Hindawi
Mobile Information Systems
Volume 2021, Article ID 3601104, 7 pages
https://doi.org/10.1155/2021/3601104

Research Article

Software-Defined Network Resource Optimization of the Data Center Based on P4 Programming Language

Fei Peng [1] and Tianjie Cao [2]

[1] *Yangzhou Polytechnic College, Yangzhou 225000, China*
[2] *China University of Mining and Technology, Xuzhou, China*

Correspondence should be addressed to Fei Peng; pengfei@yzpc.edu.cn

Received 15 June 2021; Revised 15 July 2021; Accepted 29 July 2021; Published 4 August 2021

Academic Editor: Fazlullah Khan

This paper makes use of the new architecture software-defined network (SDN) in the cloud data center based on P4 language to realize the flexible management and configuration of the network equipment to achieve (a) data center virtualization management and (b) data center resource optimization based on the P4 programming language. Furthermore, error tolerance of dynamic network optimization depends on the virtual machine (VM) online migration technology, and the load balancing mechanism has a very good flexibility. At the same time, the paper proposed a multipath VM migration strategy based on a quality of service (QoS) mechanism, which divides the VM migration resources into different QoS flows by network dynamic transmission and then selects valid forwarding for each flow path to migrate VMs. This ensures to improve the overall migration performance of VMs and ultimately the dynamic optimization of the network resources and their management. Our experimental evaluations show that the proposed model is approximately 13% and 17% better than the traditional state-of-the-art methods in terms of minimum migration time and the least downtime, respectively.

1. Introduction

At present, dynamic network optimization depends on the dynamic migration technology of virtual machines (VMs) to complete the core components of virtualization, which brings great convenience for the management and maintenance in the data center community [1, 2]. At the same time, data centers have good flexibility in resource optimization, load balancing, and error tolerance [3]. Although network optimization relies on the migration of virtual machines which has several good advantages due to the migration process, there are still certain unavoidable performance degradation issues during the migration process, especially in a cross-data center or geographically distributed data centers migration environments [4, 5]. The characteristics of low bandwidth and high latency in the wide-area network greatly reduce the performance of virtual machine online migration. Many scholars have done a lot of research work on virtual machine migration under dynamic network optimization across data centers. For example, in [6] the authors have analyzed the challenge of dynamic transmission of virtual machines in WAN. Similarly, authors in [7, 8] have used redirection and dynamic DNS to reconfigure the data center network, dynamically. In [9], the authors have used VPN services to realize the migration of network transmission virtual machine in a WAN based "CloudNet" data center platform. In [10–12], the authors have proposed several simple mathematical models to calculate the downtime of network resources during transferring memory pages in migrations.

Through related research, we know that network problems bring great challenges to the dynamic migration of virtual machines in WAN [13–15]. Most of the previous researchers started from the virtual machine migration process of network transmission and did not consider the network resources. We know from the whole network that the dynamic network transmission time is determined by a large amount of migrated data and the assigned or available transmission bandwidth. Therefore, in the process of virtual machine migration, effective allocation of network resources is an important way to improve network transmission performance and it is relatively unexplored in the literature [16–18].

In this paper, we introduce a SDN framework, which is based on the P4 programming language for the network across data centers. Moreover, a technique is presented for the dynamic optimization of the virtual machine migration environment while taking advantage of the flexibility of the SDN architecture in network updates. We propose a multipath parallel transmission strategy (known as QMA) based on the QoS flows. At the same time, we adopted dynamic routing algorithms to process different QoS flows, which can provide good network transmission resources for virtual machine migration. This method can effectively reduce the virtual machine migration time and improve the performance of the network dynamic adjustment. Following are the major contributions of our work:

(1) We introduce a P4 programming language-based SDN framework for the network across the data center to dynamically optimize the virtual machine migration environment

(2) A migration technique is suggested that takes the full advantage of the flexibility of the SDN architecture in network updates

(3) We propose a multipath parallel transmission strategy (known as QMA) which is based on the QoS flows

(4) Finally, we adopt a dynamic routing algorithm to process different QoS flows, which can provide good network transmission resources for the virtual machine migration process

The structure of the rest of the paper is organized as follows. In Section 2, we discuss some essential background. Section 3 introduces the proposed model. Section 4 describes our evaluation and experimental results in detail. Finally, Section 5 concludes the paper along with future direction for further investigation.

2. Network Architecture and Online Migration Model

In this section, we discuss the system architecture followed by the migration model.

