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


**Shuaibing Lu[1,2]** **Jie Wu[2]**

[1]Faculty of Information Technology, Beijing University of Technology, Beijing, China

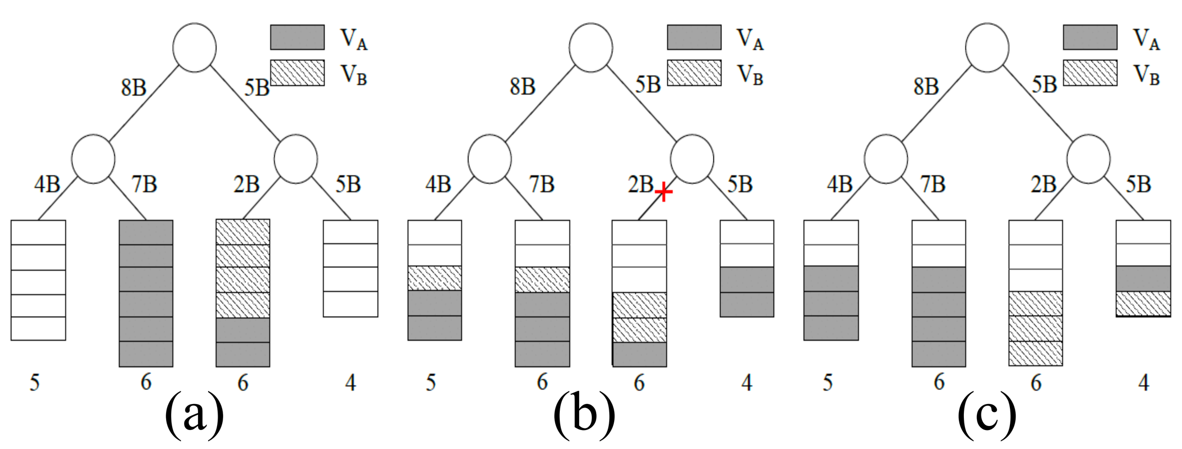
[2]Center for Networked Computing, Temple University, USA


## Introduction

**Data Center Networks (DCNs)** have become a promising and efficient data processing infrastructure for cloud computing.

- **Objectives:** Our objective is to maximize the elasticity during the placement process for the multiple virtual clusters (VCs) while satisfying the constraints on computation and communication in the DCNs.

## Motivation



- **Challenges:** Challenge comes from the trade-off between the physical machine and link elasticities, since a good allocation can achieve high elasticity. Allocate physical resource to VMs, and guarantee both computing and communication demand for multiple users to improve the utilization due to the limitation of resources and the variation of virtual requests.

## Problem Formulation

We attempt to find an appropriate embedding in the DCNs for virtual clusters in order to satisfy the resource demands of different tenants.

