A Policy-Aware Virtual Machine Management in Data Center Networks

Lin Cui*, Fung Po Tso†, Dimitrios P. Pezaros‡, Weijia Jia§, Wei Zhao¶
*Department of Computer Science, Jinan University, Guangzhou, China
†School of Computing & Mathematical Science, Liverpool John Moores University, L3 3AF, UK
‡School of Computing Science, University of Glasgow, G12 8QQ, UK
§Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai China
¶Department of Computer and Information Science, University of Macau, Macau SAR, China
Email: tcuilin@jnu.edu.cn; p.tso@ljmu.ac.uk; dimitrios.pezaros@glasgow.ac.uk; weijiaj@gmail.com; weizhao@umac.mo

***Abstract*—**

Policies play an important role in network configuration and therefore in offering secure and high performance services especially over multi-tenant Cloud Data Center (DC) environments. At the same time, elastic resource provisioning through virtualization often disregards policy requirements, assuming that the policy implementation is handled by the underlying network infrastructure. In this paper, we define *PLAN*, a PoLicy-Aware virtual machine maNagement scheme to jointly consider DC communication cost reduction through Virtual Machine (VM) migration while meeting network policy requirements.

I. INTRODUCTION

Network policies demand traffic to traverse a sequence of specified middleboxes. As a result, network administrators are often required to manually install middleboxes in the data path of end points or significantly alter network partition and carefully craft routing in order to meet policy requirements. There is a consequent lack of flexibility that makes DC networks prone to misconfiguration, and it is no coincidence that there is emerging evidence demonstrating that up to 78% of DC downtime is caused by misconfiguration [1] [2].

On the other hand, migrating a VM from one server to another will inevitably alter the end-to-end traffic flow paths, requiring subsequent dynamic change or update of the affected policy requirements [3]. Clearly, change of the point of network attachment as a result of VM migrations substantially increases the risk of breaking predefined sequence of middlebox traversals and lead to violations of policy requirements. It has been demonstrated in PACE [4] that deployment of applications in Cloud DC without considering network policies may lead to up to 91% policy violations.

In this paper, we explore the policy-aware VMs migration problem, and present an efficient PoLicy-Aware VM maNagement (*PLAN*) scheme, which, (a) adheres to policy requirements, and (b) reduces network-wide communication cost in DC networks. The communication cost is defined with respect to policies associated to each VM. In order to attain both goals, we model the *utility* (i.e., the reduction ratio of communication cost) of VM migration under middlebox traversal requirements and aim to maximize it during each migration decision. To the best of our knowledge, this is the first joint study on policy-aware performance optimization through elastic VM management in DC networks.

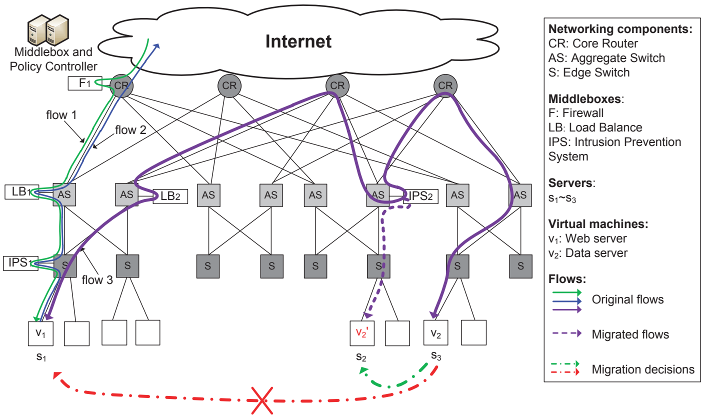


Fig. 1: Flows traversing different sequences of middleboxes in DC networks. Without policy-awareness, v_2 will be migrated to s_1, resulting in longer paths for flow 1 and wasting network resources.

II. POLICY-AWARE VM ALLOCATION MODELING

We consider a multi-tier DC network which is typically structured under a multi-root tree topology (e.g., fat-tree [5]).For a group of middleboxes $MB = \{mb_1, mb_2, \ldots\}$, there are various deployment points in DC networks. The centralized *Middlebox Controller*, see Fig. 1, monitors the liveness of middleboxes and informs the switches regarding the addition or failure/removal of a middlebox.