2.1. System Network Architecture. We designed the SDN-based cloud data center network architecture as shown in Figure 1. We give a global data center system view for easy understandability. In this network architecture, the network controller controls all network resources, while some cloud management systems such as OpenStack manage all virtual machine computing resources in the network. In this paper, we deploy the QoS routing mechanism in the network controller, which can communicate with the cloud management system to trigger appropriate decisions.

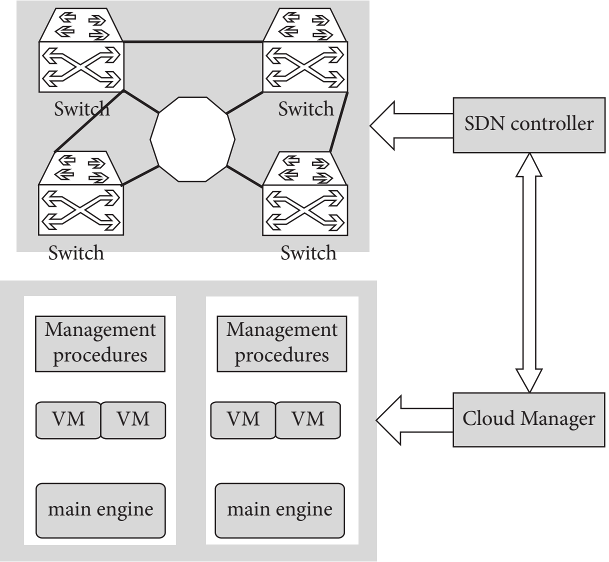

Figure 1: SDN-based network transmission in the virtual machine migration framework.

Indeed, certain devices in the network, such as switches and/or routers, can implement the data forwarding mechanism according to the table data that comprises forwarding flow details obtained from time to time. The SDN controller uses the standard OpenFlow protocol to communicate with these network devices, to obtain link-state information. At the same time, the SDN controllers are responsible for calculating the forwarding flow tables of all switching equipment [19]. Besides, the cloud controller of the OpenStack is responsible for managing the computing and storage resources of all data centers. The cloud controller is also responsible for recording necessary information about virtual machines and physical hosts in the network, such as virtual machine memory size and remaining CPU resources of the physical hosts. All computing nodes will periodically report their updated information to the cloud manager [11]. Besides, OpenStack also provides a unified resource management function. We discuss the core principles and benefits of an SDN-based network architecture, as well as issues to consider while implementing it and a comparison of the features of major SDN controllers. We go over its essential features, architecture, and modules, as well as the key technologies and programming languages that will be used to implement it. Furthermore, SDN includes a variety of protocols for communication between the controller and network devices, which takes place via the southbound interface [20, 21].

In this paper, we use the cloud manager to manage the virtual machines in the network transmission. When one of the virtual machines needs to be migrated (based on some predefined threshold values, underload, overload situation, etc.), the cloud manager sends a network cooperation request to the SDN controller, and then the SDN controller implements network control according to the current network conditions [22]. Next, we send the corresponding network forwarding strategy to the network devices in the data forwarding layer, and the cloud manager migrates the virtual machine or a set of VMs from certain hosts to other hosts according to the forwarding strategy. Note that the forwarding strategy is based on a multipath transmission mechanism and is described later in Section 3.

2.2. VM Migration Model in Network Transmission. Precopy migration is the most common way, which has been implemented in the latest virtualization tools, such as VMware, XEN, and KVM [22]. This paper adopts the precopy method, and the details are given as follows. As shown in Figure 2, there are three parts in the online migration of virtual machines, mainly including the mirror copy stage, the iterative precopy stage, and the downtime copy stage. The virtual machine content (such as mirror files) is first copied from the source physical host to the destination host, which is the mirror copy phase in the diagram. Next, we enter the iterative precopy stage of the memory. In the first iteration of the precopy, all memory pages will be copied to the destination virtual machine or host [22]. Because dirty pages will be generated in the transmission process, the next iteration will copy the changed dirty pages of memory to the target virtual machine or host. The dirty pages of memory are copied continuously until the dirty pages become relatively very few or reach a certain value, which marks the beginning of the next stage, that is, the shutdown copy stage. Subsequently, the rest of the dirty pages are copied to the destination virtual machine or physical host; after that, the virtual machine on the destination host takes over the work of the source virtual machine and continues to provide services. At the same time, the virtual machine on source host will be destroyed and terminated. The VM in the last phase is unavailable to offer services and the time it takes in switching the services is known as the downtime [22]. The overall time is called the migration time.

