

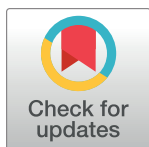
RESEARCH ARTICLE

Deploying external bandwidth guaranteed media server clusters for real-time live streaming in media cloud

Weizhan Zhang ^{*}, Zhichao He, Biao Du, Minnan Luo, Qinghua Zheng

MOEKLINNS Lab, Department of Computer Science and Technology, Xi'an Jiaotong University, Xi'an, China

^{*} zhangwzh@xjtu.edu.cn



Abstract

The cloud-based media streaming service is a promising paradigm for multimedia applications. It is attractive to media streaming service providers, who wish to deploy their media server clusters in a media cloud at reduced cost. Since the real-time live streaming service is both a bandwidth-intensive and quality-sensitive application, how to optimize the internal bandwidth utilization of a data center network (DCN) as well as guarantee the external bandwidth of the real-time live streaming application, is a key issue of deploying virtual machine (VM)-hosted media server cluster in a media cloud. Therefore, in this study, we propose an external-bandwidth-guaranteed media server cluster deployment scheme in media cloud. The approach simultaneously considers the outside bandwidth requirement of a tree-based media server cluster for live streaming and the intra-bandwidth consumption of a DCN. The proposed scheme models the optimal problem as a new terminal-Steiner-tree-like problem and provides an approximate algorithm for placing the media servers. Our evaluation results show that the proposed scheme guarantees the external bandwidth requirement of a real-time live streaming application, at the same time, greatly reduces the intra-bandwidth utilization of a media cloud with different DCN structures.

OPEN ACCESS

Citation: Zhang W, He Z, Du B, Luo M, Zheng Q (2019) Deploying external bandwidth guaranteed media server clusters for real-time live streaming in media cloud. PLoS ONE 14(4): e0214809. <https://doi.org/10.1371/journal.pone.0214809>

Editor: Francisco Martínez-Álvarez, Pablo de Olavide University, SPAIN

Received: October 5, 2018

Accepted: March 20, 2019

Published: April 3, 2019

Copyright: © 2019 Zhang et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the manuscript and its Supporting Information files.

Funding: This work is supported by The Fundamental Theory and Applications of Big Data with Knowledge Engineering under the National Key Research and Development Program of China with grant number 2016YFB1000903, the National Science Foundation of China under Grant Nos. 61772414, 61532015, 61702400, 61532004, 61721002, 61472317, 61502379, the MOE Innovation Research Team No. IRT17R86, the

Introduction

Due to the centralized management of elastic resources, the cloud can provide scalable data storage, computation, and networking services at a reduced cost [1]. Thus, media streaming service providers are enticed to deploy their applications in the cloud. Recently, the media cloud came into existence [2]. Some researchers have already studied how the media cloud can be efficiently used for cloud-based media streaming applications [2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]. Because these media streaming applications are resource-intensive, how to maximize resource utilization is still a major challenge for cloud-based media streaming service providers.

Virtualization technology provides an effective means to improve the resource utilization of the cloud. Many recent studies have concentrated on virtual machine placement and migration in the cloud. In these studies, the deployment issue of the virtual machine (VM) server is often

Project of China Knowledge Centre for Engineering Science and Technology, and the consulting research project of Chinese academy of engineering "The Online and Offline Mixed Educational Service System for 'The Belt and Road' Training in MOOC China." The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

Competing interests: The authors have declared that no competing interests exist.

described as a multidimensional vector packing problem. Optimization objectives often focus on resource utilization, networking overhead, energy efficiency, and migration times. The solutions include dynamic deployment or static placement, and their goals are mainly achieved using the heuristic greedy algorithm to obtain an approximate optimal solution [17]. Among them, several studies have taken the internal bandwidth resources of a data center network (DCN) into account [18, 19, 20, 21, 22, 23, 24, 25]. These traffic-aware solutions optimize the resource utilization of intra-bandwidth in a DCN, which is the critical bottleneck resource in a media cloud [26]. However, these VM management studies mainly took the infrastructure of the cloud into account, lacking consideration of the quality requirement of media streaming applications. Therefore, they cannot be adopted directly for many media streaming applications.

This study focuses on deploying real-time live streaming services in a media cloud. The media server cluster of a real-time live streaming service has special properties. The popular HTTP streaming or HTTP adaptive streaming architecture cannot be adopted under this context because of the pull-based nature of HTTP. Real-time live streaming services in media cloud, such as time-sensitive video conferencing and virtual classrooms, usually employ a tree-based media server cluster in data center. They usually let the root VM push the video to other VMs to provide real-time, large-scale streaming services. In this situation, the leaf VM nodes in the tree truly provide the real-time live streaming services for clients simultaneously. In this manner, the service provider can isolate server internal traffic and external video streaming for user application, to enhance the overall service quality. Therefore, the external bandwidth of the leaf VM nodes, connecting the clients outside the data center, should be guaranteed to ensure the quality of the large-scale services. For example, real-time live streaming applications based on web real-time communication (WebRTC) [27] may need a tree-based multipoint controller (MCU) cluster to enlarge the one-to-many real-time live streaming service. This research is motivated by the case of deploying a tree-based MCU media server cluster in a media cloud for a virtual classroom system. Therefore, to fully utilize the resources of the cloud as well as to provide high-quality, real-time live streaming services, the deployment strategy used for a media server cluster needs to consider both the inside and outside bandwidth resources of the data center. In this manner, the service requirement of media streaming can be fully known and guaranteed, and the efficient internal bandwidth utilization of the media cloud services can also be achieved.

In this study, we propose a VM-hosted media server cluster deployment scheme for external-bandwidth-guaranteed, real-time live streaming services. Firstly, the scheme proposes a mathematical model to illustrate the services deployment problem and demonstrates the difficulty. Then, a greedy media server placement algorithm is introduced to optimize the internal bandwidth utilization of a DCN under the application constraints of real-time live streaming services. Fig 1 shows the schematic description of the whole method. The approach contributes the following new ideas.

- First, the approach simultaneously considers the outside bandwidth requirement of a tree-based media server cluster for live streaming and the intra-bandwidth consumption of a DCN. The approach models this specific application requirement as a newly defined terminal-Steiner-tree-like problem, extending the target and domain of the traditional VM management algorithms.
- Second, the approach designs an approximation algorithm called cluster-and-union to deploy the media streaming cluster. The algorithm ensures that the number of leaf nodes of the tree-based media server cluster meets the bandwidth requirement of the real-time live streaming service, and at the same time, obtains an approximate optimal minimum tree solution to minimize the intra-bandwidth consumption of the DCN.