For a set of policies $\mathbb{P} = \{p_1, p_2, \ldots\}$, each policy p_i is defined in the form of $\{flow \rightarrow sequence\}$. $flow$ is represented by a 5-tuple: $\{src_{ip}, dst_{ip}, src_{port}, dst_{port}, proto\}$. $sequence$ is a list of middleboxes that all flows matching policy p_i should traverse them in order: $p_i.sequence = \{mb_1^i, mb_2^i, \ldots\}$. Denote p_i^{in} and p_i^{out} to be the first (ingress) and last (egress) middleboxes respectively in $p_i.sequence$. Let $\mathbb{V} = \{v_1, v_2, \ldots\}$ be the set of VMs in the DC network hosted by the set of servers $\mathbb{S} = \{s_1, s_2, \ldots\}$. Let $\lambda_k(v_i, v_j)$ denote the *traffic load* (or rate) in data per time unit exchanged between VM v_i and v_j (from v_i to v_j) following policy p_k.

This work is supported by the "Fundamental Research Funds for the Central Universities" No.21614330 and NSFC No. 61402200.

Because not all DC links are equal, and their cost depends on the particular layer they interconnect. Considering the investment cost, "lower cost" switch links are more preferable. Let c_i denote the *link weight* for l_i. Hence, the *Communication Cost* of all traffic from VM v_i to v_j is defined as: $C(v_i, v_j) = \sum_{p_k \in P(v_i,v_j)} \lambda_k(v_i, v_j) \sum_{l_s \in L_k(v_i,v_j)} c_s = \sum_{p_k \in P(v_i,v_j)} (C_k(v_i, p_k^{in}) + C_k(p_k^{in}, p_k^{out}) + C_k(p_k^{out}, v_j))$, where $L_k(v_i, v_j)$ is the routing path between v_i and v_j, $C_k(v_i, p_k^{in}) = \lambda_k(v_i, v_j) \sum\limits_{l_s \in L(v_i, p_k^{in})} c_s$ is the communication cost between v_i and p_k^{in} for flows which matched p_k. Similarly, $C_k(p_k^{out}, v_j)$ is the communication cost between p_k^{out} and v_j for p_k, and $C_k(p_k^{in}, p_k^{out})$ is the communication cost between p_k^{in} and p_k^{out}, which can be ignored as it makes no contribution to the minimization of the communication cost.

The vector R_i denotes the physical resource requirements of VM v_i, e.g., CPU cycles, memory size. The amount of physical resource provisioning by host server s_j is given by a vector H_j. We denote A to be an allocation of all VMs. $A(v_i)$ is the server which hosts v_i in A, and $A(s_j)$ is the set of VMs hosted by s_j. Considering a migration for VM v_i from its current allocated server $A(v_i)$ to another server $\hat{s}$: $A(v_i) \rightarrow \hat{s}$, the feasible space of candidate servers for v_i is: $\mathcal{S}_i = \{\hat{s} | (\sum_{v_k \in A(\hat{s})} R_k + R_i) \preceq H_j \; \hat{s} \in \mathbb{S}\}$

Let $\mathcal{C}_i(s_j)$, where $s_j = A(v_i)$ be the total communication cost induced by v_i between s_j and all ingress/egress middleboxes related to v_i: $\mathcal{C}_i(s_j) = \sum_{p_k \in P(v_i,*)} C_k(v_i, p_k^{in}) + \sum_{p_k \in P(*,v_i)} C_k(v_i, p_k^{out})$.

Migrating a VM also generates network traffic between the source and destination hosts of the migration, as it involves copying the in-memory state and the content of CPU registers between the hypervisors. The amount of migration traffic $C_m(v_i)$ can be obtained from [6]. We then consider the *utility* in terms of the expected benefit (of migrating a VM to a server) minus the expected cost incurred by such operation:

$$U(A(v_i) \rightarrow \hat{s}) = \mathcal{C}_i(A(v_i)) - \mathcal{C}_i(\hat{s}) - C_m(v_i) \qquad (1)$$

The *total utility* $\mathcal{U}_{A \rightarrow \hat{A}}$ is the summation of *utilities* for all migrated VMs from allocation A to $\hat{A}$.