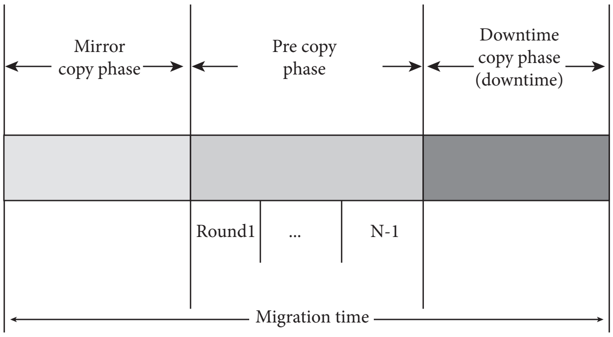

Figure 2: Online migration policy (the precopy approach).

3. Multipath Transmission Mechanism Based on QoS Flow

Migration time is an important performance indicator of the network transmission. Therefore, how to improve network transmission performance and reduce the overall migration time of virtual machines is a very important research issue. Considering the entire network, when the network is transmitting, the allocated network resources (such as transmission bandwidth) directly determine the virtual machine migration time, especially in the WAN environment. Therefore, this paper proposes a virtual machine migration mechanism (QMA) based on QoS flow multipath to solve the problem of network dynamic optimization and resource management.

3.1. Design Principle. According to the virtual machine online migration model in the network transmission, we divide the virtual machine online migration process into three consecutive stages: (i) mirror migration, (ii) memory precopy, and (iii) downtime copy, and the execution sequence is carried out in sequence. Because mirroring, memory, and CPU have different requirements for network bandwidth and delay, we adopt a streaming method to set mirror files, memory precopy, and memory shutdown copy as different levels of QoS flows. Note that different levels of migration flow meet different QoS demands—we set QoS to mainly meet the network delay and migration bandwidth requirements. Besides, the same level of flow allows multipath parallel migration during the network dynamic transmission of virtual machine migration. In SDN architecture, OpenFlow allows us to define different types of flows, such as the QoS level. Moreover, the SDN architecture has great flexibility for network configuration. The network controller can formulate routing strategies that meet QoS requirements according to different QoS flows and then send the routing strategies to the forwarding device in the form of flow tables. Details in the tables can be updated periodically with passage of time according to the current network status. Updated detail can realize a dynamic routing mechanism, realize efficient data transmission, and complete VM migration process stream transmission.

3.2. Routing Optimization Model. Different routing strategies are established according to different QoS flows; therefore, we establish a dynamic routing model. The description of the proposed routing model is as follows: we regard the network as an undirected graph $G(N, L)$, where N represents the set of network nodes and L represents the set of links between various nodes. Furthermore, we define the QoS flow request as a four-tuple (s, d, B, D), where s represents population routing, d represents egress routing, B represents the minimum bandwidth requested, and D represents the delay upper bound requirement. In the same way, P represents the entry and exit node pairs in all QoS requests and $R(s, d)$ represents the set of all routes from the entry node s to the exit node d $(R \subset L)$. In this article, we regard dynamic routing as a constrained shortest path problem. The problem has two attributes, including a cost function and several constraint conditions. These two attributes can reflect the network conditions and QoS requirements. Also, we use the link bandwidth to calculate the weight of each link as the cost measure. Moreover, we use the link bandwidth to calculate the weight of each link as the cost measure and take the end-to-end maximum delay as the constraint condition. This is due to the fact that the migration time is an important performance evaluation factor for virtual machine migration in dynamic network transmission scenarios.

The cost function is given by

$$f_c(r) = \sum_{\forall (i,j)\in r} c_{ij}, \tag{1}$$

where c_{ij} represents the weight of the link. In this paper, the remaining bandwidth rb_{ij} of the link is taken as the dynamic information available for the QoS routing, and the maximum flow of the network is considered. Therefore, we redefine the weight of the link as given by

$$c_{ij} = \frac{1 + \sum_{\forall (s,d)\in P} \chi_{\mathrm{CM}(s,d)}(x, j)}{rb_{ij}}, \tag{2}$$

where CM (s, d) is the set of minimum cut links between the two nodes, i.e., s and d. From graph theory, we know that a minimum cut set of (s, d) can affect the maximum flow between (s, d). If the link $(x, j) \in \mathrm{CM}(s, d)$, then $\chi_{\mathrm{CM}(s,d)}(x, j) = 1$. In addition, we set the link weight to be inversely proportional to the remaining bandwidth of the link. In this way, links with less remaining bandwidth have relatively fewer opportunities to participate in the new routing than the links with more remaining bandwidth. At the same time, to avoid the zero-weight assignment of links on the route by those nodes that do not belong to the in and out nodes, a constant value of 1 is added to the numerator of equation (2).

