An Availability-aware SFC placement Algorithm for Fat-Tree Data Centers

Ghada Moualla, Thierry Turletti, Damien Saucez
Université Côte d'Azur, Inria, France

June 19, 2018

1 Summary

Network Function Virtualization (NFV) [1] virtualizes network functions and places them into commodity network hardware, such as a Data Center (DC). Then, multiple VNFs are combined together in a specific order, called Service Function Chains (SFCs) [2] to provide a complete service. SFCs determine the sequence of NFs that packets must follow and placement algorithms are used to map the SFCs request in the network. However, placing SFCs request in multi-tenant data center where the tenant demands are oblivious to the actual physical infrastructure of the Data Center is particularly challenging as the demand is not known in advance and cannot be controlled. For the data center operator, it is therefore important to not overload its network, yet respecting the level of service agreed with its tenants.

Multiple works have tackled the problem of robust VM placement by deploying them on different physical nodes using specified availability constraints [3, 4, 5]. However, none of these works consider the benefits of using redundancy to ensure reliability. In Herker et al. work [6], SFC requests are mapped to the physical network to build a primary chain, and backup chains are decided based on that primary chain. Engelmann et al. [7] propose to split service flows into multiple parallel smaller sub-flows sharing the load and providing only one backup flow for reliability guarantee. Our work follows the same principle as these two proposition [6, 7] but uses an active-active approach such that resources are not wasted for backup and different placement strategy is considered.

Exploiting the replication mechanisms, we propose an availability-oriented placement algorithm for SFCs in Data Centers relying on Fat-Tree topologies. The algorithm is run by the network hypervisor and guarantees that Service Level Agreements (SLAs) with the tenants are respected, given the availability properties of the hardware deployed in the data center.

Our proposition is based on an iterative linear program that solves the placement of SFCs in an online manner without prior knowledge on placement demand distribution. The algorithm starts with one replica of this SFC and tries to place it in the network while maximizing the availability of this placement based on the linear program. Thereafter, it checks if the placement availability of the achieved solution is less than what its agreed in the SLA, the algorithm replicates the SFC and adjusts the resources requirements based on the number of SFC replicas and availability of each replicas. To make the isolation between replicas, the algorithm tries to place each SFC replica in a different fault domain by leveraging the properties of Fat-Tree topology. This approach is repeated until the SLA is satisfied or the maximum allowed time to find a solution is exceeded.

To evaluate our proposed algorithm, we have implemented a discrete event simulator in Python interfaced with the Gurobi Optimizer 8.0 solver [8].[1] All simulations have been run on a Intel i7-4800MQ CPU at 2.70GHz and 32GB of RAM running GNU/Linux Fedora core 21. In the evaluation, SFCs requests are linear and independent. Our synthetic workload for the simulations contains 2,000 SFC request arrivals made of 20 random SFCs with a size chosen uniformly between 2 and 5 NFs. Evaluations are performed on a 48-Fat-Tree topology for acceptance ratio while increasing the tenants required availability in the SLA (see Figure 1) and its replication level

[1] All the data and scripts used in this paper are available on https://team.inria.fr/diana/robstdc/.

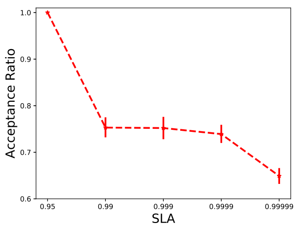

Figure 1: Comparing acceptance ratio for these different SLA values: 0.95, 0.99, 0.999, 0.9999, 0.99999.

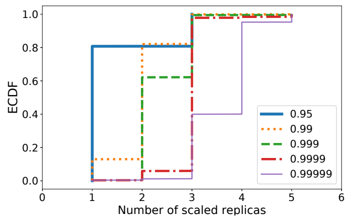

Figure 2: The ECDF for the number of created replicas with 5 different SLA with 48-Fat-Tree topology, $T_{IA} = 0.01$ and $S = 100$.

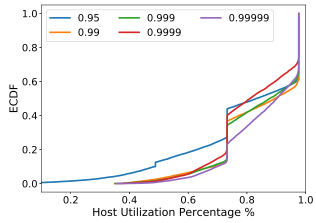

Figure 3: The ECDF for the host core utilization for different SLAs with 48-Fat-Tree topology, $T_{IA} = 0.01$ and $S = 100$.

(Figure 2). The results show that, as the required availability increases the acceptance ratio decreases and the number of required replicas increases. However even for high required availability (0.99999) it was possible to place with at most 5 replicas. We have also studied the servers cores utilization (Figure 3) and we notice that more than 40% of the servers are fully occupied, in all SLA scenarios. However, when the required availability is (0.99999), more than 80% of servers are more than 80% occupied.

Our evaluation on a very large simulated network topology shows that the algorithm is fast enough for being used in production environments and is able to satisfy as many demands as possible by spreading the load between the replicas while improving the network servers CPU utilization at the same time.

References

[1] ETSI, "Network Function Virtualisation (NFV); Architectural Framework," *NFV 001*, 2013.

[2] J. M. Halpern and C. Pignataro, "Service Function Chaining (SFC) Architecture," 2015.

[3] M. Mihailescu, A. Rodriguez, and C. Amza, "Enhancing application robustness in Infrastructure-as-a-Service clouds," in *IEEE/IFIP DSN Conference*, pp. 146–151.

[4] D. Jayasinghe, C. Pu *et al.*, "Improving performance and availability of services hosted on IAAS clouds with structural constraint-aware virtual machine placement," in *IEEE SCC Conference*, 2011.

[5] Q. Zhang *et al.*, "Venice: Reliable virtual data center embedding in clouds," in *IEEE INFOCOM*, 2014.

[6] S. Herker *et al.*, "Data-center architecture impacts on virtualized network functions service chain embedding with high availability requirements," 2015.

[7] A. Engelmann *et al.*, "A reliability study of parallelized vnf chaining," *arXiv preprint arXiv:1711.08417*, 2017.

[8] G. Optimization, "Gurobi optimizer 5.0," *Gurobi: http://www.gurobi.com*, 2013.