Networklet: Concept and Deployment

Sheng Zhang†, Yu Liang◇, Zhuzhong Qian†, Mingjun Xiao§, Jie Wu‡, Fanyu Kong♯, and Sanglu Lu†
†State Key Lab. for Novel Software Technology, Nanjing University, P.R. China, {sheng,qzz,sanglu}@nju.edu.cn
◇Nanjing University of Posts and Telecommunications, P.R. China, lyu611@163.com
§School of Computer Science and Technology / Suzhou Institute for Advanced Study,
University of Science and Technology of China, P.R. China, xiaomj@ustc.edu.cn
‡Department of Computer and Information Sciences, Temple University, jiewu@temple.edu
♯Ant Financial, P.R. China, njukongfy@gmail.com

***Abstract*—In today's datacenters, resource requests from tenants are increasingly transforming into hybrid requests that may simultaneously demand IaaS, PaaS, and SaaS resources. This paper tackles the challenge of modeling and deploying hybrid tenant requests in datacenters, for which we coin "networklet" to represent a set of VMs that collaboratively provide a PaaS or SaaS service. Through extracting networklets from tenant requests and thus sharing them between tenants, we can achieve a win-win situation for datacenter providers and tenants.**

I. Introduction

Today's public datacenters (*e.g.*, Amazon EC2, and Microsoft Azure) focus on computation-oriented resource reservation [2, 7], which only allows tenants to specify computing and memory demands, but ignores networking resources. Although simple, this model results in highly unpredictable performance of tenants virtual machines (VMs) [3]. To provide performance guarantee, prior works [3–5, 7, 8] have proposed several novel abstractions that allow tenants to explicitly specify networking as well as computing demands. However, most of them fit comfortably under one of two headings: hose [3, 5, 7], or clique [4, 8]. In the hose model, all tenant VMs are connected to a common virtual switch by links of homogeneous or heterogenous capacities; while in the clique model, tenants can specify bandwidth requirements between all pairs of VMs.

We find that these two types of abstractions represent two extremes in the design space, and we want to propose a new abstraction model that may help datacenter providers cater to resource allocation in hybrid datacenters.

In fact, the lines between Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) are becoming blurred as datacenter providers seek to create cloud platforms that can satisfy the needs of enterprises and widen their appeal to developers. For example, IaaS providers are trying to cater to application development (*e.g.*, AWS Elastic Beanstalk provides PaaS-like development layers [1]). With this kind of hybrid datacenter, resource requests from tenants are increasingly transforming into hybrid requests that may simultaneously demand IaaS, PaaS, and SaaS resources.

Motivated by these observations, this paper tackles the challenge of modeling and deploying hybrid tenant resource requests in hybrid datacenter networks (DCNs). A hybrid tenant request can be seen as a set of IaaS VMs, the bandwidth requirements between them, and a set of PaaS or SaaS services it may access. We coin the word *networklet* to represent a set of VMs that collaboratively provide some PaaS or SaaS service. As we know, tenants usually do not access or occupy a PaaS or SaaS service throughout the duration of the request; thus, it is reasonable to share networklets among multiple tenants. As long as the sum of the demands for a networklet from multiple tenants does not exceed the service capacity of a networklet, we can guarantee predictable performance of tenant applications.

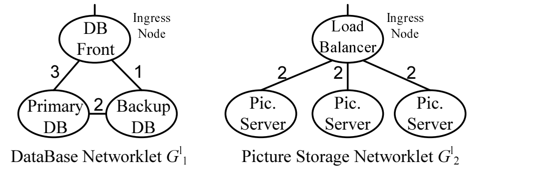


Fig. 1: Two networklet examples that represent database and picture storage services, respectively.

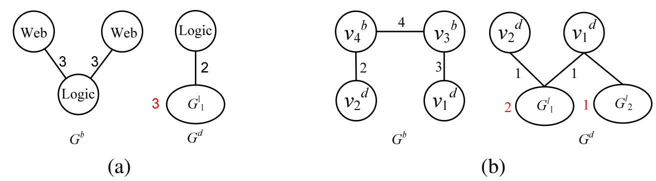


Fig. 2: HTR examples, where G^l_1 and G^l_2 denote the database and picture storage networklets, respectively, in Fig. 1. Note that, G^d is a bipartite graph specifying the connections between IaaS VMs and networklets.

II. Networklet and Hybrid Tenant Request

Fig. 1 shows two networklet examples: database networklet—a PaaS service, and picture storage networklet—a SaaS service. A networklet is denoted by an undirected graph, where vertices represent VMs and edges represent bandwidth requirements between them. For ease of exposition, we choose to abstract away details of the non-network resources as in previous work [3, 7]; thus, each VM requires a fixed VM slot in DCN servers. Every networklet has an ingress node which is the first node accessed by tenant requests. Each networklet has a service capacity, indicating how much workload it can handle in time. When an IaaS datacenter specifies L types of networklets, resource requests from tenants could be hybrid.

A hybrid tenant request (HTR) contains a set of IaaS VMs, the bandwidth requirements between them, and a set of networklets it access. Fig. 2 shows two examples.

Advantages. Table I summarizes the comparison results of hose, clique, and HTR+networklet on three design dimensions. Overall, our HTR+networklet model not only closely resembles the physical topologies used by datacenter tenants, but also improves physical resource utilization.

