This work has been accepted for publication in 2018 IEEE Global Communications Conference (GLOBECOM 2018). Copyright © 2018 by IEEE

Trade-offs in Dynamic Resource Allocation in Network Function Virtualization

Stefan Schneider, Sevil Dräxler, Holger Karl
Paderborn University, Paderborn, Germany
{stefan.schneider, sevil.draexler, holger.karl}@uni-paderborn.de

***Abstract*—Dynamic allocation of resources is a key feature in network function virtualization (NFV), enabling flexible adjustment of slices and contained network services to ever-changing service demands. Considering resource allocation across the entire network, many authors have proposed approaches to optimize the placement and chaining of virtual network function (VNF) instances and the allocation of resources to these VNF instances. In doing so, various optimization objectives are conceivable, e.g., minimizing certain required resources or the end-to-end delay of the placed services.**

In this paper, we investigate the relationship between four typical optimization objectives when coordinating the placement and resource allocation of chained VNF instances. We observe an interesting trade-off between minimizing the overhead of starting/stopping VNF instances and all other objectives when adapting to changed service demands.

I. INTRODUCTION

In network function virtualization (NFV), the demand for network services, consisting of interconnected virtual network functions (VNFs), constantly changes as new users arrive and others depart. Depending on the demand for a service and the involved VNFs, also the required resources of these VNF instances constantly change. Hence, static assignment of resources to VNFs is not ideal since it either requires considerable over-provisioning of resources to handle possible peaks or it risks a shortage of resources during these peaks, leading to either high costs or to high delays and degraded service quality, respectively.

Since both options are undesirable, dynamic resource allocation presents a promising alternative. Here, resources are allocated to VNF instances on demand. To handle increasing service demands, VNFs are scaled out and additional instances are started, e.g., on servers of nearby Points of Presence (PoPs). Furthermore, existing VNF instances may be scaled up such that more resources are allocated to them, enabling them to process higher loads per instance. Vice versa, with decreasing service demands unnecessary VNF instances or unneeded resources can be removed to avoid wasting resources and energy.

In recent years, many authors have tackled the problem of coordinating resource allocation to network services and their VNFs according to given service demands [1]. The proposed optimization problems and heuristics optimize the placement and chaining of VNF instances in an underlying substrate network with limited resource capacities. In doing so, it is important to readjust placed network services to changed demands, ideally taking the current placement into account, rather than completely recomputing a new placement from scratch [2]. In addition to mere placement of VNF instances, some approaches also consider their dynamic scaling, i.e., deciding the number of VNF instances and their allocated resources [3], [4], [5]. Since scaling and placement are interdependent, it is beneficial to decide both jointly in a single step [4].

In practice, such optimization approaches run inside a management and orchestration (MANO) system, where the MANO executes the decisions of the optimization approach, handling the deployment and coordination of network services on the underlying NFV infrastructure. Incoming service requests can then be scheduled and assigned to the deployed network services using available scheduling approaches [6].

In the context of network slices, this optimization may be performed separately within each slice based on the given slice resources and the distinct optimization goals corresponding to the slice. For example, in some contexts (e.g., connected cars or augmented reality) short delay is of utmost importance while in others energy efficiency is more important. However, typically even within one slice there are multiple objectives that should be optimized, possibly according to some prioritization.

Given the complexity of the problem and the various possible optimization objectives, the relation between these objectives is non-trivial. However, when deciding which objectives to optimize it is crucial to understand their relationship and whether they can be optimized jointly or whether there are conflicts and trade-offs between some of the objectives.

As *main contribution* in this paper, we provide initial insights in the relation between common optimization objectives in dynamic resource allocation. Specifically, we perform a Pareto analysis to investigate the relationship of four different objectives using an optimization approach from our previous work [5], which supports joint scaling and placement of network services (Sec. II). The results of this Pareto analysis illustrate that most objectives can be optimized jointly but are conflicting with the minimization of operational overhead when starting and stopping VNF instances to readjust an embedding (Sec. III).

II. DYNAMIC RESOURCE ALLOCATION APPROACH

We use the optimization approach from our previous work [5], called B-JointSP, to dynamically scale, place and interconnect VNF instances of network services as well as optimize their allocated resources. In this section, we provide a short overview of the problem definition, the optimization approach, and the four objectives that are then used for the Pareto analysis in Sec. III. We picked B-JointSP as an example supporting optimization of multiple objectives, but we expect similar outcomes of the Pareto analysis when using a different optimization approach with the same objectives.

