High-elasticity Virtual Cluster Placement in Multi-tenant Cloud Data Centers

Shuaibing Lu[1,2], Jie Wu[2], and Zhiyi Fang[3]
[1]Faculty of Information Technology, Beijing University of Technology, Beijing, China
[2]Center for Networked Computing, Temple University, USA
[3]College of Computer Science and Technology, Jilin University Changchun, China
Email: lushuaibing11@163.com

***Abstract*—In recent years, Data Center Network (DCN) has become a promising and efficient data processing infrastructure for cloud computing. One important mission of DCN is to serve the ever-growing demand for computation, storage, and networking for multiple tenants in cloud computing. This paper uses the notion of elasticity to measure the potential growth of multiple tenants in terms of both computation and communication resources. Our objective is to maximize the elasticity for DCNs. We consider the multiple virtual cluster placement problem with the hose model under the computation and communication constraints. We first formulate this problem as an Integer Linear Programming (ILP) problem. Unfortunately, the formulated ILP problem cannot be solved by the simplex or eclipse methods because of a large number of variables and constraints. Therefore, we propose an efficient scheme based on the Dynamic Programming (DP) and analyze its optimality and complexity. Furthermore, we propose a heuristic algorithm for placement that maximizes the elasticity and guarantees the bandwidth demand as well as lower complexity. Extensive evaluations demonstrate that our schemes outperform existing state-of-the-art methods in terms of both elasticity and efficiency.**

***Index Terms*—Data Center Networks (DCNs), elasticity, multi-tenant, Virtual Machine (VM) placement.**

I. Introduction

In recent years, cloud computing offers a popular central platform for hardware and software services over the Internet. With the extensive growth of data volumes and varieties, Data Center Networks (DCNs) have become a promising and efficient data processing infrastructure for cloud computing. As reported in the public data of Azure[1], the deployment size of tenants is very bursty and unpredictable in terms of cores, memory, or bandwidth demands. The limited computation (CPU or memory) and communication (bandwidth) resources of the servers become the bottleneck when an increasing amount of tenants are employed in large-scale DCNs [2]. One important mission of the DCNs is to serve the ever-growing demand on computation, storage, and networking for multiple tenants in cloud computing.

To address these problems, we propose the use of elastic placement schemes to deal with the resource allocation for multiple tenants. For each tenant, we use one virtual cluster to represent its demands. One virtual cluster is an abstraction of a set of Virtual Machines (VMs) that connect to one virtual switch, which has both computing and communications

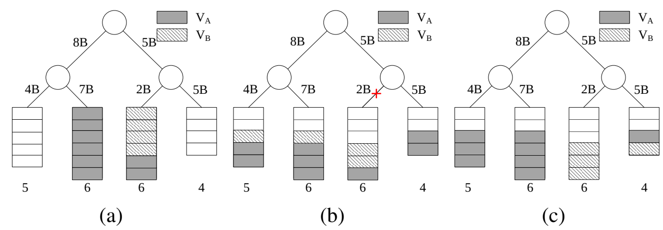


Fig. 1. An motivational example of the elastic virtual cluster placement.

resource requirements. In this paper, we consider the tree-structured DCNs using the hose model for communication, where each node has an aggregated performance guaranteed to the set of all other nodes [3, 4]. We use elasticity to measure the potential growth of virtual clusters in the DCNs, which is defined as the degree of a system that is able to adapt to the workload changes by provisioning and de-provisioning resources in an autonomic manner [3, 5]. In this paper, our objective is to maximize the elasticity during the placement process for the multiple virtual clusters while satisfying the constraints on computation and communication in the DCNs.

A. Motivation

We first look at an example that motivates our work in this paper. Some assumptions and notations are not explicitly stated, which will be explained in a later section. We consider a two-level network with four Physical Machines (PMs) as the DCN architecture. The capacity of each PM is slotted, and each slot can only host one VM. The capacity of each Physical Link (PL) is represented by the communication bandwidth, and each unit is denoted by B. Each VM has $1B$ total communication with other intra VMs of one virtual cluster. We use hose model as our communication model. As shown in Figure 1, we suppose that there are two virtual clusters, V_A and V_B, whose requests are 8 and 4 VMs, respectively. One extreme assignment for these two virtual clusters is to place them in the same set of PMs when possible by a greedy method in [6, 7] to minimize the resource consumption, which is shown in Figure 1(a). This solution can save more computation and communication resources; however, if one of V_A and V_B wants to scale its demand, there will be no other solution except migration or reconfiguration of VMs of V_A and V_B. Another extreme assignment is to place the VMs separately by using the method in [3] to maximize the elasticity for each virtual cluster, we have the following distribution which is shown in Figure 1(b). Even though this solution can obtain high elasticity, there will be congestion on the highlight link without bandwidth guarantee. Since the elasticity is the minimum one between the potential growth of computation and communication, the elasticities are 0 under these two solutions. It means that there is no growth for the virtual clusters in the DCN. Between the two extreme assignments, we prefer to propose an efficient scheme to maximize the elasticity under the constraints as shown in Figure 1(c), which can avoid the redistribution or congestion for the virtual cluster placement.

B. Contributions and Paper Organization

In this paper, we focus on the multiple virtual cluster placement in DCNs that maximizes the elasticity under both computation and communication constraints. Our contributions can be summarized as follows:

- We consider the placement problem for multiple virtual clusters with the hose model and show that there is a trade-off between elasticity and the resource consumption. We formulate it as an Integer Linear Programming (ILP) problem and suggest that it cannot be solved by the simplex method or the eclipse method based on the sizes of variables and constraints.
- We address the multiple virtual cluster placement problem for the hose model by maximizing the minimum elasticity using Dynamic Programming (DP). We analyze its optimality and complexity.
- In order to reduce the complexity, we propose a heuristic algorithm, which identifies an occupation with the proportion of maximum admissible VMs during the tree traversal.
- We present a few observations on tracing the public data of Microsoft Azure, and we conduct various evaluations with several state-of-the-art techniques on both simulation and real testbed. The results are shown from different perspectives to provide conclusions.