maximize $E$

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

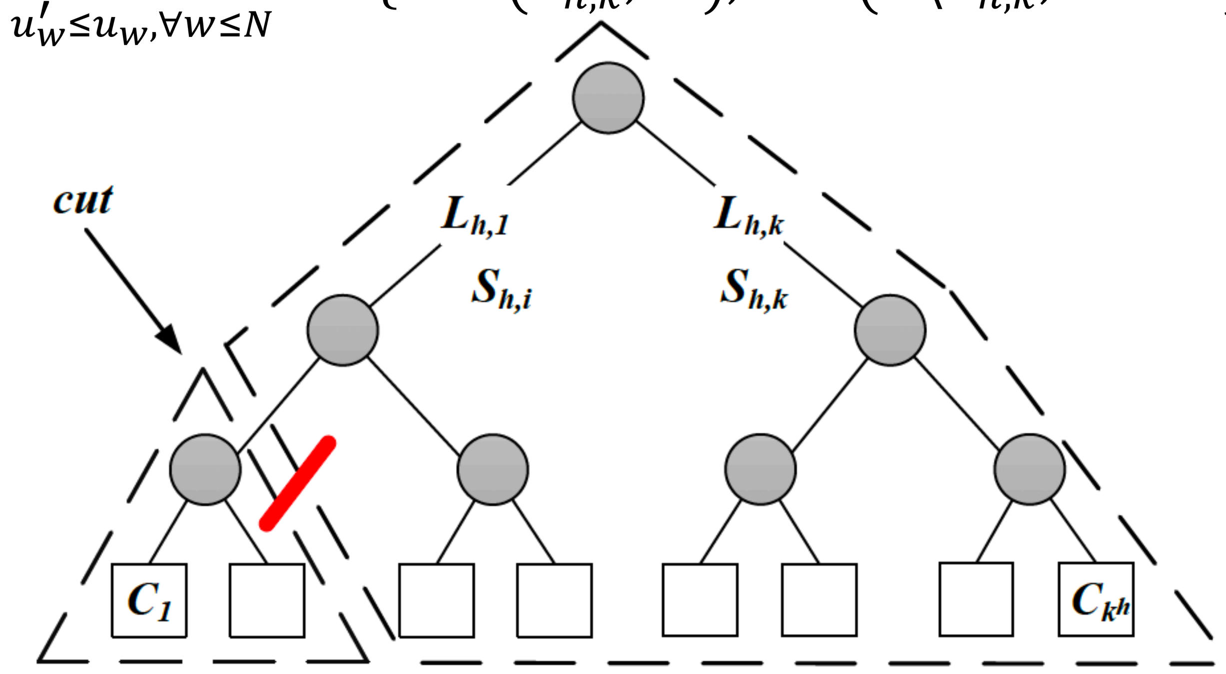
$c_i^* = \sum_{j=1}^{|V|} v_j^i,\ c_i^* \le c_m$ and $c_m^* \in \mathbb{Z}^n$

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

## Solutions

- **Dynamic Programming (DP) based Placement Scheme:** 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.
- **Optimal substructure**

