-aware and reliability-guaranteed virtual machine placement optimization in cloud datacenters. In 2019 IEEE 12th International Conference on Cloud Computing (CLOUD) (pp. 91-98). IEEE.

[33] Lu, K., Yahyapour, R., Wieder, P., Kotsokalis, C., Yaqub, E., & Jehangiri, A. I. (2013). Qos-aware vm placement in multi-domain service level agreements scenarios. In 2013 IEEE Sixth International Conference on Cloud Computing (pp. 661-668). IEEE.

[34] Lu, K., Yahyapour, R., Wieder, P., Kotsokalis, C., Yaqub, E., & Jehangiri, A. I. (2013). Qos-aware vm placement in multi-domain service level agreements scenarios. In 2013 IEEE Sixth International Conference on Cloud Computing (pp. 661-668). IEEE.

[35] Narantuya, J., Ha, T., Bae, J., & Lim, H. (2019). Dependency Analysis based Approach for Virtual Machine Placement in Software-Defined Data Center. Applied Sciences, 9(16), 3223.

[36] Nazir, B. (2018). QoS-aware VM placement and migration for hybrid cloud infrastructure. The Journal of Supercomputing, 74(9), 4623-4646.

[37] Hong, H. J., Chen, D. Y., Huang, C. Y., Chen, K. T., & Hsu, C. H. (2013). QoE-aware virtual machine placement for cloud games. In 2013 12th Annual Workshop on Network and Systems Support for Games (NetGames) (pp. 1-2). IEEE.

[38] Upadhyaya, B., Zou, Y., Keivanloo, I., & Ng, J. (2014). Quality of experience: What end-users say about web services?. In 2014 IEEE International Conference on Web Services (pp. 57-64). IEEE.

[39] Mann, V., Vishnoi, A., Iyer, A., & Bhattacharya, P. (2012). Vmpatrol: Dynamic and automated qos for virtual machine migrations. In 2012 8th international conference on network and service management (cnsm) and 2012 workshop on systems virtualiztion management (svm) (pp. 174-178). IEEE.

[40] Satpathy, A., Addya, S. K., Turuk, A. K., Majhi, B., & Sahoo, G. (2018). Crow search based virtual machine placement strategy in cloud data centers with live migration. Computers & Electrical Engineering, 69, 334-350.

[41] Ahmad, R. W., Gani, A., Hamid, S. H. A., Shiraz, M., Yousafzai, A., & Xia, F. (2015). A survey on virtual machine migration and server consolidation frameworks for cloud data centers. Journal of network and computer applications, 52, 11-25.

[42] Sharma, S., & Chawla, M. (2013). A technical review for efficient virtual machine migration. In 2013 International Conference on Cloud & Ubiquitous Computing & Emerging Technologies (pp. 20-25). IEEE.

View publication stats