The remainder of this paper is organized as follows. Section II surveys related works. Section III describes the model and formulates the problem. Section IV investigates the problem by proposing two schemes that depend on using DP and heuristic methods. Section V presents the experiments. Finally, Section VI concludes the paper.

II. Related work

DCNs have recently received significant attention as a cost-effective infrastructure for storing large volumes of data and hosting large-scale service applications [8, 9]. Virtualization technology is used to ensure the flexibility of the provisioning of workloads in the DCNs. Several works have been done on VM placement, with various solutions proposed in [6, 10–17]. [6] proposes a virtual cluster abstraction Stochastic Virtual Cluster (SVC), and introduces a network sharing framework and efficient VM allocation algorithms to meet the bandwidth demands by tenants. However, these min-guarantees fail to consider the potential growth of a tenant's demand. [12, 13] show that there is a hard trade-off between min-guarantee and network proportionality, and they propose a set of properties to explicitly express the trade-off. [14] focuses on the utilities of resources, which contains load balancing and fairness by using a designed link establishment algorithm. [11, 15] study the congestion control in cloud DCN by using congestion-aware algorithms in the VM placement processes. [16, 17] propose redundant VM placement optimization approaches to enhancing the reliability of cloud services. These works all consider the performance guarantee for each tenant, but ignore the combinational relationship among multiple tenants.

In cloud computing, elasticity has been considered one of the central attributes [5, 18]. [19] designs a two-tier traffic-aware algorithm that efficiently solves the VM placement problem for guaranteeing the potential performance of the large problem size. [20] and [21] focus on dynamically adjusting the cluster size by considering bandwidth guarantee under the hose model by using online migration. However, it is hard to determine when an arbitrary VM has shown representative behavior and the VMs migration can cause high cost. [22] proposes a hierarchical VM placement algorithm on satisfying the growth of the VMs demands in semi-homogeneous DCN configuration under both limitations of PMs' and PLs' capacities. [23] analyzes the elastic management of cluster-based services in the cloud, which separates the resource provisioning from the service management and provides important benefits: elastic service capacity to adapt it to its dynamic workload. [24] realizes the elasticity by proposing ElasticSwitch which can be implemented in hypervisors to offer guaranteed allocations for resources. Most of these works focus on the virtual cluster placement problem by only considering the bandwidth guarantee, and they fail to consider the elasticity of the DCN infrastructure. Although some of these works consider both limitations of PMs' and PLs' capacities [23], the results can only adapt to the single tenant. In this paper, we consider the virtual cluster placement on the elastic scaling in multi-tenant cloud DCNs. Our objective is to maximize the elasticity for the DCNs by satisfying the requests of multi-tenants with the constraints of both computation and communication.

III. Model and Problem Formulation

The virtual cluster placement problem attempts to find an appropriate embedding in the DCNs for virtual clusters in order to satisfy the resource demands of different tenants. We use elasticity to measure the potential growth of the resource allocation, which is also an important factor for weighting the scalability of the DCNs. Our objective is to maximize the elasticity during the placement process for the multiple virtual clusters with satisfying the constraints on computation and communication in the DCNs.

A. DCN model

In this paper, we consider the tree-structured DCN as our physical topology, which is denoted by $\mathbf{G} = \{C, L\}$. C is the set of leaf nodes (PMs), which is denoted by $C = \{C_m\}$,

TABLE I
NOTATIONS

$\mathbf{G}$	A tree-structured DCN, where $G = \{C, L\}$.
C	The set of PMs, which is denoted by $C = \{C_m\}$.
C_m	m^{th} PM in the DCN $\mathbf{G}$.
c_m	Capacity of the m^{th} PM in the DCN
L	The set of PLs, which is denoted by $L = \{L_{ij}\}$.
L_{ij}	j^{th} PL on level i in the DCN
l_{ij}	Capacity of j^{th} PL on level i in the DCN
G_{ij}	Subtree under L_{ij}
V_w	w^{th} virtual cluster in the set V
N_w	Number of VMs in V_w
B_w	Bandwidth demand of each VM in V_w
U	A vector of the feasible occupation of set V.
$f(\cdot)$	Communication demand of the virtual cluster

and the capacity for each leaf node (PM) is c_m. We use S_{ij} to denote the j^{th} non-leaf node (switch) on the i^{th} level. L is the set of links (PLs), which is denoted by $L = \{L_{ij}\}$. L_{ij} denotes the j^{th} link on the i^{th} level in the DCN $\mathbf{G}$. The capacity for each link is l_{ij}, and we use G_{ij} to denote the subtree under the link L_{ij}.

B. Virtual Cluster

The virtual cluster is an abstraction that allows each tenant to specify both the VMs and per-VM bandwidth demand of its service [10]. Let $V = \{V_w\}$ denote the set of virtual clusters, and each virtual cluster consists of a set of VMs and one virtual switch. Let V_w denote the w^{th} virtual cluster in set V, where $V_w = \langle N_w, B_w \rangle$. We use $|V|$ to denote the number of virtual clusters in set V. N_w is the number of VMs in V_w, and B_w is the bandwidth demand between VMs and the virtual switch. Let N be the total number of VMs in set V, where $N = \sum_{w=1}^{|V|} |V_w|$. In this paper, we consider the virtual cluster abstraction based on the hose model [4]. In the hose model, each customer specifies a set of endpoints to be connected with a common endpoint-to-endpoint performance guarantee [3]. Let $f(\cdot)$ denote communication demand for virtual clusters, suppose that x VMs on one side and $N_w - x$ on another side, $f(V_w) = \min\{x, N_w - x\} \cdot B_w$. Each virtual cluster only communicates with the intra VMs, and there is no communication between inter virtual clusters.