To satisfy the maximum flow and add fewer links to the routing path, C (R) should be as small as possible. In this paper, we define the end-to-end delay between the entry-exit nodes as constraint conditions, as given by

$$f_D(r) = \sum_{\forall (i,j)\in r} \left(\frac{M}{R} + \frac{M_{ij}}{B_{ij}} + pd_{ij}\right) c_{ij} < D_{\max}. \tag{3}$$

In this paper, we assume that the end-to-end delay mainly includes the transmission delay, the queue delay, and the propagation delay. In equation (3), M is the maximum packet length, R is the minimum bandwidth allocated by the selected link, M_{ij} is the maximum length of a data packet passing through the link (i, j), B_{ij} is the bandwidth of the link (i, j), and B_{ij} is the propagation delay of the (i, i) link. Based on the above mathematical model, the QoS routing problem can be described as given by

$$r^* = \arg\min\{f_C(r) | r \in R(s, d) \subseteq L\}. \tag{4}$$

When the delay condition is met, we find the path that meets the condition to minimize the cost function, and this path is the optimal path that satisfies the QoS constraints. Because the cost functions are an inverse function of the remaining bandwidth, the smaller the cost function, the more the remaining bandwidth that can be allocated.

3.3. Algorithm Implementation. According to the design principles as described above, this article assumes five tuples $(s, d, S_{\text{disk}}, S_{\text{pre-mean}}, S_{\text{stop-mem}})$ to represent the network dynamic transmission virtual machine migration request from a particular node s to node d. According to the routing optimization model given above, the network dynamic transmission virtual machine migration resource is divided into three parts, and each part sets the corresponding QoS request. When a virtual machine with network transmission is being migrated, the cloud manager will establish a QoS flow request priority queue for each virtual machine migration request. Next, the cloud manager will send the request queue information to the cloud controller, as shown in Figure 1, and the controller will perform each flow according to the QoS queue priority process. Flows of the same level will be processed in sequence according to the set migration order. Moreover, different routing and forwarding strategies will be determined, and then the flow table will be issued. Finally, the data transmission will be completed. The algorithm steps are as given in Algorithm 1.

The algorithm has a total of seven steps. The first two steps are the process of creating QoS flow queues, and the latter steps are mainly the flow processing process, which is the solution to the dynamic routing problem. Note that this article uses an improved Lagrangian relaxation algorithm, based on the well-known Dijkstra algorithm, to obtain a more optimized and the shortest migration path that improves the dynamic transmission performance of the entire network and, therefore, performance of the migration procedure [11, 14].

4. Experiments and Analysis

In this section, we evaluate the performance of the proposed algorithm in a simulated environment.

4.1. Experimental Setup. To verify the performance of the proposed strategy, this paper adopts Expand ANSNET network topology for simulation verification. The topology consists of 18 nodes and 30 links. All links are bidirectional, and there are a total of three-person-one-out node pairs in this topology, namely, (1, 17), (2, 26), and (4, 15). The propagation delay of each link is assumed to be evenly distributed within a certain range. The simulation parameter settings are shown in Table 1.

4.2. Result Analysis. This paper mainly compares and analyzes two aspects/metrics of the migration performance: (i) overall migration time and (ii) migration downtime. The simulation selects the benchmark algorithm to compare the performance with the QoS-based multistream transmission algorithm proposed in this paper. The benchmarked algorithm is based on the traditional network architecture. When there is a virtual machine migration request, it will use the traditional routing mechanism to select the transmission path from the source to the destination and then use the virtualization migration tool to realize the migration of the virtual machine on the network [21]. Note that multistream parallel transmission is not supported under the proposed architecture and is not within the scope of our current work.