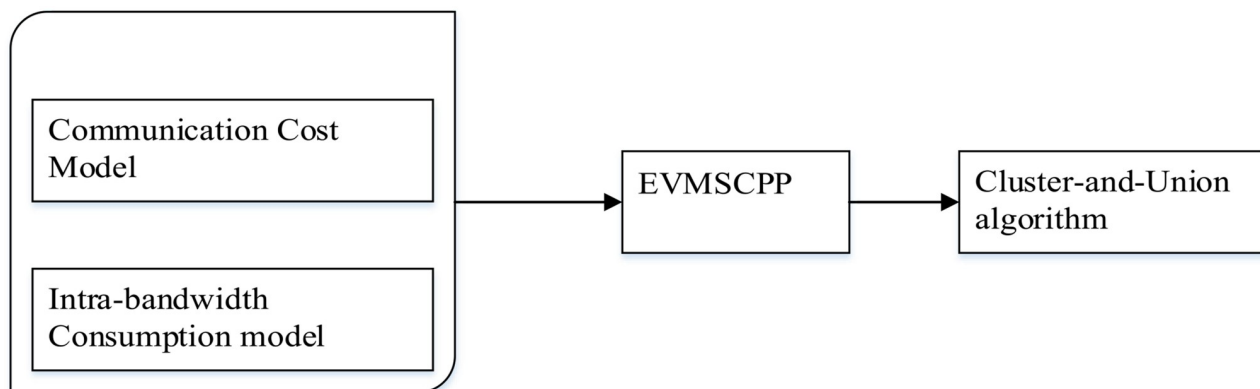


Fig 1. Model of the core idea.

<https://doi.org/10.1371/journal.pone.0214809.g001>

The remainder of this paper is organized as follows. We summarize the related works in Section 2. We give the problem formulation, establish the mathematical model, and introduce the approximation algorithm solution in Section 3, and then we evaluate the performance of our proposal via a simulation in Section 4. Finally, we conclude the paper and discuss potential future works in Section 5.

Related works

Media cloud computing

Recently, taking advantage of the centralized management of elastic resources in cloud computing, media cloud computing has come into existence [2]. Research on media cloud computing mainly focuses on resource management of the media cloud. Researchers have tried to enhance media streaming services by optimizing the resource allocation of the media cloud to fully utilize the elastic and on-demand nature of cloud resource provision. Specifically, the authors of [28] proposed a novel SDN-based game-aware network management method for cloud gaming. The scheme assigns game servers and selects the communication path by considering the types of games, sever loads, and path delays. GamingAnywhere [29] also concentrated on cloud gaming and provided an open source system. DRES [6] derived an optimal redirection policy under a cloud-centric media network, which reallocates user requests to multiple VM hosts to scale the service capacities. The research in [30] provided a generalized framework for computing the resources required to support multiple services via the utilization of an earliest deadline first strategy. DCC-VOD [7] presented a distributed video-on-demand system design based on cloud computing, focusing on the implementation of load balancing among servers. OPT-ORS [8] investigated the tradeoff between the cost incurred by VM instance procurement and the achieved QoE of end users under commercial pricing models. DREAM [9] proposed a distributed heuristic algorithm, aiming to solve the resource reservation and scheduling problem of configuring the cloud utility to meet SLAs for VoD applications at a modest cost. CALMS [13] adaptively leased and adjusted cloud server resources in a granularity to accommodate the temporal and spatial dynamics of demands from live streaming users. The research [15] introduced a segment-based storage and transcoding trade-off strategy for multi-version VoD systems in the cloud. In addition, some media cloud computing studies also examined the load balancing problem among servers for multi-media computing, such as for image enhancement or video transcoding, to efficiently utilize the parallel computing power of the cloud [16, 31].

In summary, some studies have focused on providing media streaming services based on the media cloud. Among them, some have already considered the resource allocation among different VM-hosted servers. However, as of now, few efforts have been devoted to exploring in detail how the server cluster of real-time live streaming can be deployed in a DCN.

Traffic-aware VM management in the cloud

Virtualization technology provides an effective means to optimize the resource utilization of an elastic cloud, which has recently been studied intensively. Most related VM management approaches can be roughly categorized as either VM placement or VM migration schemes.

VM placement or migration studies address the problem of how to place or replace VMs in a physical machine according to the specific optimization needs. In these studies, the management issue of the VM server is often described as a multidimensional vector packing problem with different optimization objectives. Among them, some researchers have already paid attention to the internal bandwidth consumption of a DCN. The research presented in [18] proposed a traffic-aware VM placement algorithm to improve network scalability. The study optimized the placement of VMs on host machines, by which traffic patterns among VMs can be aligned with the communication distance between hosts. The research [19] proposed an online joint VM placement and routing algorithm to minimize the traffic costs of a DCN, which employed VM migration to combine the VM placement and the data routing among VMs. The authors of [20] focused on maximizing the benefit of the overall communication sent by the VMs to a single designated point in the data center. Shadow [21] proposed a combined VM routing and VM placement algorithm, named the Shadow scheme, which minimizes the maximum appropriately defined data center utilization. The research presented in [22] focused on the optimized placement of VMs to minimize the cost by balancing DCN traffic and the utilization of physical machines. The authors of [23] analyzed how much bandwidth is required to guarantee the total migration time and downtime of live VM migrations and gave the bandwidth value required to satisfy the performance metrics of live VM migrations. Vmbuddies [24] designed and implemented a coordination system for correlated VM migrations, which is one of the earliest researches to solve the correlated migration problem of multi-tier application in cloud environments. The study migrates tightly-coupled VMs simultaneously, and minimizes the migration costs of multi-tier applications. Our precious work [25] also proposed a VM collaborative migration scheme. It targets at media cloud shared by concurrent media applications, and provides a generalized solution to solve the problem of VM migration in media cloud.

In summary, these studies focused on multiobjective constrained VM placement or migration. Some studies have already taken the intra-traffic of a DCN as the constraint, but the relevant studies did not consider the tree-based server clusters or bandwidth requirement of real-time live streaming.

Materials and methods

Communication cost model of DCN

The internal bandwidth of DCNs has increasingly become a bottleneck restricting the utilization of DCNs [1]. The VM-hosted media server cluster placement scheme should be fully aware of the DCN structure since media applications are bandwidth intensive. In recent years, researchers have carried out extensive studies on DCN construction [32]. According to different construction rules and interconnected technologies, DCNs can be roughly categorized as either switch-based DCNs, server-based DCNs, module-based DCNs, or random-based DCNs.

Although the topologies and the interconnection rules of DCNs vary, their communication cost model can be uniformly defined as a weighted connected graph. Suppose that $G = (V, E, w)$ is a simple undirected weighted complete graph with n vertices and $n(n-1)/2$ edges. The vertex set V in G indicates the deployment point (DP) set in the DCN in which a physical machine is placed for hosting VMs. E is defined as the set of edges connecting two DPs. w denotes the weighting function of E . The value of $w(u, v)$, $u, v \in V$, indicates the communication cost between the DPs u and v . It denotes the minimum number of links by which data is transferred. In addition, the weighting function w satisfies the following triangle inequality: $w(u, v) < w(u, a) + w(a, v)$, $a, u, v \in V$.

In this manner, all of the communication costs between two arbitrary DPs composes an intra-communication cost matrix, which is denoted by $C = \{c_{ij}\}_{n \times n}$, where $c_{ij} = w(v_i, v_j)$. Taking a traditional three-tier tree-based DCN as an example, the communication cost between DP v_i and DP v_j is 2 when they are connected via the same access switch and is 4 when they are connected to two different access switches belonging to the same aggregation switch. If they belong to different aggregation switches, the communication cost between DP v_i and DP v_j is 6.