C. Problem Formulation

In cloud computing, elasticity is defined as the degree to which a system is able to adapt to workload changes by provisioning and de-provisioning resources in an autonomic manner [5]. This paper uses the elasticity to measure the growth potential of virtual clusters in the DCN. Let E denote the elasticity of the DCN, we use the minimal remaining resource on each PM to denote the combinational elasticity, as shown in Equation 1. Our objective is to find a placement scheme for multiple virtual clusters that supports maximum elasticity (uniform growth) in both computation and communication without resorting to reassignment. Let c_m and c_m^* denote the maximum space and the used space of the m^{th} PM, respectively. Let l_{ij} and l_{ij}^* denote the maximum bandwidth and the used bandwidth of the j^{th} link on level i, respectively. Here, ν_w^m is a boolean variable which is used to indicate whether the VM of V_w belongs to the PM C_m. For each virtual request, the total number of VMs is $N_w = \sum_{m=0}^{n} \nu_w^m$. We have the following problem formulation:

$$\text{maximize} \quad E \tag{1}$$

$$\text{s.t.} \quad E = \min_{i,j}\{1 - \frac{c_m^*}{c_m}, 1 - \frac{l_{ij}^*}{l_{ij}}\} \tag{2}$$

$$c_i^* = \sum_{j=1}^{|V|} \nu_j^i \tag{3}$$

$$c_i^* \leq c_m \text{ and } c_m^* \in \mathbb{Z}^n \tag{4}$$

$$l_{ij}^* = \sum_{w=1}^{k} \{\min\{\sum_{C_m \in G_{ij}} \nu_w^m, N_w - \sum_{C_m \in G_{ij}} \nu_w^m\} \cdot B_w\} \tag{5}$$

$$l_{ij}^* \leq l_{ij} \tag{6}$$

Equations 1 and 2 show the objective of maximizing the elasticity. Equation 3 and Equation 4 are constraints on computation resource, which means the total number of VMs belonging the PM cannot exceed its capacity c_m. Equation 5 and Equation 6 are the constraints on the communication resource, which means the bandwidth consumption based on the hose model cannot exceed the link capacity l_{ij}.

IV. MULTIPLE VIRTUAL CLUSTER PLACEMENT

In this section, we address the multiple virtual cluster placement problem for the hose model to maximize the elasticity for the multi-tenant cloud DCN. To solve this problem, we first convert it to an Integer Linear Programming (ILP) formulation as shown in the Appendix. Since the numbers of variables and constraints are large, our problem cannot be efficiently solved by the simplex or eclipse methods. Therefore, we propose an efficient scheme based on the Dynamic Programming (DP) and analyze its optimality and complexity. Furthermore, we propose a heuristic algorithm that maximizes the elasticity and guarantees the bandwidth demand as well as lower complexity.

A. Dynamic Programming (DP) based Placement Scheme

In this subsection, we propose a solution using Dynamic Programming (DP) to find the optimal solution for the binary tree-structured DCN. The insight of our algorithm is to cut the DCN into two partitions level by level on each link, and to the calculation from bottom to up.

1) Maximize the elasticity: We use $OPT(\mathbf{G}, U)$ to denote the optimal occupation state for the set of virtual clusters V that maximizes the elasticity E of $\mathbf{G}$. Let U be a vector that contains all possible occupations of VMs for each virtual cluster, where $U = (u_1, u_2, ..., u_{|V|})$. The variable u_w in U is the number of allocated VMs of virtual cluster V_w, where $u_w \leq N_w$. As the example shown in Figure 1, there are two virtual clusters V_A and V_B with 8 and 4 VMs, respectively. Vector $U = (3, 2)$ denotes an occupation with 3 VMs of V_A and 2 VMs of V_B. We cut the tree-structured DCN with two partitions as shown in Figure 2, which are

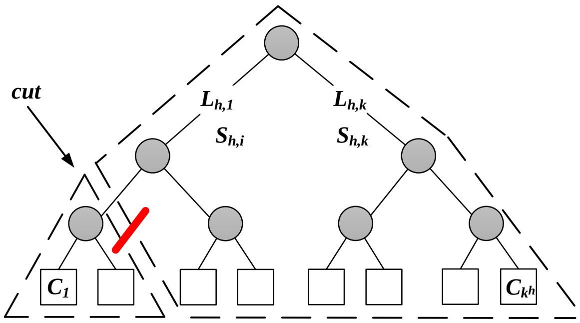


Fig. 2. The optimal substructure with DP of the DCN.

subtree $G_{h,k}$ and the subtrees excepting $G_{h,k}$ of $\mathbf{G}$, i.e., $\mathbf{G} \setminus G_{h,k}$. Let $OPT((G_{h,k}, U'))$ and $OPT(\mathbf{G} \setminus G_{h,k}, U - U')$ be the maximum elasticities under the subtrees $G_{h,k}$ and $\mathbf{G} \setminus G_{h,k}$ with the assignments of vectors U' and $U - U'$, respectively. Here, $\mathbf{G} \setminus G_{h,k}$ denotes the subtrees expecting $G_{h,k}$, and vector $U - U'$ denotes an assignment for the rest VMs. Under the allocation U, we use $\hat{C}$ to denote the PM which holds the minimum value of the elasticity, i.e., $E = E_{\hat{C}} = OPT(G, V)$. Here, $E_{\hat{C}}$ is the elasticity of PM $\hat{C}$. Obviously, we have $E_{\hat{C}} \leq OPT((G_{h,k}, U'))$, and $E_{\hat{C}} \leq OPT(\mathbf{G} \setminus G_{h,k}, U - U')$. The DP process can be recursively formulated to compute the optimal value as in Equation 7, which can be discussed under two cases of the optimal elasticity: (i). If $\hat{C} \in G_{h,k}$, we have $E = E_{\hat{C}} \leq OPT(G_{h,k}, U')$. Suppose that $E_{\hat{C}} < OPT(G_{h,k}, U')$, we allocate VMs of vector U' to the subtree $G_{h,k}$. We have that there is no PM with elasticity $E_{C_m \in G_{h,k}}$ smaller than $E_{\hat{C}}$ under the $G_{h,k}$. It is a contradiction with our assumption $\hat{C} \in G_{h,k}$. Thus, we must have $E_{\hat{C}} = OPT(G_{h,k}, U')$. (ii). If $\hat{C} \notin G_{h,k}$, we have $E = E_{\hat{C}} \leq OPT(\mathbf{G} \setminus G_{h,k}, U - U')$. We find that it the same as case (i), then we have $E_{\hat{C}} = OPT(\mathbf{G} \setminus G_{h,k}, U - U')$.

$$OPT(G, U) = \max_{u'_w \leq u_w, \forall w \leq N} \min\{OPT(G_{h,k}, U'), OPT(\mathbf{G} \setminus G_{h,k}, U - U')\} \quad (7)$$

Equation 7 shows the optimal substructure of the multiple virtual cluster placement problem. Therefore, given the optimal values of the subtrees, the optimal value $OPT(G, V)$ can be found by searching for optimal $G_{h,k}$. We can use DP to find the optimal allocation in a given tree, and we propose a DP-based scheme in Algorithm 1.