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

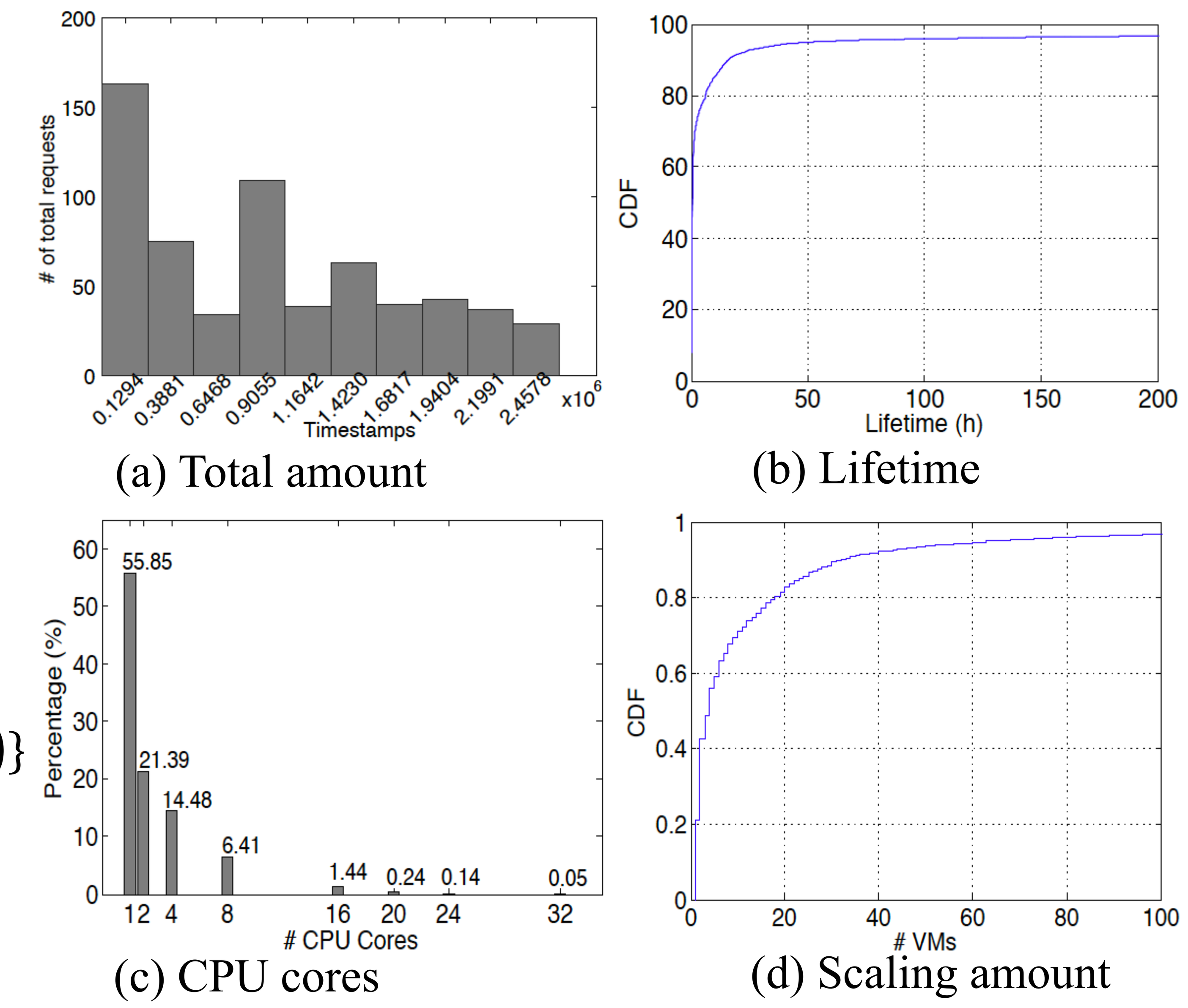



The optimal substructure with DP of the DCN.

- **Multi-tenant Virtual Cluster Placement Scheme:** The insight of our heuristic is to identify a feasible occupation with the proportion of the maximum admissible loads during the tree traversal, and we greedily choose the placement scheme with increasing order of the bandwidth demand.

## Evaluations

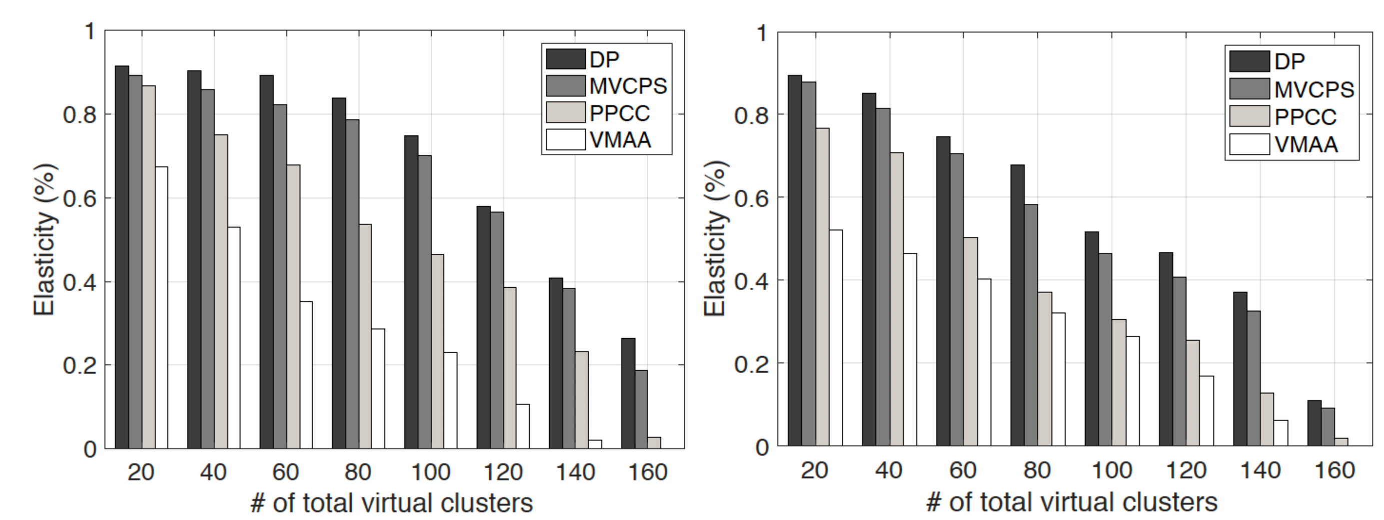
- **Real Data Analysis:** We first present a few observations on analyzing the public data of Microsoft Azure including the deployments and the workload conditions.