Intra-bandwidth consumption model of DCN

For a VM-hosted cluster-based application, the traffic pattern among VMs can also be described as a weighted connected graph $G' = (V', E', w')$.

A vertex $v'_i, v'_j \in V'$, indicates a VM in the cluster, and an edge (v'_i, v'_j) indicates traffic flow between v'_i and v'_j . $w'(v'_i, v'_j)$ shows the average traffic rate between v'_i and v'_j ; if $(v'_i, v'_j) \notin E'$, $w'(v'_i, v'_j) = 0$. Thus, the traffic rate matrix for the VM-based cluster application can be denoted by $D = \{d_{ij}\}_{m \times m}$, where $d_{ij} = w'(v'_i, v'_j)$. Suppose that function ψ maps the VM to the DP. For instance, if the VM v'_i is deployed to the DP v_j , then $\psi(v'_i) = v_j$. Thus, it is easy to quantify the intra-bandwidth consumption caused by deploying the logic structure-known VM-based cluster application in a DCN. This consumption can be modeled using the following formula:

$$\chi(G, G', \psi) = \sum_{(v'_i, v'_j) \in E'} w'(v'_i, v'_j) * w(\psi(v'_i), \psi(v'_j)) \quad (1)$$

External-bandwidth-guaranteed, VM-hosted media server cluster placement problem (EVMSCPP)

In this section, we define and formulate the external-bandwidth-guaranteed, VM-hosted media server cluster placement problem (EVMSCPP) based on the communication cost model and the intra-bandwidth consumption model of a DCN. We apply it to a terminal-Steiner-tree-like problem and then analyze its complexity.

The proposal focuses on a VM-hosted, cluster-based, real-time live streaming application such as large-scale live video broadcasting or a virtual classroom. Such applications organize the VM cluster into a tree-like topology structure. Only the VMs, functioning as leaf nodes in the tree, provide real-time live streaming services for clients simultaneously, as they are allocated sufficient external bandwidth resources. Suppose that they receive a copy of media data produced by the root VM at the rate of d and that each VM leaf node in the cluster is allocated the same output bandwidth resource b . In this manner, if the guaranteed external bandwidth requirement of the media streaming application is B , we should construct a logical tree

topology cluster with $k = \lceil B/b \rceil$ leaf nodes and determine a way to deploy the tree-like VM cluster to physical hosts in the DCN. To optimize the intra-bandwidth consumption of the DCN caused by media server cluster deployment, we can define the EVMSCPP as a problem that involves constructing a tree T that represents the topology of the VM-host media server cluster, with a constraint on the number of leaf nodes, and finding the VM placement method, denoted as ψ , used to minimize the following objective function:

$$\chi(G, T, \psi) = d \sum_{(v_i, v_j) \in T} w_G(\psi(v_i), \psi(v_j))$$

$$s.t.$$

$$leaf(T) = k \in (0, |V(G)|) \wedge |V(T)| \leq |V(G)|$$

Note that $leaf(T)$ stands for the number of leaf nodes in tree T , $d = w_T(e)$ is a constant and $\psi: V(T) \rightarrow V(G)$ is an injective function that reflects the mapping relation between the VM and DP.

In formula 2, when G , k , and d are given, minimizing the intra-bandwidth consumption of the DCN χ is equivalent to minimizing the simplified intra-bandwidth consumption of the DCN $\varphi(G, T')$, as shown in the following objective function:

$$\min_{T' \in \Pi} \{\varphi(G, T') = \sum_{(v_i, v_j) \in T'} w_G(v_i, v_j)\}$$

$$s.t.$$

$$\Pi = \{T' | V(T') \subset V(G) \wedge E(T') \subset E(G) \wedge leaf(T') = k\}$$

That is, the EVMSCPP above (formula 3) tries to find a subtree T' with k leaves whose weight is minimum in a weighted complete graph G . The simplified intra-bandwidth consumption $\varphi(G, T')$ represents the sum of the communication costs among DPs caused by T' . It is similar to the terminal Steiner tree (TST) problem in mathematics. The TST problem is defined as follows [33]. Let $G = (V, E, w)$ be a positive edge-weighted complete graph with vertex set V , edge set E , edge weighting function w satisfying the triangle inequality, and target vertex set $S \subset V$. The terminal Steiner tree problem involves finding a minimum weighted subtree denoted by T^* in G that interconnects all target vertices such that every target vertex in S appears as a leaf vertex in the subtree T^* . From the definition of the TST problem, the set of leaf nodes S in the TST problem is given. However, in our EVMSCPP, only the number of leaf nodes k is known. The elements of the leaf nodes can be selected from the entire vertex set V and are unknown in advance. Thus, the terminal Steiner tree problem is a special case of the EVMSCPP. The TST problem has been demonstrated to be NP-hard and MAX SNP-hard (APX-hard) [2]. Thus, the EVMSCPP is also NP-hard and has a nondeterministic polynomial time solution. In the next section, we will introduce the approximation algorithm solution of the EVMSCPP in detail.

Approximation solution of the EVMSCPP

There have been some studies on the approximate solution of the TST problem. For example, GH Lin put forward a three-step TST approximation algorithm [33]. In the EVMSCPP, since the set of leaf nodes is nondeterministic, the traditional approximation algorithms of the TST problem cannot be directly adopted to solve our problem. We need to choose the leaf node set from G in advance before we use the corresponding approximate TST solutions. We can give

an analysis of the approximation ratio for employing the TST approximate solutions to solve our EVMSCPP problem.

Lemma Suppose that there is an approximation algorithm for the TST problem with approximation ratio ρ ; then, there is also a corresponding approximation algorithm for the EVMSCPP with approximation ratio ρ .

Proof Let T' denote an optimal tree in G such that $\text{leaf}(T') = k$. Let $S_i, i \in \{1, 2, \dots, C_{|G|}^k\}$, denote a vertices set in G , and $|S_i| = k, \forall i \in \{1, 2, \dots, C_{|G|}^k\}$. Let T^i denote the TST tree for target vertices $|S_i|$ in G obtained using one TST approximation algorithm. Let T^* denote the approximate EVMSCPP tree obtained by employing the TST approximation algorithm $C_{|G|}^k$ times. That is, T^* is the minimum tree selected from $T^i, i \in \{1, 2, \dots, C_{|G|}^k\}$, by employing the TST approximation algorithm $C_{|G|}^k$ times. By definition,

$$\begin{aligned} w_G(T^*) &= \min(w_G(T^i), i = 1, 2, \dots, C_{|G|}^k) \\ &\leq w_G(T^i), \forall i \in \{1, 2, \dots, C_{|G|}^k\} \\ &\leq \rho w_G(T') \end{aligned} \quad (4)$$

where ρ is the approximation ratio of the adopted TST approximation algorithm. This proves the lemma.