2) Algorithm Description: In this section, we present our virtual cluster placement scheme, which is based on DP for multi-tenants. We take the tree topology of the DCN $\mathbf{G}$ and the $|V|$ virtual clusters in set V as our inputs. The DCN occupation state for the set V is our output. The algorithm traverses the topology tree starting at the leaf nodes (PMs), and determines the number of VMs that occupied on $\mathbf{G}$ of $|V|$ virtual clusters through cutting the tree into two partitions level by level until the root. During the traversal process, for any visited link L at level l, the algorithm records the elasticities of subtrees and recursive recall on non-leaf nodes.

Algorithm 1 DP Algorithm

Input: DCN topology $\mathbf{G}$, set of virtual clusters V;
Output: DCN occupation state for set V;

1: **if** $N \geq \sum_{m \leq |C|} c_m$ **then**
2:    **return** False;
3: **for** each leaf node (PM) $C_m \in C$ **do**
4:    **for** each vector $U' \leq U$ **do**
5:       $OPT(C_m, U') = \min\{\frac{c_m - \sum_{u_w \in U'} u_w}{c_m}, \frac{l_{0,j} - f(\sum_{u_w \in U'} u_w)}{l_{0,j}}\}$;
6: **for** each cut $S_{k,l} \in \mathbf{G} \setminus C$ from bottom-up **do**
7:    **for** each vector $U' \leq U$ **do**
8:       $OPT(G_{k,l}, U') = \max_{U'' \leq U'}\{\min\{OPT_{i<k}(G_{i,j}, U''), OPT_{i<k}(G_{k,l} \setminus G_{i,j}, U' - U''), \frac{l_{i,j} - f(\sum_{u_w \in U'} u_w)}{l_{i,j}}\}\}$;
9: **return** $OPT(G, U)$;

In lines 1 and 2, we first do the feasible checking for the set of virtual clusters V that whether the total number of VMs N of set V is beyond the constraint of the PMs capacities. In lines 3 to 5, the algorithm starts to record the elasticities at leaf nodes level with consideration of every vector U' that can be occupied. $U' \leq U$ denotes that all values of U' are less than or equal to U, where $u'_w \leq u_w$ for all $w \leq |V|$. We use $OPT(C_m, U')$ to denote the maximum elasticity of C_m with the occupation U', which is equal to the smaller one between the PM elasticity $\frac{c_m - \sum_{u_w \in U'} u_w}{c_m}$ and the PL elasticity $\frac{l_{0,j} - f(\sum_{u_w \in U'} u_w)}{l_{0,j}}$, i.e., $OPT(C_m, U') = \min\{\frac{c_m - \sum_{u_w \in U'} u_w}{c_m}, \frac{l_{0,j} - f(\sum_{u_w \in U'} u_w)}{l_{0,j}}\}$. For each vector U', if the occupation of VMs based on U' satisfies the constraints of both computation and communication, the value of elasticity will be $E = OPT(C_m, U')$, otherwise, it will be $E = -\infty$. In lines 6 to 8, we record the elasticities for each cut upon non-leaf node $S_{k,l}$ that considers every vector U' from bottom-up, where $S_{k,l} \in G \setminus C$ and $U' \leq U$. Let $OPT(G_{k,l}, U')$ be the maximum elasticity of subtree $G_{k,l}$ with the occupation U', which is the occupation with maximum value in all cases of cuts of $G_{k,l}$. The cut of $G_{k,l}$ divides it into two parts, subtrees inside or outside $G_{i,j}$. For each cut, the elasticity is the smallest one of the PM elasticities of two partitions and the PL elasticity. The elasticities of two partitions $G_{i,j}$ and $G_{k,l} \setminus G_{i,j}$ are $OPT_{i<k}(G_{i,j}, U'')$ and $OPT_{i<k}(G_{k,l} \setminus G_{i,j}, U' - U'')$. Since $G_{i,j}$ is a subtree of $G_{k,l}$, we have $i < k$. The PL elasticity is $\frac{l_{i,j} - f(\sum_{u_w \in U'} u_w)}{l_{i,j}}$. Thus, for each case of cut, we have $\min\{OPT_{i<k}(G_{i,j}, U''), OPT_{i<k}(G_{k,l} \setminus G_{i,j}, U' - U''), \frac{l_{i,j} - f(\sum_{u_w \in U'} u_w)}{l_{i,j}}\}$. In line 9, we return the DCN occupation state U for the set of virtual clusters V with the maximum elasticity $OPT(\mathbf{G}, U)$.

We find the optimal allocation for $\mathbf{G}$ by maintaining the set of all possible occupations of vector U that can be allocated in each subtree. The total number of cuts during the DP is related to the number of links in $\mathbf{G}$, which is $\sum_{j=1}^{h-1} 2^j = 2^h$. For each partition, since the orientation is changed, we need to reassign

Algorithm 2 Multi-tenant Virtual Cluster Placement Scheme(MVCPS)

Input: DCN topology **G**, set of virtual clusters V;
Output: DCN occupation state for set V;

1: **if** $N \geq \sum_{m \leq |C|} c_m$ **then**
2: **return** False;
3: Calculate the maximum admissible VMs of **G**;
4: Select the node with the maximum admissible VMs be the new root;
5: **for** each level from root to leaf **do**
6: Hierarchical place the VMs based on the load proportions of each branch;
7: **while** $U \neq 0$ **do**
8: **for** each branch from left to right **do**
9: **for** each vector $U' \leq U$ **do**
10: **if** $E \neq -\infty$ under vector U' **then**
11: Place VMs as vector U' with minimum bandwidth demand by $\arg\min_{U' \leq U} f(U')$;
12: **else**
13: **repeat**
14: Adjust VMs in U' to right nearest branch;
15: **until** $E \neq -\infty$;
16: Place VMs as vector U' with minimum bandwidth demand by $\arg\min_{U' \leq U} f(U')$;
17: Update $U = U - U'$;
18: **return** DCN occupation state for set V;

the VMs for $h-j-1$ non-leaf nodes, where h is the height of the DCN. Since there are $|V|$ virtual clusters, the total number of possible combinations is $prod_{w=1}^{|V|}(u_w+1)^{h-j}$. Thus, the time complexity of Algorithm 1 is $O(2^h \cdot \prod_{w=1}^{|V|}(u_w+1)^{h-j})$.