A. Problem definition

The goal of B-JointSP is to jointly scale and place network services, i.e., deciding in a single step the number of VNF instances, their interconnections, their location in the substrate network, and their allocated resources according to the following three inputs.

1) The underlying *substrate network* is modeled as a directed graph with nodes and links. The nodes can either just forward traffic or represent PoPs with certain processing capacities (here: CPU and memory). The links represent network links with certain delays, depending on the distance of the interconnected nodes, and specific data rate capacities.

2) The *network services* that need to be embedded in the substrate network are described by flexible service templates. These templates define the involved VNF components and the general service structure without specifying a fixed number of VNF instances or a fixed resource consumption. Instead, for each VNF the resource requirements are specified as a function of the traversing traffic volume that it has to handle, i.e., relative to the load. Similarly, the outgoing data rate at each VNF depends on the data rates of ingoing flows and the characteristics of the VNF, e.g., whether it compresses or augments traversing traffic. The VNFs of a network service can be connected as simple chain or in more complex service structures with branches, merges and even loops.

3) *Sources* represent users or sensors, for example, and can be located at any nodes in the network. Each source has outgoing flows for a specific network service and with a specific data rate.

Given these inputs, B-JointSP computes an embedding of the network service(s) on the substrate network. In doing so, each VNF is instantiated at least once, placed at a substrate node, and connected with the other VNFs of the service (Fig. 1). The number and location depends on the topology of the substrate network, the available resources, the source locations and the data rates of their flows. If a previous embedding of the network services already exists, it is taken into account and simply readjusted to the changed requirements.

B. Optimization objectives

There are four minimization objectives in B-JointSP:

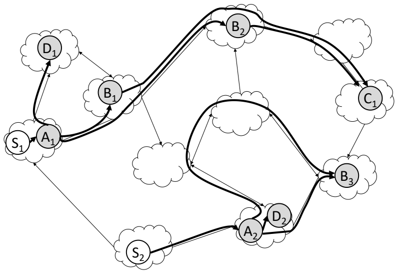


Fig. 1. Example embedding of a generic network service with sources S and VNFs A, B, C, and D.

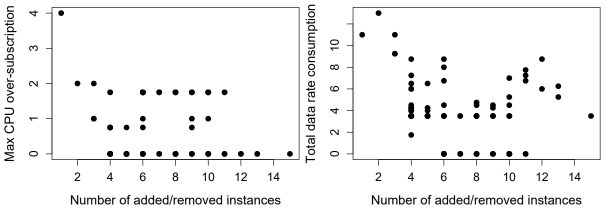
- Objective 1: Min. total resource over-subscription (CPU, memory, and data rate)
- Objective 2: Min. number of added or removed VNF instances compared to a previous embedding
- Objective 3: Min. total resource consumption (CPU, memory, and data rate)
- Objective 4: Min. total delay

In scenarios with very high load or very limited resources, it may not always be feasible to satisfy all resource requirements such that some resources may be over-subscribed. Objective 1 aims at avoiding or at least controlling and minimizing such over-subscription to avoid violations of service-level agreements (SLAs). Objective 2 focuses on minimizing the overhead of starting, stopping, or migrating VNF instances when adjusting an embedding to ongoing changes in the network. Objective 3 tries to minimize the allocated resources (both CPU and memory at the nodes as well as allocated data rate on the links) to reduce costs and energy consumption. Finally, objective 4 minimizes delay to improve the service quality. B-JointSP supports minimizing each of the objectives individually, which we use for the Pareto analysis in Sec. III.

III. PARETO ANALYSIS

Finding solutions that are good with respect to all four objectives can be non-trivial or impossible due to conflicts between the objectives. To investigate such conflicts, we performed a Pareto analysis. In this Pareto analysis, several independent computations are performed in which each objective is optimized individually while the other three non-optimized objectives are bounded by four different bounds each. For example, in one computation, objective 1 may be minimized while objectives 2 to 4 are bounded by upper bounds. In preliminary range estimation runs without any bounds, we observed the range of each objective (i.e., highest and lowest value) and then derived sensible bounds within these ranges. Overall, all 256 combinations of the different objectives and bounds are executed.

In doing so, we can observe the impact of the different objectives on each other. For example, is it possible to find solutions with low resource consumption (objective 3) while also choosing small bounds for the other objectives? Or do such small bounds of objectives 1, 2, 4



(a) Max. CPU over-subscription (b) Total allocated data rate

Fig. 2. Few added/removed instances (objective 2) conflicts with other objectives.

prevent the optimization approach from finding solutions with low resource consumption (objective 3)?