(a) Total amount (b) Lifetime (c) CPU cores (d) Scaling amount

- **Experiment Results**

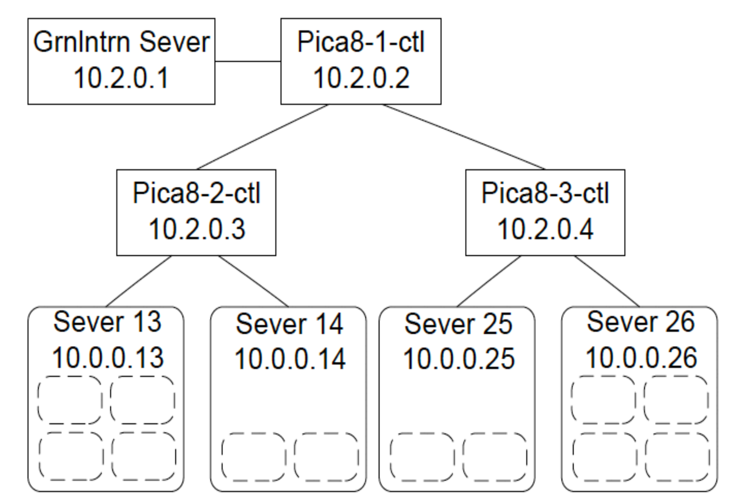
We use the same four algorithms (VMAA, PPCC, DP, MVCPS) on each group of data set and calculate the elasticities for the requests that ranging from [20,160].

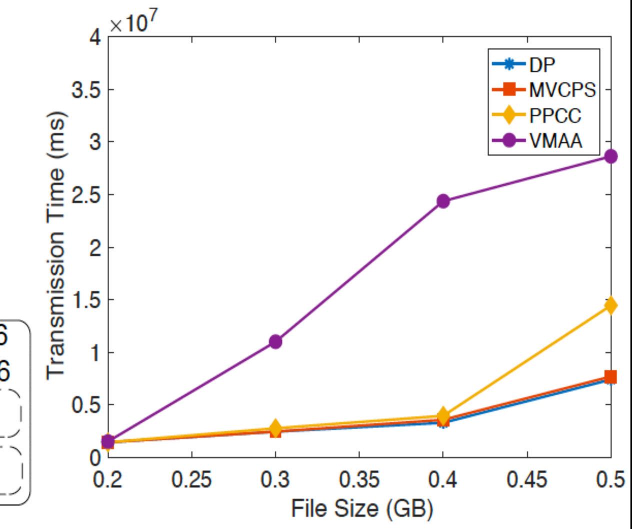


(a) # of VCs from 0 to 10 (k=6). (b) # of VCs from 10 to 20 (k=6).

- **Evaluations on Testbed:** we deploy the realistic transmission experiments on the real testbed of our lab.

| 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 |



Data tracing of CPU cores and VMs scaling.

## Conclusion

- 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 address the problem by maximizing the minimum elasticity using Dynamic Programming (DP). 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 on both simulation and real testbed. The results are shown from different perspectives to provide conclusions.