The proof indicates that approximation solutions of the EVMSCPP can always be obtained with the same approximation ratio by employing the approximation solutions of the TST problem. However, this means that the complexity of the EVMSCPP approximation algorithm is $C_{|G|}^k$ times that of the corresponding TST approximation algorithm. Therefore, we further propose another approximation algorithm called “cluster-and-union” in this study, which leverages the network topology feature of data centers. The approximation ratio of the cluster-and-union algorithm is related to the different DCN structures for specific applications, but the complexity of the proposed cluster-and-union approximation solution is reduced to $O(n)$. The algorithm is as follows. As mentioned above, the EVMSCPP is NP-hard. However, if $k = |G| - 1$, then it is easy to find a subtree T whose vertex set covers all the vertices in G and in which only one vertex (i.e., the root) is a non-leaf node, with the other vertices functioning as leaves. Thus, the main idea in solving this problem is to reduce the EVMSCPP to several, small special cases and then to unify the subtrees constructed in those special cases. The cluster-and-union algorithm is mainly composed of three steps.

First, we partition the n DPs into p clusters by using the intra-communication cost among DPs and then sort the set of clusters according to the simplified intra-bandwidth consumption value. We run the classical clustering algorithms based on the intra-communication cost matrix that satisfies the triangle inequality to partition the DPs into several clusters. Taking the three-tier architecture network of DCNs as an example, the clustering characteristic of DPs is obvious, and DPs connected to the same access switch can be gathered as a cluster. Thus, the number of clusters p can be set to the number of access switches in the DCN. Second, we select the first t clusters, the total simplified intra-bandwidth consumption costs of which are the top t costs when ranked in ascending order, and make sure that the sum of the numbers of leaf nodes of all subtrees created from the clusters is not less than k . For each cluster, we create a subtree whose number of leaves is one less than the number of DPs in each cluster and whose weight is minimum. Notice that the minimum span tree will be the same as the EVMSCPP tree in this special case. Finally, after obtaining t subtrees with a total of k leaves and t roots via the previous steps, we unify the t subtrees to construct the final approximate subtree by joining the t roots after running the classical minimum spanning tree algorithm. The pseudocode for the algorithm is described below. Fig 2 shows the workflow of the algorithm.

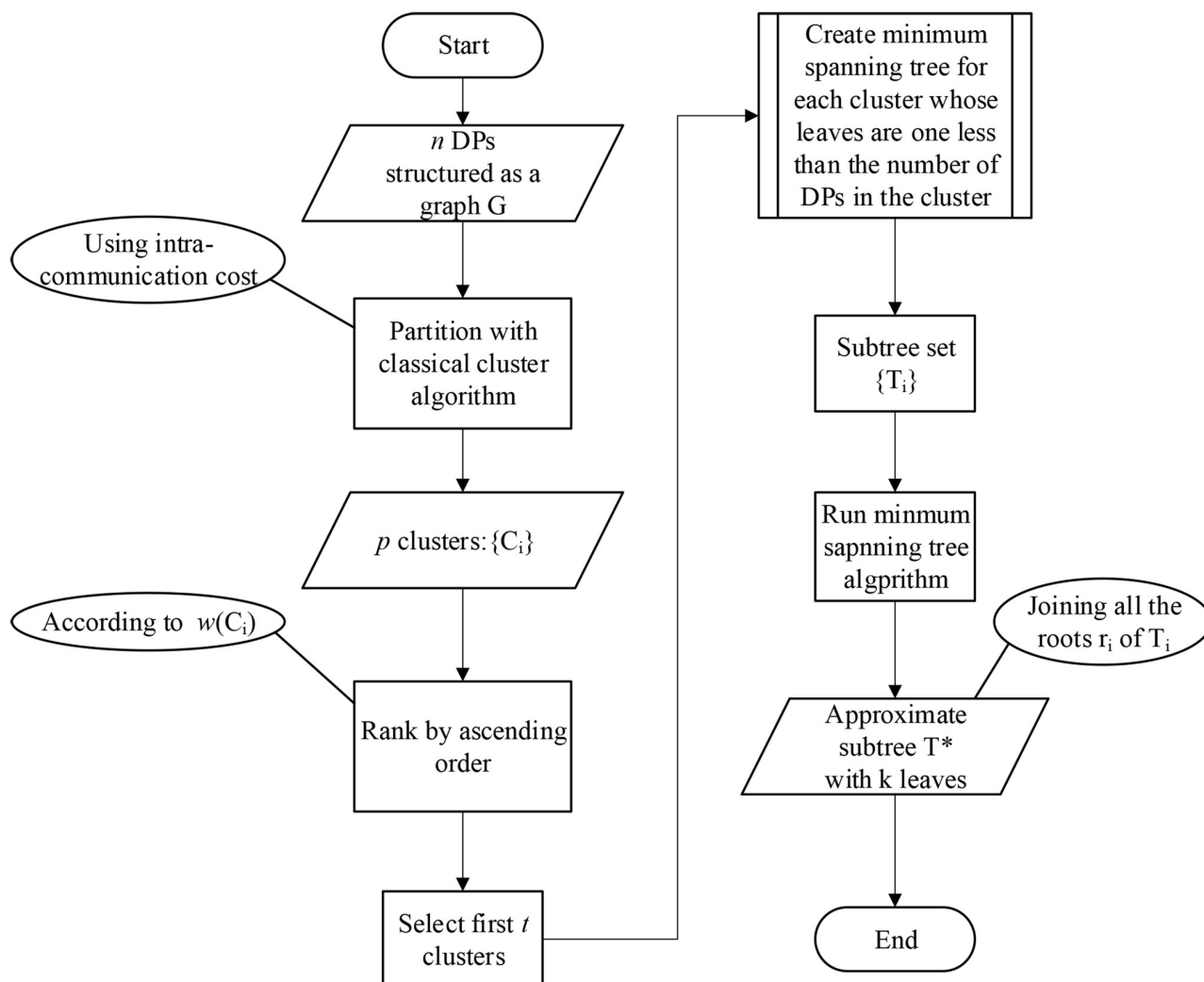


Fig 2. Workflow of Cluster-and-Union algorithm.

<https://doi.org/10.1371/journal.pone.0214809.g002>

Algorithm 1 Cluster-and-Union

INPUT:

G (Simple undirected weighted complete graph).

k (Number of leaves of the required subtree).

p (Number of physical machine clusters).

OUTPUT: T^* (The approximate subtree with k leaves).

1: Partition G into p clusters: $\{C_i\}$.

2: Sort $\{C_i\}$ in ascending order according to $w(C_i)$: $C_1, C_2, \dots, C_p$.

3: Select the first t clusters $(C_1, C_2, \dots, C_t)$, where

$$\sum_{i=1}^t |C_i| - t - k \in [0, |C_t|).$$

4: **for** $i = 0$ to t **do**

5: **if** $i \neq t$ **then**

6: Create the minimum spanning tree T_i in C_i with $|C_i| - 1$ leaves, the root of which is r_i .

7: **else**

8: Create the minimum spanning tree T_t in C_t with $k - \sum_{i=1}^{t-1} (|C_i| - 1)$ leaves, the root of which is r_t .

9: **end if**

```

10: end for
11: Union( $\{T_i\}$ ) returns the approximate subtree  $T^*$  with  $k$  leaves by
    joining all the roots  $r_i$  of  $T_i$  after running the classical
    minimum spanning tree algorithm.