B. Multi-tenant Virtual Cluster Placement Algorithm

For the Algorithm 1, the time complexity depends on the size of feasible vectors for the set of virtual clusters V, which can be reduced by some heuristic. In this subsection, we propose a heuristic algorithm for the multi-tenant virtual cluster placement. The insight of our heuristic is to identify a feasible occupation with the proportion of the maximum admissible loads (VMs) [3] during the tree traversal, and we greedily choose the placement scheme with increasing order of the bandwidth demand. The input and output in Algorithm 2 are the same with Algorithm 1. In lines 1 and 2, we do the same feasible checking for the set of users as Algorithm 1. In line 3, we calculate the maximum admissible VMs of **G** by using the linear algorithm in [3]. The insight of this linear algorithm is to calculate the loads from different orientations and choose the best one. In line 4, we select the node with the maximum admissible VMs to be the new root. In lines 5 to 17, we start to place the VMs from top-to-bottom hierarchically. We first allocate the VMs depending on the load proportions of each branch. For each branch from left to right, we search any vector U' that is smaller than U. If vector U' is feasible ($E \neq -\infty$), we place VMs as vector U' with minimum bandwidth demand by $\arg\min_{U' \leq U} f(U')$. Otherwise, we start to adjust the VMs in U' one by one to the right neighboring branch until we find a feasible vector for this branch. After that we place VMs as the reconstructed vector U' with minimum bandwidth demand by $\arg\min_{U' \leq U} f(U')$, and we update the vector with $U = U - U'$. In line 18, we return the DCN occupation state for the set of virtual clusters V. The time complexity of Algorithm 2 is $O(h \cdot N \cdot \prod_{w=1}^{|V|}(u_w+1)^2)$.

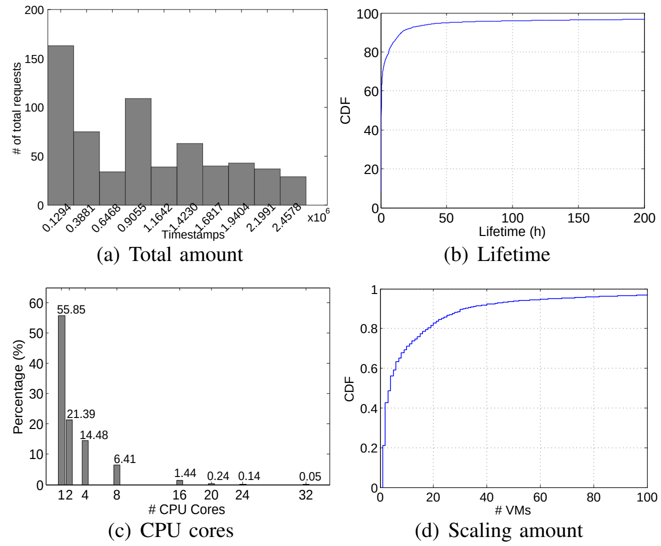


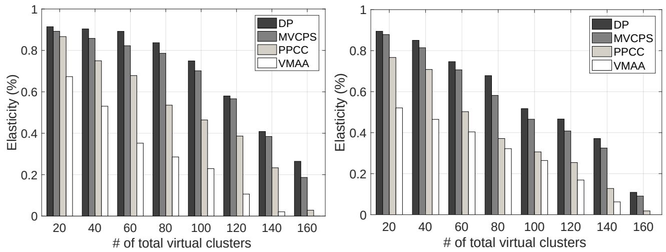
Fig. 3. Data tracing of requesting virtual clusters.

V. Evaluations

In this section, we conduct extensive simulations and experiments to study the elastic virtual cluster placement in multi-tenant DCNs. These experiments are conducted to evaluate the performances of the proposed algorithms on both simulation and real testbed. After presenting the datasets and settings, the results are shown from different perspectives to provide insightful conclusions.

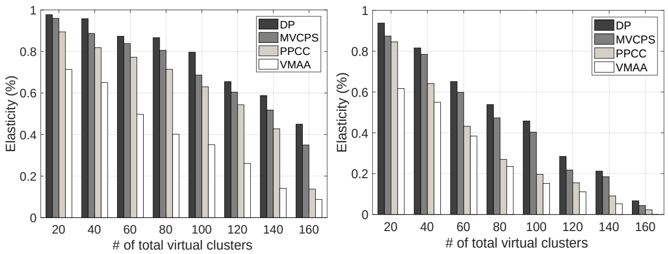
A. Real Data Analysis

We first present a few observations on analyzing the public data of Microsoft Azure [1] in Figure 3, including the deployments and the workload conditions. We first analyze the data of total requests from the first and third parties in one week, the distribution is shown in Figure 3(a). During the timestamps, the total number of requests is fluctuating within $[20, 160]$. The lifetimes of VMs from the same request are distributed in $[10, 50]$ hours (almost 95% of VMs are running less than 50 hours), as shown in Figure 3(b). It means that, once the requesting resources by the tenants are satisfied, they will not release within a short period of time. Then we analyze the data of CPU cores and the scales of virtual requests at an hourly granularity. As shown in Figure 3(c), it specifies that most VMs require few virtual CPU cores (almost 80% of virtual clusters require 1 or 2 cores). According to the tracing results, we find that users do not always deploy their VMs in one-time, and each deployment may grow (and shrink) over time before it is terminated. Almost 80% of virtual clusters



(a) # of virtual clusters from 0 to 10. (b) # of virtual clusters from 10 to 20.