The *PoLicy-Aware VM maNagement (*PLAN*)* problem:

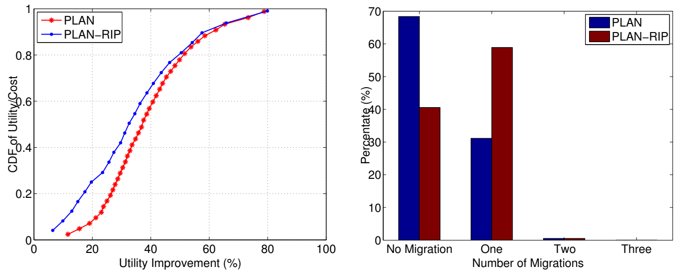
Definition 1. *Given the set of VMs* $\mathbb{V}$*, servers* $\mathbb{S}$*, policies* $\mathbb{P}$*, and an initial allocation* A*, we need to find a new allocation* $\hat{A}$ *that maximizes the total utility:*

$$\begin{aligned} &\max \mathcal{U}_{A \rightarrow \hat{A}} \\ &s.t. \; \mathcal{U}_{A \rightarrow \hat{A}} > 0 \\ &\qquad \hat{A}(v_i) \in S_i, \forall v_i \in \mathbb{V} \end{aligned} \qquad (2)$$

It can be easily proved that *PLAN* is NP-Hard, by reducing from the Multiple Knapsack Problem (MKP).

III. Policy-Aware Migration Algorithms

We design a decentralized heuristic scheme to perform policy-aware VMs migration. Server hypervisors will monitor all traffic load for each collocated VM v_i. A migration decision phase will be triggered periodically during which v_i will compute the appropriate destination server $\hat{s}$ for migration. The migration request is initialized by VMs and hybpervisors will decide whether to accept according to their residual resources. If no migration is needed, $U(A(v_i) \rightarrow \hat{s}) = 0$. Otherwise, the total *utility* is increased after migration when $A(v_i) \neq \hat{s}$.



(a) CDF: ratio of utility to communication cost

(b) Number of migrations before convergence

Fig. 2: Performance of PLAN

Policy-aware initial placement of VMs is also critical for new VMs in DC networks. Initially, predefined application-specific policies should be known for v_i. Since the VM has just been initialized, its traffic load might not be available. However, we can still choose the best server to host v_i by considering traffic of all policies for v_i equally.

IV. Evaluation

We have implemented *PLAN* in ns-3 and evaluated it under a fat-tree DC topology. We have also simulated *PLAN* without using the initial placement algorithm (which is referred to as *PLAN* with Random Initial Placement or *PLAN-RIP* in the sequel). Fig. 2a depicts the improvement of individual VM's communication cost after each migration through calculating the ratio of *utility* to the communication cost of that VM before migration. It can be observed that each migration can reduce communication cost by 39.06% on average for *PLAN* and 34.19% for *PLAN-RIP*, respectively. Fig. 2b shows the number of migrations per VM as *PLAN* converges. In *PLAN*, as a result of initial placement, only 30% of VMs need to migrate only once to achieve stable state throughout the whole experimental run. Nevertheless, in both schemes, there are very few ($< 1\%$) VMs need to migrate twice and no VM needs to migrate 3 times or more.

References

[1] J. Sherry, S. Hasan, C. Scott, A. Krishnamurthy, S. Ratnasamy, and V. Sekar, "Making middleboxes someone else's problem: Network processing as a cloud service," *ACM SIGCOMM Computer Communication Review*, vol. 42, no. 4, pp. 13–24, 2012.

[2] D. A. Joseph, A. Tavakoli, and I. Stoica, "A policy-aware switching layer for data centers," in *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4. ACM, 2008, pp. 51–62.

[3] S. Sivakumar, G. Yingjie, and M. Shore, "A framework and problem statement for flow-associated middlebox state migration," 2012.

[4] L. E. Li, V. Liaghat, H. Zhao, M. Hajiaghayi, D. Li, G. Wilfong, Y. R. Yang, and C. Guo, "PACE: Policy-aware application cloud embedding," in *Proceedings of 32nd IEEE INFOCOM*, 2013.

[5] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4. ACM, 2008, pp. 63–74.

[6] V. Mann, A. Gupta, P. Dutta, A. Vishnoi, P. Bhattacharya, R. Poddar, and A. Iyer, "Remedy: Network-aware steady state vm management for data centers," in *NETWORKING 2012*. Springer, 2012, pp. 190–204.