```

Results

Evaluation environment and comparison schemes

The performance of the proposed cluster-and-union algorithm is demonstrated by CloudSim [34] in this section. We take the simplified intra-bandwidth consumption of DCN φ caused by the media server cluster, which was defined in formula 3, as the metric to evaluate the resource utilization of our scheme. The simplified intra-bandwidth consumption of DCN φ finally equals to the sum of the communication costs among DPs at which the media servers are located. It has the dimensions of hops. A lower φ value represents better resource utilization of the DCN resources. To show the effectiveness of the proposed algorithm, the simulation also sets up a comparison study. We use the straightforward random solution and the variation of the cluster-and-cut scheme [18] as the benchmarks. Different from the proposed algorithm, the random solution and the cluster-and-cut scheme require knowledge of the topology structure of the media server cluster. The traffic rate matrix D should be known in advance. Therefore, in the comparison study, the k -leaf tree topology obtained using the proposed algorithm is utilized as the input of the two benchmarks.

In detail, the default simulation settings are as follows. Four DCN topologies, i.e., Tree, Fat-tree, VL2, and BCube, are selected. The DCN can support 1024 DPs. The media server cluster is constructed with at least 32 leaf nodes. In this manner, the bandwidth requirement of a real-time live streaming service can be quantified and guaranteed. In the experimental study, there are three scenarios. First, the number of deployment points is set to 1024, and the number of leaf nodes of the media server cluster ranges from 64 to 512, which indicates the different bandwidth requirements of a live streaming service. Second, the number of deployment points ranges from 256 to 1024, and the number of leaf nodes is set to 64, which indicates the different scales of the DCN. The last scenario is similar to the first one. The number of deployment points is set to 1024, and the number of leaf nodes of the media server cluster still ranges from 64 to 512. However, different network parameters are adopted for the Tree, VL2, and BCube topologies to study the effectiveness of the proposed algorithm under different network parameters for the given DCN structures. Finally, to reduce the randomness of the simulation and experimental results, the evaluation results presented in this paper are the average results over twenty runs.

Results for different media server cluster scales

This subsection demonstrates the efficiency of the cluster-and-union algorithm for different media server cluster scales. The simulation presents the simplified intra-bandwidth consumption results obtained using the cluster-and-union algorithm for different numbers of leaf nodes, ranging from 64 to 512, of the media server cluster. For comparison, the simulation takes the traditional random solution as a basic benchmark. At the same time, the simulation takes the cluster-and-cut scheme as an additional benchmark to show the efficiency of the proposed algorithm. The detailed results of the intra-bandwidth consumption are shown in Fig 3. With increasing scale of the media server cluster under the given 1024 DPs, compared with the traditional random or cluster-and-cut media server deployment, the cluster-and-union algorithm greatly reduces the intra-bandwidth consumption for different cluster scales. This result occurs because the proposed method clusters the media servers together to minimize the

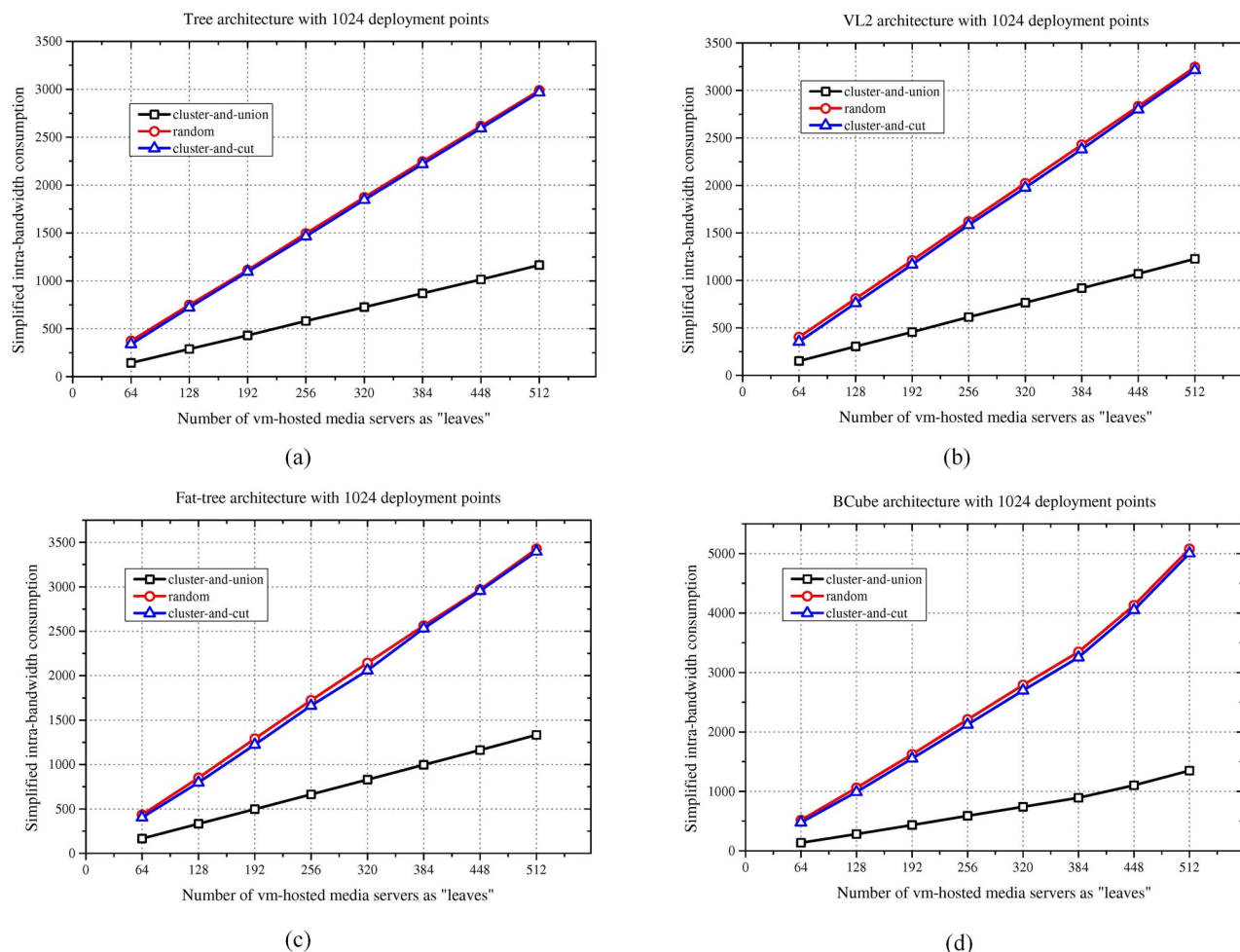


Fig 3. Intra-bandwidth consumption results for different scales of the media server cluster. (a) Tree. (b) VL2. (c) Fat-tree. (d) BCube.