TABLE I: Comparison between the hose, clique, and HTR+networklet models

Abstraction	Tenant Cost	Provider Revenue	Provider Flexibility
VC/VOC [3]	High	Low	High
VDC [6]/VN [8]	Medium	Medium	Low
HTR+Networklet	Low	High	Medium

III. Deployment

Overview. We concentrate on the online version of the deployment problem. HTRs arrive one by one over time, we want to design an algorithm to allocate resources for an HTR.

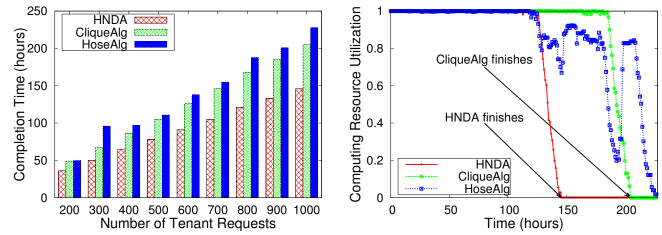
The ultimate goal of deployment is to maximize provider revenue while guaranteeing tenant application performance. As we mentioned before, this goal is equivalent to maximizing physical resource utilization, since DCN usually charges a tenant based on the amount of resources reserved for it. Maximizing resource utilization is further reduced to conserving physical resource consumption.

To conserve physical resource consumption, we have two sorts of strategies. The first one is to share networklets among multiple tenants, after observing a tenant usually does not occupy networklets throughout the duration of the HTR. As long as the sum of the demands for a networklet from multiple tenants does not exceed the service capacity of a networklet, we can guarantee predictable performance of tenant applications. The second one is to place VMs that have large bandwidth requirements between them as close as possible, so as to reduce the bandwidth consumption in underlying DCNs.

Evaluation. We compare the proposed algorithm, *i.e.*, HNDA, with the following two algorithms: *HoseAlg*, which converts a HTR+networklet request into a hose request and then deploys it using the allocation algorithm devised in [3], and *CliqueAlg*, which converts a HTR+networklet request into a clique request and then deploys it using the algorithm proposed in this paper.

Fig. 3(a) shows the completion time achieved by three algorithms versus the number of requests. Note that, the completion time is the time to complete all tenant requests, not the running time of each deployment algorithm. In general, we find that, in all settings, the completion time of all requests using HNDA is the smallest among them, while using HoseAlg has the largest completion time. This observation is consistent with the perceptual comparison results in Table I.

Fig. 3(b) shows the computing resource utilization over time when the number of tenant requests is 1000. The completion times of these 1000 requests using HNDA, CliqueAlg, and HoseAlg are 146, 205, and 228, respectively. After HNDA or



(a) Completion time (b) Resource utilization

Fig. 3: Simulation results

CliqueAlg finishes the batch of tenant requests, we assume their computing resource utilization is zero. We plot the computing resource utilizations at the beginning of each hour. We see that, the utilization ratio of HNDA is always higher than the other two algorithms before it finishes the batch of requests; CliqueAlg achieves a better resource utilization than HoseAlg, which is especially clear from the 120th hour to the 145th hour in the figure.

In summary, the proposed model and deployment algorithm achieve shorter completion time and better resource utilization than the hose or clique-based algorithm. We admit that the above-presented results are far from exhaustive; however, we hope these results can provide insights into modeling and deploying hybrid tenant resource requests and thus open a new avenue for resource allocation in multi-tenant datacenters.

IV. Conclusions

In this paper, we introduce networklet, a notion that helps to explore the tradeoff between performance guarantee and resource utilization in hybrid datacenters. Sharing networklets between multiple tenants achieves better multiplexing which benefits both tenants and providers.

Acknowledgments This work was supported in part by NSFC (61502224, 61472181, 61472185, 61321491), China Postdoctoral Science Foundation (2015M570434, 2016T90444), CCF-Tencent Open Research Fund (AGR20160104), Jiangsu NSF (BK20151392, BK20151390), and Collaborative Innovation Center of Novel Software Technology and Industrialization.

References

[1] "AWS Elastic Beanstalk," http://aws.amazon.com/elasticbeanstalk.
[2] 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," *CACM*, vol. 53, no. 4, pp. 50–58, 2010.
[3] H. Ballani, P. Costa, T. Karagiannis, and A. I. Rowstron, "Towards predictable datacenter networks," in *Proc. ACM SIGCOMM 2011*, pp. 242–253.
[4] M. Chowdhury, M. Rahman, and R. Boutaba, "ViNEYard: Virtual network embedding algorithms with coordinated node and link mapping," *IEEE/ACM TON*, vol. 20, no. 1, pp. 206–219, 2012.
[5] J. Duan, Z. Guo, and Y. Yang, "Cost efficient and performance guaranteed virtual network embedding in multicast Fat-Tree DCNs," in *Proc. IEEE INFOCOM 2015*, pp. 136–144.
[6] C. Guo, G. Lu, H. J. Wang, S. Yang, C. Kong, P. Sun, W. Wu, and Y. Zhang, "Secondnet: a data center network virtualization architecture with bandwidth guarantees," in *Proc. ACM CoNEXT 2010*, pp. 1–12.
[7] D. Xie, N. Ding, Y. C. Hu, and R. Kompella, "The only constant is change: incorporating time-varying network reservations in data centers," in *Proc. ACM SIGCOMM 2012*, pp. 199–210.
[8] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: substrate support for path splitting and migration," *ACM SIGCOMM CCR*, vol. 38, no. 2, pp. 17–29, 2008.