A. Evaluation setup

We obtain optimal solutions using the B-JointSP optimization problem [5], where the MILP was solved using the Gurobi Optimizer v 7.0.2, running on machines with Intel Xeon E5-2695 v3 CPUs running at 2.30 GHz.

Our evaluation is based on the Abilene data set [7], representing a real backbone network. Due to the long runtime of the MIP, for comparing the performance of the MIP and the heuristic, we only considered the western half of the network, consisting of 6 nodes and 14 directed links with uniform capacities. We calculated the link delay $d(l)$ for each link l based on the distances between the geographical locations of the nodes.

Based on common NFV use cases, we chose a bidirectional video streaming network service to be embedded, in which users request videos from a cache or server. Before streaming videos from the cache, they are transcoded by a video optimizer, reducing their data rate.

Using the Abilene network and the video streaming template, we created an initial embedding with one source. Using this initial embedding as starting point for the Pareto analysis, the embedding had to be readjusted as a new source appeared in the network.

B. Analysis results

Analyzing the results, we observe one main conflict between minimizing the number of added/removed instances (objective 2) and the remaining three objectives. With fewer added/removed instances, more existing instances have to be reused by the flow from the new source, leading to higher load per instance. Fig. 2a shows that with fewer added instances, as expected, they have to deal with higher load and the maximum CPU over-subscription increases (minimized in objective 1). Furthermore, reusing existing instances rather than migrating them or placing new ones close to the sources leads to longer paths in between. Routing flows via longer paths increases the total allocated data rate (Fig. 2b) and delay, which are minimized in objective 3 and objective 4, respectively.

Therefore, being able to add or remove instances with low overhead (which is in line with the goals of NFV) is crucial for optimizing the remaining three objectives. While no solution is optimal with respect to all four objectives, there are solutions that present a good trade-off, having low values for all four objectives. Hence, B-JointSP can be used to find high-quality embeddings with low over-subscription, resource consumption, and delay while keeping the required changes of existing embeddings at a minimum.

IV. CONCLUSION

In this paper, we presented early research findings on the trade-offs between typical objectives of dynamic resource allocation. The findings of the Pareto analysis suggest that over-subscription of resources, the total resource consumption, as well as total delay can be minimized jointly. However, minimizing these objectives can require significant adjustments in an already existing embedding, i.e., starting and stopping many VNF instances. Minimizing this overhead of adjusting existing embeddings conflicts with the other objectives.

Being aware of this trade-off is crucial as optimizing only one of the objectives, e.g., resource consumption, may lead to exceptional overhead every time an embedding needs to be adapted to new sources or other changes. Furthermore, this finding emphasizes the importance of low overhead when starting and stopping VNF instances to support full flexibility and optimal resource allocation in NFV scenarios.

In future research, we intend to further investigate this trade-off through more extensive evaluations with different scenarios, optimization approaches, and objectives.

ACKNOWLEDGMENTS

This paper has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No. 761493 (5GTANGO), and from the German Research Foundation (DFG) within the Collaborative Research Centre "On-The-Fly Computing" (SFB 901).

REFERENCES

[1] J. G. Herrera and J. F. Botero, "Resource allocation in nfv: A comprehensive survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 518–532, 2016.

[2] J. Liu, W. Lu, F. Zhou, P. Lu, and Z. Zhu, "On dynamic service function chain deployment and readjustment," *IEEE Transactions on Network and Service Management*, vol. 14, no. 3, pp. 543–553, 2017.

[3] M. Dieye, S. Ahvar, J. Sahoo, E. Ahvar, R. Glitho, H. Elbiaze, and N. Crespi, "Cpvnf: Cost-efficient proactive VNF placement and chaining for value-added services in content delivery networks," *IEEE Transactions on Network and Service Management*, vol. 15, no. 2, pp. 774–786, 2018.

[4] S. Dräxler, H. Karl, and Z. A. Mann, "Joint optimization of scaling and placement of virtual network services," in *IEEE/ACM CCGrid*, 2017.

[5] S. Dräxler, S. Schneider, and H. Karl, "Scaling and placing bidirectional services with stateful virtual and physical network functions," in *2018 IEEE Conference on Network Softwarization (NetSoft)*, 2018.

[6] L. Qu, C. Assi, and K. Shaban, "Delay-aware scheduling and resource optimization with network function virtualization," *IEEE Transactions on Communications*, vol. 64, no. 9, pp. 3746–3758, 2016.

[7] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, "SNDlib 1.0 – Survivable Network Design Library," in *ENOG INOC*, 2007.