<https://doi.org/10.1371/journal.pone.0214809.g003>

internal bandwidth consumption of the DCN. In this manner, the media server cluster is located in a local area and has a low internal bandwidth consumption. In contrast, the random method yields a random deployment result due to the uncertainty of the deployment point selection. The intra-bandwidth consumption of the DCN experiences linear growth as the scale of the media server cluster increases. It is interesting to find that the behavior of the cluster-and-cut method is the same as that of the random method. For a real-time live streaming application, the traffic rate between VM pairs is homogeneous, resulting in random clustering and cutting of the VM cluster. For this scenario, as had been indicated by the authors of the research [18], the cluster-and-cut method will provide little improvements no more than a random method. This outcome further demonstrates that the traditional traffic-aware VM management schemes in the cloud may fail to work when they do not consider the characteristics of the live streaming application.

Results for different DCN scales

This subsection of the simulation presents the simulation results of the cluster-and-union algorithm for different DCN scales. In this simulation, the number of leaf media server nodes is set to 64, while the scale of the DCN ranges from 256 to 1024. The detailed results regarding the

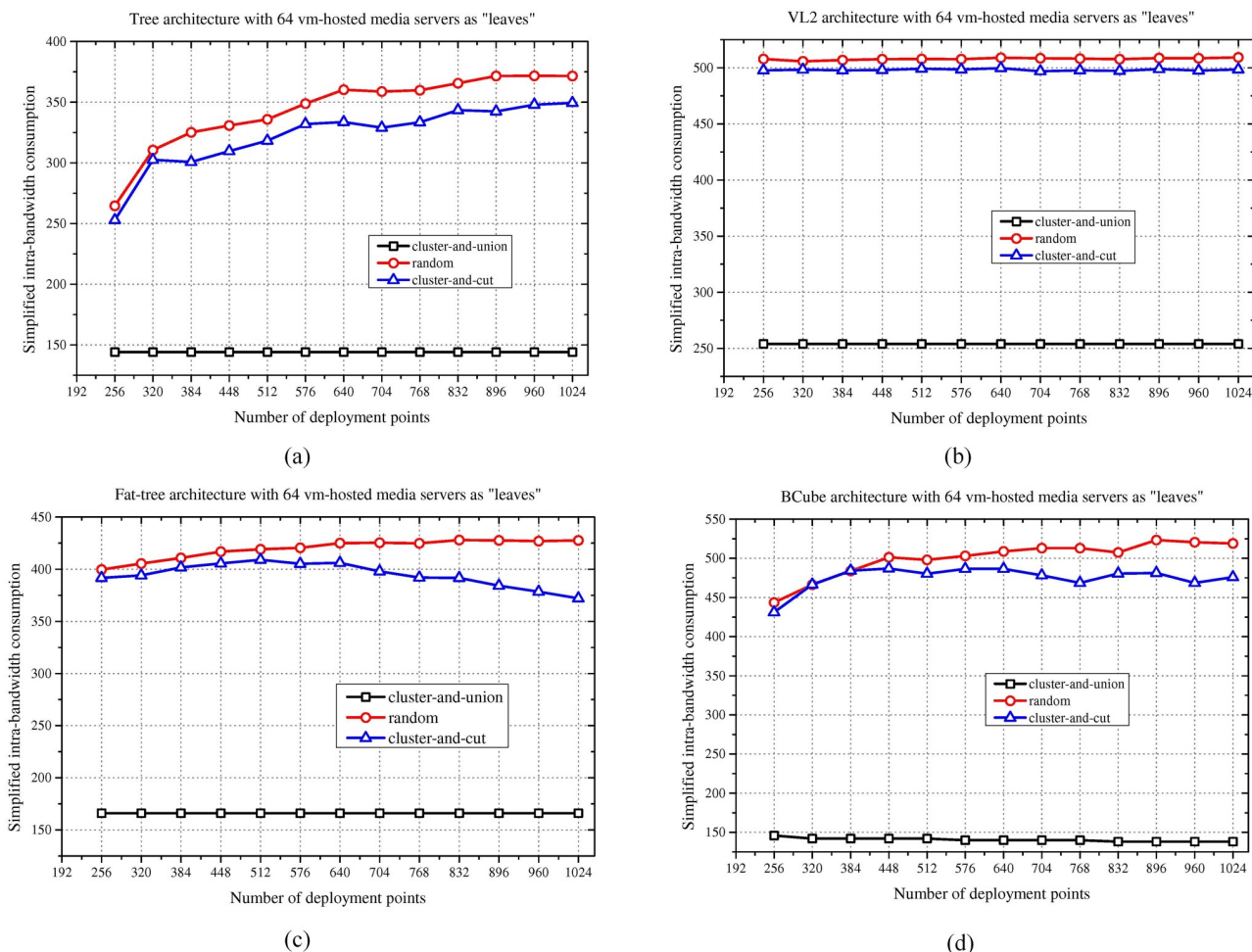


Fig 4. Intra-bandwidth consumption results for different DCN scales. (a) Tree. (b) VL2. (c) Fat-tree. (d) BCube.

<https://doi.org/10.1371/journal.pone.0214809.g004>

simplified intra-bandwidth consumption are shown in Fig 4. From the simulation results, as the DCN scale increases under the given 64 leaf nodes of the media server cluster, the intra-bandwidth consumption essentially remains stable at a low value for our method. This outcome occurs because the proposed methods cluster together the media servers associated with the same access switches or aggregation switches. In this manner, the deployment of the media server cluster is not greatly influenced as the DCN topology scale increases. In contrast, the random method may be influenced by the DCN topology because of the uncertainty of the deployment point selection. The cluster-and-cut method and the random method behave in the same way due to the homogeneous traffic rate.

Results for different parameters for the same DCN scale

This subsection of the simulation presents the simulation results for DCNs with the same size but different parameters. For the DCN with the Tree structure, the size of the DCN is determined as follows: $size = p_0 * p_1 * p_2$, where p_i indicates the downward connections of the i th layer switches. For VL2, the size of the DCN is determined as follows: $size = p_0 * n$, where p_0 indicates the downward connections of the access layer switches and n represents the number of upper