Fig. 4. The elasticity for the DCN ($k = 6$).



(a) # of virtual clusters from 0 to 10. (b) # of virtual clusters from 10 to 20.

Fig. 5. The elasticity for the DCN ($k = 8$).

require scale 20 VMs [1], as shown in Figure 3(d). Based on the analysis of real data, we deploy the basic setting of our simulations.

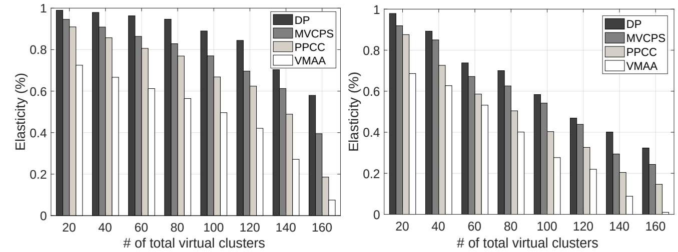
B. Basic Setting

We simulate a DCN of three-level k-ary tree topology, and each rack consists of k PMs ($k \in \{6, 8, 10\}$). The unit of each PM resource is slotted, which can be easily interpreted to a real configuration. The capacity of PMs ranges from 0 to 100 slots, and one slot can only hold one VM. Each PM has a $1Gbps$ PL to connect to a layer-2 switch, and every ten layer-2 switches are connected to a core switch with $100Gbps$. The unit bandwidth demand between per-pair of VMs is $B = 1Gbps$. We divide the dataset of virtual clusters into two different categories, $[0, 10]$ and $[10, 20]$, which depends on the requesting size of each virtual cluster. For each virtual cluster, it is generated in the ranges randomly. We use 20 as step length to divide our dataset into 7 groups, which ranges within $[20, 160]$. All the settings are based on the data tracing results in Figure 3. In addition to the proposed scheduling algorithms, two baseline algorithms are used, VM allocation Algorithm (VMAA) [6] and Proportion with Physical Combinational Capacities (PPCC) [3].

- VM allocation Algorithm (VMAA): virtual clusters are placed one by one, and each virtual cluster is placed so as to minimize the link resource consumption.
- Proportion with Physical Combinational Capacities (PPCC): virtual clusters are placed one by one, and each virtual cluster is placed to maximize the elasticity.

C. Experiment Results

Figures. 4, 5, and 6 present the elasticities for virtual clusters in which the ranging numbers of the VMs are divided into two groups $[0, 10]$ and $[10, 20]$, respectively. We use the same four algorithms (VMAA, PPCC, DP, MVCPS) on each group of data set and calculate the elasticities for the



(a) # of virtual clusters from 0 to 10. (b) # of virtual clusters from 10 to 20.

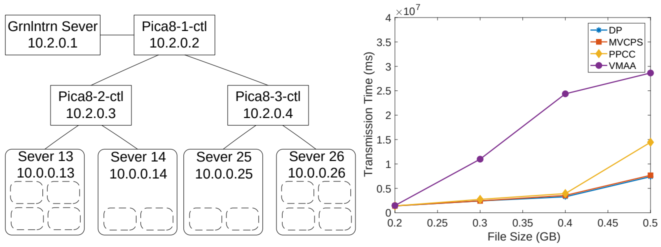
Fig. 6. The elasticity for the DCN ($k = 10$).

requests that ranging from $[20, 160]$. Additionally, we have the following observations: (i). With the higher amount of the total virtual requests for the DCN, the impact of the algorithms on the elasticities is greater. As shown in Figures. 4, 5, and 6, the elasticity for the DCNs which belong to the same group is decreasing with the scaling of amount of virtual clusters. The reason is that more virtual clusters demand more resources, which will lead to the increase in the combinational utilization of the clusters and thus lower the elasticity of the DCNs. From Figure 5 and Figure 6, since the VMAA focuses on the demand of communication resource, the performance of elasticity decreases significantly with the increase of the scaling of virtual clusters. For the PPCC, the elasticity of the DCN is not much different when the total amount of virtual clusters is small, as shown in Figure 6 (b). (ii). For the same amount of the total virtual requests, a larger size of the virtual cluster will lead to lower elasticity. As shown in Figure 6, we can see that the elasticity of the DCN for virtual clusters with scaling under the range $[0, 10]$ is much higher than $[10, 20]$. The distribution of virtual clusters' sizes is random, for small-scale DCN, DP can be deployed with the optimal solution. However, for large-scale DCN, MVCPS method is more efficient. (iii). When the scale of the DCN is larger, the impacts of algorithms on the elasticities are higher. The elasticity depends on the localities of VMs, which means a good placement scheme can support more virtual clusters in the larger DCN. Compared with sub-figures (a) and (b) of Figures. 4, 5, and 6, the elasticity has the same trend with the increasing amount of PMs in the DCN, which means more VMs can be supported by the providers.

In summary, compared with VMAA and PPCC, DP and MVCPS have better performances in elasticity across the multiple virtual cluster placement. DP can obtain the optimal solution for virtual clusters, however, the computing time is extremely high, when either the total amount or the sizes of virtual clusters are scaling. However, the complexity of the optimal solution is too large and the computing time will increase dramatically. Besides, we can observe that MVCPS is very close to the optimal value. MVCPS has lower computing complexity, and the average elasticity can reach 89.3% and 88.4% compared with the optimal solution under the ranges $[0, 10]$ and $[10, 20]$, respectively.

D. Evaluations on Testbed

1) Testbed Configuration: According to the simulation results, we deploy the realistic transmission experiments on



(a) (b)

Fig. 7. Data tracing of CPU cores and VMs scaling.