Step 1. Receive a migration request (5-tuple).
Step 2. Establish a QoS flow migration queue. According to the online migration model, the three streams of mirror file, memory copy, and shutdown copy are added to the migration queue.
Step 3. Process each flow in the migration queue. First, calculate the weights of all links in the current network by formula (2).
Step 4. Use the Dijkstra algorithm to calculate the minimum cost path r_c, according to formula (4), if $f_D(r) < D\max$, then tC is the optimal path, which is used as the path to transmit the current stream; otherwise, jump to step 5.
Step 5. Use the Dijkstra algorithm to calculate the minimum delay path r_d from s to d. If r_d does not exist, use r_c as an alternative path. If there is, proceed to step 6.
Step 6. Let
$\lambda = ((f_c(r_c) - f_c(r_d))/(f_D(r_c) - f_D(r_d)))$
Recalculate the joint weight for each link
$c_{ij}^{\lambda} = c_{ij} + \lambda d_{ij}$
Step 7. Recalculate the minimum cost path r from s to d. If $f_{c^\lambda}(r_\lambda) - f_{c^\lambda}(r_\lambda)$, then r_λ is the optimal path. Otherwise, let $r_d = r_\lambda$ and go to step 6 to recalculate.

Algorithm 1: The algorithm flow.

Table 1: Experimental parameters used in simulations.

Parameter	Parameter value
Link capacity	1 Gbps
Disk size	10–20 G
Memory size	1–10 G
Dirty pages of remaining memory	100–300 MB
Dirty page rate	100 Mbps
Link propagation delay	30 ms

4.2.1. Comparison of Overall Migration Time. We choose one-person-one-exit node as the source host and destination host for virtual machine migration. In evaluating the overall performance of the migration, we vary the size of the mirror file between 10 GB and 20 GB. Similarly, we set the memory size of each VM to 1 GB. The dirty page rate is set to 100 Mbps, and the remaining dirty page size is 100 MB [22]. The simulation results, in terms of the migration time, are shown in Figure 3.

It can be seen from Figure 3 that as the size of the image file increases, the overall migration time of the dynamic network transmission of virtual machines will also increase. Moreover, a comparison of the two algorithms shows that the QMA strategy proposed in this paper takes significantly less time to migrate network dynamic transmission of virtual machines than the benchmark algorithm because the implementation of the QMA strategy is based on the SDN architecture. The centralized controller can formulate QoS dynamic routing strategies based on the current network status information, which can provide better transmission bandwidth for network dynamic transmission virtual machine migration. In addition, the multipath transmission is allowed in our implementation. Therefore, it can be seen from the figure that, with the increase of mirror files, QMA can shorten the overall migration time; therefore, the dynamic transmission performance of the network can be significantly improved.

Also, under the condition of a certain mirror file, we studied the impact of the changes and variations in memory size on the overall migration time. The image size is set to 10 GB in the simulation. It can be seen from Figure 4 that when the size of the image file is certain, as the size of the memory file increases, the overall migration time of the virtual machine dynamically transmitted over the network

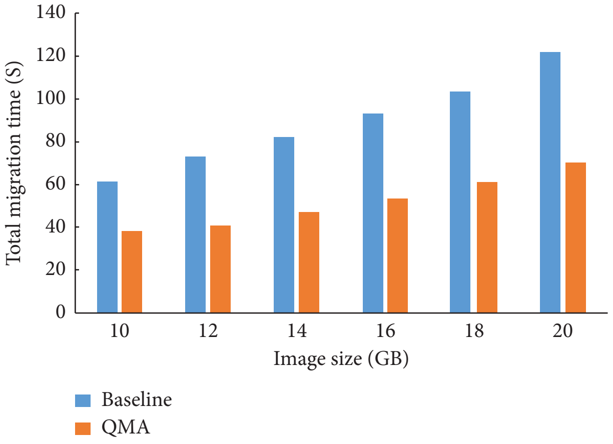

Figure 3: Relationship between the overall migration time and the image file size.

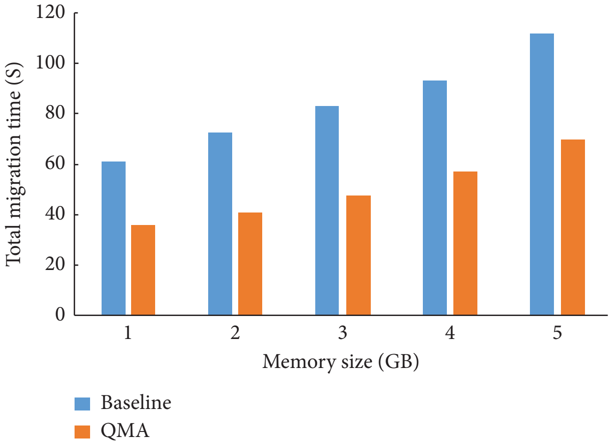

Figure 4: Relationship between the overall migration time and the memory size.