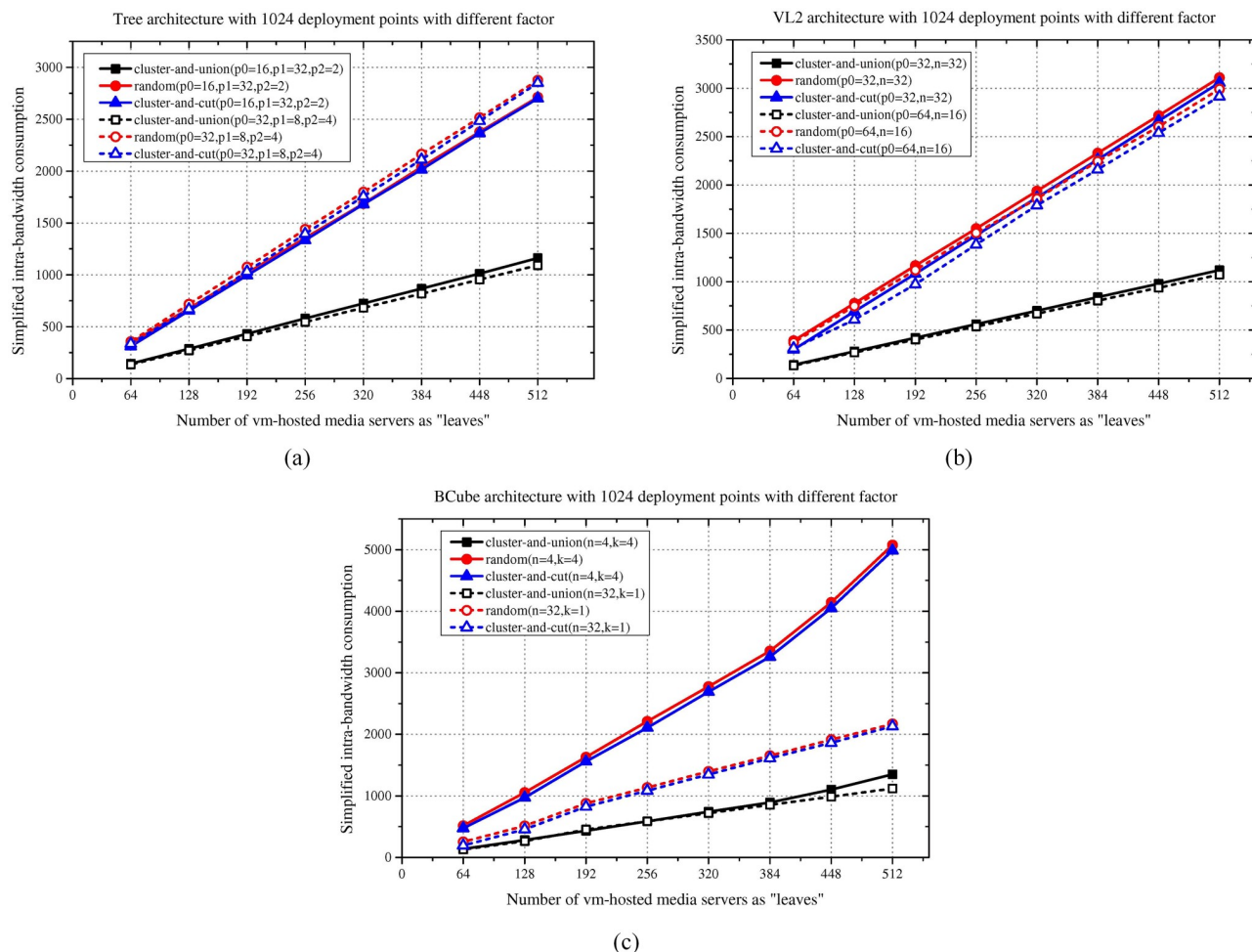


Fig 5. Intra-bandwidth consumption results for different parameters. (a) Tree. (b) VL2. (c) BCube.

<https://doi.org/10.1371/journal.pone.0214809.g005>

layer switches. For BCube, the size of the DCN is determined as follows: $size = n^{(k+1)}$, where k indicates the layers involved and n indicates the downward connections of the switches. Finally, for Fat-tree, the size of the DCN is determined as follows: $size = k^3/4$, where k indicates the downward connections between the access layer switches and the deployment points. Since the size of the Fat-tree topology is determined by only the connections k between the switches and the host servers, we cannot maintain the size of the DCN while changing the value of k . Therefore, in this subsection, we only provide the simulation results corresponding to the Tree, VL2, and BCube topologies. In the simulation, the number of deployment points is firmly set to 1024, but the network parameters for the given DCN structures are different. In the simulation, the number of leaf nodes still ranges from 64 to 512. According to the simulation results in Fig 5, the variation of parameters has very little impact on the final results for the Tree and VL2 DCNs. The proposed scheme outperforms the benchmarks significantly. Meanwhile, the variation of parameters in BCube influences the results obtained using the random and cluster-and-cut methods. However, our proposed algorithm still stably outperforms the two benchmarks.

Conclusion

This study addresses the VM-hosted media server cluster deployment issue that occurs in a media cloud for real-time live streaming services. An external-bandwidth-guaranteed VM placement scheme is presented to optimize the internal bandwidth utilization in a DCN as well as to guarantee the external bandwidth of a real-time live streaming application. The scheme models the optimal resource utilization problem as a terminal-Steiner-tree-like problem and demonstrates the difficulty of the problem. Furthermore, a greedy algorithm, which includes a greedy media server placement algorithm to find the minimum tree with the necessary leaf nodes, is introduced to solve the target problem. From the evaluation results, when compared with the traditional methods for different DCN topologies and media server cluster settings, the proposed scheme achieves a lower DCN intra-bandwidth consumption under the ultra-bandwidth constraint, which is beneficial for enlarging the application scale while still guaranteeing a satisfactory external bandwidth for the service.

In this study, although the proposed scheme considers the external bandwidth requirement of a real-time live streaming application, some other quality of service parameters or quality of experience metrics of live streaming application are not addressed. For example, the future work should include of designing a quality of experience driven VM placement method for live streaming services. Furthermore, future work should also attempt to devise a dynamic media server migration method in accordance with the dynamic nature of the cloud. Finally, in addition to an evaluation via simulation, future work should evaluate the efficiency of the algorithm in a real cloud-based virtual classroom environment and obtain experimental judgments.

Supporting information

S1 Dataset. Original data of experiments.
(ZIP)

Acknowledgments

This work is supported by The Fundamental Theory and Applications of Big Data with Knowledge Engineering under the National Key Research and Development Program of China with grant number 2016YFB1000903, the National Science Foundation of China under Grant Nos. 61772414, 61532015, 61532004, 61721002, 61702400, 61472317, 61502379, the MOE Innovation Research Team No. IRT17R86, the Project of China Knowledge Centre for Engineering Science and Technology, and the consulting research project of Chinese academy of engineering “The Online and Offline Mixed Educational Service System for ‘The Belt and Road’ Training in MOOC China”.

Author Contributions

Conceptualization: Weizhan Zhang.

Funding acquisition: Weizhan Zhang, Qinghua Zheng.

Investigation: Weizhan Zhang.

Methodology: Weizhan Zhang, Minnan Luo.

Software: Zhichao He, Biao Du.

Validation: Zhichao He, Biao Du.

Writing – original draft: Weizhan Zhang.

Writing – review & editing: Qinghua Zheng.