TABLE II
DEPLOYMENTS OF THE ALGORITHMS

Algorithms	Server 13	Server 14	Server 25	Server 26
VMAA	4A	1A	0	3B
PPCC	2A	0	1A,1B	2A,2B
DP	3B	1A	1A	3A
MVCPS	3A	1A	1A	3B

the real testbed of our lab, whose topology is shown in Figure 7 (a). The testbed contains two Cisco Catalyst 3560G switches (8 ports) and three Pica8 P-3297 switches (48 ports). Each Pica switch connects two 64 bits Dell Power Edge R210 servers, and each server has 2.4 GHz CPU and 4 GB memory. Each server can hold 4 VMs, and the maximum rate for per-VM is $0.25Gbps$ through virtual ports. All servers are accessible via the connections with Pica8 switches. $Grnlntrn$ is a controller connected to port 10 on the Cisco switch, which is constructed by a Dell Power Edge R210 server. The capacity of every physical link is $1Gbps$.

2) Evaluations results: In this subsection, we deploy our algorithms on the real testbed and focus on the capacities constraints which comes from the communication demand scaling in the DCN. We evaluate the performance of the DNC by monitoring the transmission time for files between VMs. In order to create a heterogeneous DCN topology, we do the VM allocation on each server through the controller $Grnlntrn$. For each link in the testbed, we use bandwidth control on the physical ports in the switches Pica8-1, Pica8-2, and Pica8-3. The topology after configuration is shown in Figure 7, and the capacities of the PMs are 4, 2, 2, and 4, respectively. We consider the condition with two virtual clusters (V_A and V_B), and the request demands for VMs are 5 and 3, respectively. The deployments of the VMs under the four algorithms are shown in Table II. According to Figure 7 (a), we set the capacities of the links between switched and servers to be $1Gbps$, $0.5Gbps$, $0.5Gbps$, and $1Gbps$. All upper links are $1Gbps$. For each group, we trace the transmission time of the files between servers 13, 14, 25, and 26. Servers 13 and 25 serve as the senders, and servers 14 and 26 serve as the receivers. The size of the file fluctuates from 0.2 GB to 0.5 GB in the unit of 0.1 GB, which denotes the scaling demand of the communication resource of the VMs.

The results appear in Figure 7 (b). As shown in Figure 7 (b), the transmission time increases with the scaling of the file size gradually. For each group, the performances of VMAA and PPCC are nearly the same when the file size is limited to $0.25Gbps$, which is the capacity constraint of the physical link. However, when the file size becomes larger, the transmission time of VMAA increases dramatically, as the purple line shown in Figure 7 (b). The performances of DP and MVCPS are very close, and the average minor difference of 4.02% is expected to come from the performances of server 13 and server 26.

VI. CONCLUSION

In this paper, we use elasticity to measure the potential growth of multiple tenants in terms of both computation and communication resources. Our objective is to maximize the elasticity for the Data Center Network (DCN). Specifically, we consider the multiple virtual cluster placement problem with the hose model under the constraints. We first formulate this problem as an Integer Linear Programming (ILP) problem, and demonstrate how it cannot be solved by the simplex or eclipse methods based on the large number of variables and constraints. We then propose an efficient scheme based on DP and analyze its optimality and complexity. Based on that we propose a heuristic algorithm, which can realize the multiple virtual cluster placement on maximizing the elasticity with bandwidth demands guarantee and lower complexity. Extensive evaluations demonstrate that our schemes outperform existing state-of-the-art methods in terms of both elasticity and efficiency.

ACKNOWLEDGEMENT

This work of the first author was done during her stay as a visitor scholar at Temple University. This research was supported in part by NSF grants CNS 1757533, CNS 1629746, CNS 1564128, CNS 1449860, CNS 1461932, CNS 1460971, IIP 1439672, and CSC 20163100.

REFERENCES

[1] E. Cortez, A. Bonde, A. Muzio, M. Russinovich, M. Fontoura, and R. Bianchini, "Resource central: Understanding and predicting workloads for improved resource management in large cloud platforms," in *Proceedings of the 26th Symposium on Operating Systems Principles*. ACM, 2017, pp. 153–167.

[2] W. Jiang, W. Jiang, W. Wang, H. Wang, Y. Pan, and J. Wang, "A fine-grained rule partition algorithm in cloud data centers," *Journal of Network and Computer Applications*, vol. 113, pp. 14–25, 2018.

[3] J. Wu, S. Lu, and H. Zheng, "On maximum elastic scheduling of virtual machines for cloud-based data center networks," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[4] N. G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. K. Ramakrishnan, and J. E. van der Merive, "A flexible model for resource management in virtual private networks," in *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 4, 1999, pp. 95–108.

[5] D. PLUMMER, D. SMITH, T. BITTMAN, D. CEAR-LEY, D. CAPPUCCIO, D. SCOTT, R. KUMAR, and B. ROBERTSON, "Study: Five refining attributes of public and private cloud computing," Tech. rep., Gartner, 2009. http://www. gartner. com/DisplayDocument, Tech. Rep.

[6] L. Yu and H. Shen, "Bandwidth guarantee under demand uncertainty in multi-tenant clouds," in *Distributed Computing Systems (ICDCS), IEEE 34th International Conference on*, 2014, pp. 258–267.

