Accepted Manuscript

Optimal Virtual Network Function Placement in Multi-Cloud Service Function Chaining Architecture

Deval Bhamare , Mohammed Samaka , Aiman Erbad , Raj Jain ,
Lav Gupta , H. Anthony Chan

PII: S0140-3664(17)30190-1
DOI: 10.1016/j.comcom.2017.02.011
Reference: COMCOM 5468

To appear in: *Computer Communications*

Received date: 31 July 2016
Revised date: 9 February 2017
Accepted date: 14 February 2017

Please cite this article as: Deval Bhamare , Mohammed Samaka , Aiman Erbad , Raj Jain , Lav Gupta , H. Anthony Chan , Optimal Virtual Network Function Placement in Multi-Cloud Service Function Chaining Architecture, *Computer Communications* (2017), doi: 10.1016/j.comcom.2017.02.011

Optimal Virtual Network Function Placement in Multi-Cloud Service Function Chaining Architecture

Deval Bhamare[Ψ], Mohammed Samaka[Ψ], Aiman Erbad[Ψ], Raj Jain[¥], Lav Gupta[¥], H. Anthony Chan[€]
[Ψ]Department of Computer Science and Engineering, Qatar University, Doha, Qatar
[¥]Dept. of Computer Sci. and Engineering,Washington University in St. Louis, St. Louis, MO 63130
[€]Huawei R&D USA, Plano, TX 75024
{devalb, samaka.m, aerbad}@qu.edu.qa, {jain, lavgupta}@wustl.edu, h.anthony.chan@huawei.com

Abstract— Service Function Chaining (SFC) is the problem of deploying various network service instances over geographically distributed data centers and providing inter-connectivity among them. The goal is to enable the network traffic to flow smoothly through the underlying network, resulting in an optimal quality of experience to the end-users. Proper chaining of network functions leads to optimal utilization of distributed resources. This has been a de-facto model in the telecom industry with network functions deployed over underlying hardware. Though this model has served the telecom industry well so far, it has been adapted mostly to suit the static behavior of network services and service demands due to the deployment of the services directly over physical resources. This results in network ossification with larger delays to the end-users, especially with the data-centric model in which the computational resources are moving closer to end users. A novel networking paradigm, Network Function Virtualization (NFV), meets the user demands dynamically and reduces operational expenses (OpEx) and capital expenditures (CapEx), by implementing network functions in the software layer known as virtual network functions (VNFs). VNFs are then interconnected to form a complete end-to-end service, also known as service function chains (SFCs). In this work, we study the problem of deploying service function chains over network function virtualized architecture. Specifically, we study virtual network function placement problem for the optimal SFC formation across geographically distributed clouds. We set up the problem of minimizing inter-cloud traffic and response time in a multi-cloud scenario as an ILP optimization problem, along with important constraints such as total deployment costs and service level agreements (SLAs). We consider link delays and computational delays in our model. The link queues are modeled as M/D/1 (single server/Poisson arrival/deterministic service times) and server queues as M/M/1 (single server/Poisson arrival/exponential service times) based on the statistical analysis. In addition, we present a novel affinity-based approach (ABA) to solve the problem for larger networks. We provide a performance comparison between the proposed heuristic and simple greedy approach (SGA) used in the state-of-the-art systems. Greedy approach has already been widely studied in the literature for the VM placement problem. Especially we compare our proposed heuristic with a greedy approach using first-fit decreasing (FFD) method. By observing the results, we conclude that the affinity-based approach for placing the service functions in the network produces better results compared against the simple greedy (FFD) approach in terms of both, total delays and total resource cost. We observe that with a little compromise (gap of less than 10% of the optimal) in the solution quality (total delays and cost), affinity-based heuristic can solve the larger problem more quickly than ILP.

Index Terms— affinity; greedy; Multi-cloud; Network function virtualization; Optimal placement; Service function chaining.

1. INTRODUCTION

Lately there has been an exponential growth in user data traffic due to the explosion of mobile devices and the emergence of novel networking paradigms such as Internet of Things (IoT). The unprecedented increase in data traffic has resulted in excessive CapEx and OpEx for the Internet service providers (ISPs) and application service providers (ASPs) [61]. The networks built with proprietary hardware devices are complex, difficult to debug, and expensive to cater to increased demands and new emerging complex services. In addition, recently, services are moving from host-centric to data-centric model in which the computational resources are moving closer to end users. As a result, application service providers (ASPs) and ISPs are increasingly using virtualization technologies to deploy network functions over the standard high-volume infrastructure. This way of establishing network elements on the clouds is called Network Function Virtualization (NFV) [1, 2, 59].

To consolidate the gains further, the virtual infrastructure is generally obtained from cloud service providers (CSPs). In this way, users neither require knowledge, control, and ownership in the computing infrastructure nor they need to host, control or own an infrastructure in order to deploy their applications. Instead, they simply access or rent the hardware or software paying only for what they use. The possibility of paying-as-you-go along with on-demand elastic operations by cloud hosting providers is gaining popularity in the enterprise computing model. The individual functions, which were monolithic specialized hardware equipment in the past, are being replaced by a set of software-based functions called Virtualized Network Functions (VNFs) [5, 6, 55]. These functions are generally spread across multiple clouds depending on the total deployment cost, availability of the required resources and proximity to end-users [3, 4, 56].