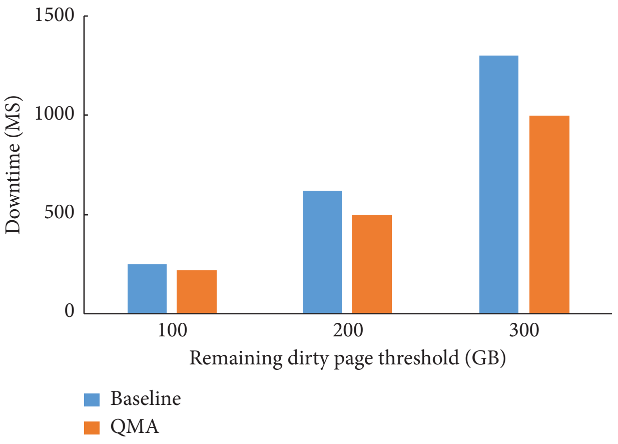

Figure 5: Relationship between downtime and memory remaining dirty pages (thresholds).

will also increase. Comparing the two algorithms, the migration time of the QMA strategy proposed in this article is less than that of the benchmark algorithm. This suggests that the QMA strategy can provide better routing strategies, ensure the minimum transmission bandwidth for migration, and perform network dynamic transmission virtual under relatively good network conditions. We observed that the proposed migration policy can effectively reduce the migration time and improves the performance of the network dynamic transmission. The results are shown as the average values across several experiments.

4.2.2. Downtime Comparison. Next, we verify the impacts of different dirty page (predefined threshold values) on the migration downtime of dynamic network transmission of virtual machines, and the simulation outcomes are shown in Figure 5.

As shown in Figure 5, the downtime is significantly affected by the threshold values used for deciding the remaining dirty pages of the VM memory. As the amount of remaining dirty pages increases, the downtime copy time of the network dynamic transfer virtual machine migration will gradually increase. Since the QMA strategy can be used to optimize the selection of paths and obtain better transmission bandwidth during migration, therefore, the execution effects of the proposed strategy are considerably better than that of the benchmark algorithm. We are aware that different threshold values will produce different outcomes; therefore, we ran the experiments several times and the results are the mean values. We observed significantly large variations (standard deviation values) amongst the obtained results. This verifies the generalization of outcomes of the proposed algorithms under different situations.

5. Conclusions and Future Work

This article focuses on improving the migration performance through optimizing the network resources for migration time in cross-data center scenarios, thereby reducing the migration time and downtime. First, a new network architecture, based on the SDN, is introduced. The SDN architecture can make the network configuration in a more flexible and adaptable way. Under this architecture, a QoS-based stream parallel transmission mechanism (QMA) is proposed to dynamically transmit virtual machines to the network. The migration resources are divided into different QoS flows, and the corresponding QoS flow queues are established. Moreover, the SDN controller is used to make dynamic routing decisions for each QoS flow in the queue according to the current network conditions and optimize the routing decision algorithm for each migration flow. We observed that the proposed algorithm provides better network resources to optimize the network dynamic transmission bandwidth and the transmission delay. The simulation results show that, compared with the virtual machine migration under the traditional network, the QMA mechanism can significantly reduce the overall migration time and downtime, thereby improving the dynamic transmission performance of the network. In the future, we will consider generalizing our model while using different experimental setup, heterogenous nodes, large number of VMs running various kinds of applications, and other experimental parameters. We will also consider using some real virtual machine migration datasets to provide a better view of the obtained outcomes. Other robust migration models should also be considered in the evaluation process. Moreover, migration might not be effective from energy consumption point of view; therefore, energy efficiency and electricity prices at different data centers and their geographical locations should be considered for further research and investigation. Scenarios should also be identified in cases when migration will not be effective and, therefore, should be avoided.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was supported by Jiangsu Modern Educational Technology Research Project (2019-R-84230) and Jiangsu Social Education Planning Project (JSS-L-2019018).

References

[1] Y. Tang, W. Wang, W. Wang, and B. Xu, "Design and implementation of SDN and traditional IP network interconnection architecture," *Computer Engineering and Science*, vol. 3, no. 12, 2017.