References

1. Armbrust M, Fox A, Griffith R, Joseph AD, Katz R, Konwinski A, et al. A view of cloud computing. *Communications of the ACM*. 2010; 53(4):50–58. <https://doi.org/10.1145/1721654.1721672>
2. Zhu W, Luo C, Wang J, Li S. Multimedia cloud computing. *IEEE Signal Processing Magazine*. 2011; 28(3):59–69. <https://doi.org/10.1109/MSP.2011.940269>
3. Kaseb SA, Mohan A, Koh Y, Lu YH. Cloud Resource Management for Analyzing Big Real-Time Visual Data from Network Cameras. *IEEE Transactions on Cloud Computing*. 2017; PP(99):1–1. <https://doi.org/10.1109/TCC.2017.2720665>
4. Clegg R, Landa R, Griffin D, Rio M, Hughes P, Kegel I, et al. Faces in the clouds: long-duration, multi-user, cloud-assisted video conferencing. *IEEE Transactions on Cloud Computing*. 2017; PP(99):1–1. <https://doi.org/10.1109/TCC.2017.2680440>
5. Wen Y, Zhu X, Rodrigues JJ, Chen CW. Cloud mobile media: reflections and outlook. *IEEE Transactions on Multimedia*. 2014; 16(4):885–902. <https://doi.org/10.1109/TMM.2014.2315596>
6. Tang J, Tay WP, Wen Y. Dynamic request redirection and elastic service scaling in cloud-centric media networks. *IEEE Transactions on Multimedia*. 2014; 16(5):1434–1445. <https://doi.org/10.1109/TMM.2014.2308726>
7. Shu C, Zhang X. Research on virtualization-based video-on-demand services architecture. In: *Proc. of SPIE*; 2011. p. 83491A–83491A9.
8. He J, Wen Y, Huang J, Wu D. On the Cost-QoE Trade-off for Cloud-based Video Streaming under Amazon EC2's Pricing Models. *IEEE Transactions on Circuits and Systems for Video Technology*. 2014; 24(4):669–680.
9. Zhao Y, Jiang H, Zhou K, Huang Z, Huang P. Meeting service level agreement cost-effectively for video-on-demand applications in the cloud. In: *Proc. of IEEE INFOCOM*. IEEE; 2014. p. 298–306.
10. Jin Y, Wen Y, Westphal C. Optimal transcoding and caching for adaptive streaming in media cloud: An analytical approach. *IEEE Transactions on Circuits and Systems for Video Technology*. 2015; 25(12):1914–1925. <https://doi.org/10.1109/TCSVT.2015.2402892>
11. Wu Y, Wu C, Li B, Qiu X, Lau F. CloudMedia: When Cloud on Demand Meets Video on Demand. In: *Proc. of IEEE ICDCS*; 2011. p. 268–277.
12. Song X, Peng X, Xu J, Shi G, Wu F. Cloud-based distributed image coding. *IEEE Transactions on Circuits and Systems for Video Technology*. 2015; 25(12):1926–1940. <https://doi.org/10.1109/TCSVT.2015.2416562>
13. Wang F, Liu J, Chen M. CALMS: Cloud-assisted live media streaming for globalized demands with time/region diversities. In: *Proc. of IEEE INFOCOM*. IEEE; 2012. p. 199–207.
14. Wu Y, Zhang Z, Wu C, Li Z, Lau F. Cloudmov: cloud-based mobile social tv. *IEEE Transactions on Multimedia*. 2013; 15(4):821–832. <https://doi.org/10.1109/TMM.2013.2240670>
15. Zhao H, Zheng Q, Zhang W, Du B, Li H. A Segment-based Storage and Transcoding Trade-off Strategy for Multi-version VoD Systems in the Cloud. *IEEE Transactions on Multimedia*. 2017; PP(99):1–1.
16. Zhao H, Zheng Q, Zhang W, Wang J. Prediction-based and Locality-aware Task Scheduling for Parallelizing Video Transcoding over Heterogeneous MapReduce Cluster. *IEEE Transactions on Circuits and Systems for Video Technology*. 2016; PP(99):1–1.
17. Jennings B, Stadler R. Resource Management in Clouds: Survey and Research Challenges. *Journal of Network and Systems Management*. 2014; p. 1–53.
18. Meng X, Pappas V, Zhang L. Improving the scalability of data center networks with traffic-aware virtual machine placement. In: *Proc. of IEEE INFOCOM*. IEEE; 2010. p. 1–9.
19. Jiang JW, Lan T, Ha S, Chen M, Chiang M. Joint VM placement and routing for data center traffic engineering. In: *Proc. of IEEE INFOCOM*. IEEE; 2012. p. 2876–2880.
20. Cohen R, Lewin-Eytan L, Seffi Naor J, Raz D. Almost optimal virtual machine placement for traffic intense data centers. In: *Proc. of IEEE INFOCOM*. IEEE; 2013. p. 355–359.
21. Guo Y, Stolyar AL, Walid A. Shadow-routing based dynamic algorithms for virtual machine placement in a network cloud. In: *Proc. of IEEE INFOCOM*. IEEE; 2013. p. 620–628.
22. Li X, Wu J, Tang S, Lu S. Let's Stay Together: Towards Traffic Aware Virtual Machine Placement in Data Centers. In: *Proc. of IEEE INFOCOM*. IEEE; 2014. p. 1842–1850.

23. Zhang J, Ren F, Lin C. Delay Guaranteed Live Migration of Virtual Machines. In: Proc. of IEEE INFOCOM. IEEE; 2014. p. 574–582.
24. Liu H, He B. VMbuddies: coordinating live migration of multi-tier applications in cloud environments. *IEEE Transactions on Parallel and Distributed Systems*. 2015; 26(4):1192–1205. <https://doi.org/10.1109/TPDS.2014.2316152>
25. Zhang W, Chen Y, Gao X, Mo Z, Zheng Q, Lu Z. Cluster-Aware Virtual Machine Collaborative Migration in Media Cloud. *IEEE Transactions on Parallel and Distributed Systems*. 2017; PP(99):1–1.
26. Al-Fares M, Loukissas A, Vahdat A. A scalable, commodity data center network architecture. *ACM SIGCOMM Computer Communication Review*. 2008; 38(4):63–74. <https://doi.org/10.1145/1402946.1402967>
27. Perkins C, Westerlund M, Ott J. Web Real-Time Communication (WebRTC): Media Transport and Use of RTP. draft-ietf-rtcweb-rtp-usage-05. 2012;.
28. Amiri M, Al Osman H, Shirmohammadi S, Abdallah M. SDN-based game-aware network management for cloud gaming. In: Network and Systems Support for Games (NetGames), 2015 International Workshop on. IEEE; 2015. p. 1–6.
29. Huang CY, Chen KT, Chen DY, Hsu HJ, Hsu CH. GamingAnywhere: The first open source cloud gaming system. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*. 2014; 10(1s):10.
30. Aggarwal V, Gopalakrishnan V, Jana R, Ramakrishnan K, Vaishampayan V. Optimizing cloud resources for delivering IPTV services through virtualization. *IEEE Transactions on Multimedia*. 2013; 15(4):789–801. <https://doi.org/10.1109/TMM.2013.2240287>
31. Lathey A, Atrey PK. Image enhancement in encrypted domain over cloud. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*. 2015; 11(3):38.
32. Bari MF, Boutaba R, Esteves R, Granville LZ, Podlesny M, Rabbani MG, et al. Data center network virtualization: A survey. *IEEE Communications Surveys & Tutorials*. 2013; 15(2):909–928. <https://doi.org/10.1109/SURV.2012.090512.00043>
33. Lin G, Xue G. On the terminal Steiner tree problem. *Information Processing Letters*. 2002; 84(2):103–107. [https://doi.org/10.1016/S0020-0190\(02\)00227-2](https://doi.org/10.1016/S0020-0190(02)00227-2)
34. Calheiros RN, Ranjan R, Beloglazov A, De Rose CA, Buyya R. CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and experience*. 2011; 41(1):23–50.