[7] L. Yu, H. Shen, Z. Cai, L. Liu, and C. Pu, "Towards bandwidth guarantee for virtual clusters under demand uncertainty in multi-tenant clouds," *IEEE Transactions on Parallel and Distributed Systems*, 2017.
[8] M. F. Bari, R. Boutaba, R. Esteves, L. Z. Granville, M. Podlesny, M. G. Rabbani, Q. Zhang, and M. F. Zhani, "Data center network virtualization: A survey," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 2, pp. 909–928, 2013.
[9] G. Ananthanarayanan, C. Douglas, R. Ramakrishnan, S. Rao, and I. Stoica, "True elasticity in multi-tenant data-intensive compute clusters," in *Proceedings of the Third ACM Symposium on Cloud Computing*, 2012, p. 24.
[10] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron, "Towards predictable datacenter networks," in *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 4. ACM, 2011, pp. 242–253.
[11] C. Tian, A. Munir, A. X. Liu, Y. Liu, Y. Li, J. Sun, F. Zhang, and G. Zhang, "Multi-tenant multi-objective bandwidth allocation in datacenters using stacked congestion control," in *2017 Proceedings IEEE INFOCOM*, 2017, pp. 1–9.
[12] L. Popa, G. Kumar, M. Chowdhury, A. Krishnamurthy, S. Ratnasamy, and I. Stoica, "Faircloud: sharing the network in cloud computing," in *ACM SIGCOMM Computer Communication Review*, 2012, pp. 187–198.
[13] A. Shieh, S. Kandula, A. G. Greenberg, C. Kim, and B. Saha, "Sharing the data center network." in *NSDI*, vol. 11, 2011, pp. 23–23.
[14] J. Duan and Y. Yang, "A load balancing and multi-tenancy oriented data center virtualization framework," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 8, pp. 2131–2144, 2017.
[15] F. Yan, T. T. Lee, and W. Hu, "Congestion-aware embedding of heterogeneous bandwidth virtual data centers with hose model abstraction," *IEEE/ACM Transactions on Networking (TON)*, vol. 25, no. 2, pp. 806–819, 2017.
[16] A. Zhou, S. Wang, B. Cheng, Z. Zheng, F. Yang, R. Chang, M. Lyu, and R. Buyya, "Cloud service reliability enhancement via virtual machine placement optimization," *IEEE Transactions on Services Computing*, 2016.
[17] S. Yang, P. Wieder, R. Yahyapour, S. Trajanovski, and X. Fu, "Reliable virtual machine placement and routing in clouds," *IEEE Transactions on Parallel and Distributed Systems*, 2017.
[18] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.
[19] X. Meng, V. Pappas, and L. Zhang, "Improving the scalability of data center networks with traffic-aware virtual machine placement," in *2010 Proceedings IEEE INFOCOM*, 2010, pp. 1–9.
[20] C. Fuerst, S. Schmid, L. Suresh, and P. Costa, "Kraken: Online and elastic resource reservations for multi-tenant datacenters," in *2016 Proceedings IEEE INFOCOM*, 2016, pp. 1–9.
[21] L. Yu and Z. Cai, "Dynamic scaling of virtual clusters with bandwidth guarantee in cloud datacenters," in *2016 Proceedings IEEE INFOCOM*, 2016, pp. 1–9.
[22] K. Li, J. Wu, and A. Blaisse, "Elasticity-aware virtual machine placement for cloud datacenters," in *Cloud Networking (CloudNet), IEEE 2nd International Conference on*, 2013, pp. 99–107.
[23] R. Moreno-Vozmediano, R. S. Montero, and I. M. Llorente, "Elastic management of cluster-based services in the cloud," in *Proceedings of the 1st workshop on Automated control for datacenters and clouds*. ACM, 2009, pp. 19–24.
[24] L. Popa, P. Yalagandula, S. Banerjee, J. C. Mogul, Y. Turner, and J. R. Santos, "Elasticswitch: Practical work-conserving bandwidth guarantees for cloud computing," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 351–362, 2013.
[25] Y. T. Lee, A. Sidford, and S. C. w. Wong, "A faster cutting plane method and its implications for combinatorial and convex optimization," in *Foundations of Computer Science (FOCS), IEEE 56th Annual Symposium on*, 2015, pp. 1049–1065.

APPENDIX

In the Appendix, we use a linear programming (LP) approach to maximize the elasticity for set V in $\mathbf{G}$. For each subtree G_{ij} under the link L_{ij}, let $\mathbb{V}_w^m = \sum_{C_m \in G_{ij}} \nu_w^m$ denote the total number of VMs in the w^{th} virtual cluster under the subtree G_{ij}. We convert Equation 6 to Equation 8.

$$l_{ij}^* = \sum_{w=1}^{k} \{\min\{\mathbb{V}_w^m, N_w - \mathbb{V}_w^m\} \cdot B_w\} \tag{8}$$

Here, we define a intermediate variable y_w^m, where

$$y_w^m = \min\{\mathbb{V}_w^m, N_w - \mathbb{V}_w^m\} \cdot B_w \tag{9}$$

Then Equation 8 will be transfer to

$$l_{ij}^* = \sum_{w=1}^{k} y_w^m \tag{10}$$

Since we have Equation 9, it can be transferred to

$$y_w^m \leq \mathbb{V}_w^m \cdot B_w \text{ and } y_w^m \leq (N_w - \mathbb{V}_w^m) \cdot B_w \tag{11}$$

Then Equation 8 will be transfer to Equation 12 by combing Equations 10 and 11

$$l_{ij}^* \leq \sum_{w=1}^{k} \mathbb{V}_w^m \cdot B_w \text{ and } l_{ij} \leq \sum_{w=1}^{k} (N_w - \mathbb{V}_w^m) \cdot B_w \tag{12}$$

Our formulation will be converted to a ILP formulation:

$$\text{maximize} \quad E \tag{13}$$

$$\text{s.t.} \quad E \leq 1 - \frac{c_m^*}{c_m} \text{ and } c_m^* \leq c_m \text{ for } \forall m \tag{14}$$

$$E \leq 1 - \frac{l_{ij}^*}{l_{ij}} \text{ and } l_{ij}^* \leq l_{ij} \text{ for } \forall i, \forall j \tag{15}$$

$$l_{ij}^* \leq l_{ij} \text{ for } \forall i, \forall j \tag{16}$$

$$l_{ij}^* \leq \sum_{w=1}^{k} \mathbb{V}_w^m \cdot B_w \tag{17}$$

$$\text{and } l_{ij}^* \leq \sum_{w=1}^{k} (N_w - \mathbb{V}_w^m) \cdot B_w \text{ for } \forall m, \forall w \tag{18}$$

Equation 13 is the same as Equation 1. Equation 14 converts Equation 2 by separating the minimum constraint of each PM. Similarly, Equation 17 and Equation 18 convert Equation 6 by separating the minimum constraint of each link. We suppose that the DCN is a full binary tree with n leaf nodes, and the number of links is $2n-2$. Equation 14 includes n constraints, Equation 16 includes $2n-2$ constraints, and Equation 17 includes $2k(2n-2)$ constraints. In total, the ILP formulation has kn variables, and $4(1+k)(n-1)+2$ constraints. It has $\Theta(kn)$ variables and $\Theta(kn)$ constraints, which means that our problem cannot be efficiently solved by the simplex or eclipse methods [25]. This kind of inefficiency motivates us to find other solutions for the elasticity maximization problem.