[2] Y. Bo and C. Huang, "Research on new network architecture of SDN," *Journal of Jiamusi University (Natural Science Edition)*, vol. 6, no. 3, 2018.

[3] L. Liu and X. Fang, “Research and discussion on SDN architecture and security,” *Computer Knowledge and Technology*, vol. 11, no. 13, 2015.

[4] P. Song, J. Wang, and L. Wu, “Design and application of data center export SDN architecture,” *Journal of Wuhu Vocational and Technical College*, vol. 8, no. 2, 2020.

[5] P. Zhang and Q. Sun, “Research on overlay VPN technology based on sdn architecture,” *Telecommunication Technology*, vol. 12, no. 6, 2016.

[6] Y. Wang, K. Yao, and Y. Lu, “Reentry trajectory design based on neural network dynamic optimization,” *Systems Engineering and Electronic Technology*, vol. 37, no. 10, pp. 2347–2351, 2015.

[7] W. Tian and Y. Zhao, “Data center resource optimal scheduling: theory and practice,” *Computer Security*, vol. 11, no. 3, p. 56, 2014.

[8] L. Wang, G. Du, R. Yu et al., “IDVE: virtual domain consistency migration mechanism for cross data centers,” *Journal of Sichuan University: Engineering Science Edition*, vol. 47, no. 1, pp. 20–26, 2015.

[9] W. Ma, C. Hu, W. Wang et al., “Research on consistency algorithm in low bandwidth WAN environment,” *Computer Engineering*, vol. 42, no. 9, pp. 94–99 + 104, 2016.

[10] S. Liu, J. Jiang, and G. Yang, “Replica based fast migration algorithm for virtual machines across data centers,” *Journal of Tsinghua University: Natural Science Edition*, vol. 55, no. 5, pp. 579–584, 2015.

[11] X. Dai, J. Nan, L. Huang et al., “Improved dynamic migration algorithm of memory pre copy,” *Computer Engineering and Application*, vol. 52, no. 23, pp. 88–93, 2016.

[12] D. Gu and X. Liu, “Research on virtual private cloud network architecture based on virtual private LAN service,” *Computer CD Software and Application*, vol. 17, no. 18, pp. 105-106, 2014.

[13] M. Lu and F. Zhao, “Optimization of virtual machine online migration in cloud computing environment,” *Information Technology*, vol. 9, no. 11, pp. 128–132, 2014.

[14] D. Darsena, G. Gelli, A. Manzalini et al., “Live migration of virtual machines among edge networks via WAN links,” in *Proceedings of the Future Network and Mobile Summit*, pp. 1–10, Lisboa, Portugal, July 2013.

[15] L. Sun and S. Shen, “A cloud management platform based on openstack,” *Computer Technology and Development*, vol. 26, no. 1, pp. 185–189, 2016.

[16] J. Zhang and Y. Wu, “Design and implementation of flow table forwarding in openflow switch,” *Journal of China Institute of Metrology*, vol. 26, no. 3, pp. 316–323, 2015.

[17] C. Xue, C. Ma, Z. Liu et al., “Design of a secure SDN controller architecture,” *Information Network Security*, vol. 4, no. 9, pp. 34–38, 2014.

[18] Z. Wang, X. Wang, and X. Jin, “Mbalaneer: dynamic prediction and allocation of virtual machine memory resources,” *Acta Software Sinica*, vol. 25, no. 10, pp. 2206–2219, 2014.

[19] S. Liu, “Improvement of transmission bandwidth by using transmission equipment in Internet services,” *Electronic Technology and Software Engineering*, vol. 4, no. 16, p. 55, 2013.

[20] D. Gao, L. Wang, and B. Xie, “Network selection optimization based on terminal QoS requirements,” *Communication Power Technology*, vol. 30, no. 6, pp. 75–77, 2013.

[21] A. Ahmad, L. Lu, Y. Zhu, and J. Yu, “An energy efficient algorithm for virtual machine allocation in cloud datacenters,” in *Proceedings of the 11th Conference on Advanced Computer Architecture*, IEEE, Weihai, China, August 2016.

[22] A. A. Khan, M. Zakarya, R. Khan, I. U. Rahman, M. Khan, and A. U. R. Khan, “An energy, performance efficient resource consolidation scheme for heterogeneous cloud datacenters,” *Journal of Network and Computer Applications*, vol. 150, Article ID 102497, 2020.