Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 34 (2014) 655 – 660

Procedia
Computer Science

International Workshop on Software Defined Networks for a New Generation of Applications and Servicess (SDN-NGAS)

SLA-driven Federated Cloud Networking: Quality of Service for cloud-based Software Defined Networks

Alexander Stanik[a,*], Marc Koerner[a], Leonidas Lymberopoulos[b]

[a]Technische Universität Berlin, Einsteinufer 17, 10587 Berlin, Germany
[b]EXUS S.A., 1 Estias Str. & 73-75 Messogion Av., 115 26 Athens, Greece

Abstract

Since the cloud paradigm becomes more and more popular for the dynamic resources allocation, new techniques and performance improvements for scalability as well as new cloud services on all three layers of the cloud stack were developed. Furthermore, another well covered topic is cloud federation concerning processing power and strength. However, the flexibility of the cloud is limited in terms of network services and federated networking between autonomous cloud data-center. With the promising opportunities of Software Defined Networking (SDN), this gap can be closed and enables cloud environments to establish networking federations between autonomous data-centers and virtual network partitioning in a single cloud infrastructure.

In this paper we are introducing and describing an architectural approach for a generic layered model and API based software architecture to orchestrate and federate heterogeneous networks. In particular, we present an architecture that enables Quality of Service (QoS) aware configurations of network resources in a cloud infrastructure of one data-center and federated networking between different SDN based cloud networks over and above the data-center network edge. Furthermore, this architecture uses a Service Level Agreement (SLA) protocol and language to expose Key Performance Indicators (KPI) and to negotiate appropriated QoS constrains which are applied to the virtually sliced underlying network substrate. In this way, capabilities of the orchestration and the current utilization of the network are building the foundation for dynamic negotiated SLAs and the within guaranteed QoS of network resources. The approach presented in this paper will change today's IT landscape and allows every organization to purchase required network characteristics on demand.

Selection and peer-review under responsibility of Conference Program Chairs

Keywords: Software Defined Networking; OpenFlow; Cloud Computing; Cloud Federation; Service Level Agreements; WS-Agreement; Federated Networking

* Corresponding author. Tel.: +49-30-314-22616; fax: +49-30-314-21060.
E-mail address: alexander.stanik@tu-berlin.de

1877-0509
Selection and peer-review under responsibility of Conference Program Chairs
doi:10.1016/j.procs.2014.07.093

1. Introduction

The cloud paradigm has changed the IT landscape in the last years due powerful cloud middlewares that provide cloud services on-demand[1]. While Infrastructure-as-a-Service (IaaS) builds the foundation where resources are virtualized and provided to the paying customer, Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS) are build on top. Virtualization is a powerful enabling technology in this domain, allowing flexible and on-demand provisioning of IT resources from the users perspective, but this is also simplifying the management of the data center for the cloud operator. However, the flexibility of cloud computing is currently offering is very limited when it comes to accessing or configuring network resources. For example the network topology behind the allocated virtual machines in the cloud is pretty opaque, inflexible and provides only basic network policies[2].

Network virtualization is a key technology for deploying next generation networks and applications[3,4,5,6]. Virtual networks can be deployed over a shared network infrastructure, while ensuring isolation of the traffic among concurrent virtual networks. This technology allows companies to specify customized virtual topologies according to their requirements. Software Defined Networking (SDN) complements the network virtualization approach by offering programmability features to the owners of virtual networks[7]. Thus, network functionalities (e.g. routing, firewalls, etc.) which were traditionally implemented by vendors within the hardware, can now be realized in software. This provides a great degree of programmability, since data center providers may develop their own network functions and thus control their virtual networks according to their needs. However, the full range of SDN capabilities is not utilized in today's cloud middlewares. Furthermore, SDN is used to facilitate future data center networks, but the benefits are often not passed to the paying customer.

The approach presented in this paper will improve cloud middlewares by integrating SDN functionalities with appropriated business models which will allow cloud providers to sell additional network services to their customers. One issue that this approach has in focus is to support Quality of Service (QoS) characteristics to the virtual environment of the deployed virtual machines. Consequently, the provider of virtual networks can apply guarantees in Service Level Agreements (SLA) to their users. Thus, customers will be able to configure the virtual network according to their requirements, which results in an SLA that not only describes the service levels, the objectives, and the prices but can also be used to monitor the fulfillment or violations of advertised guarantees.

The architectural approach which we are presenting in this paper is using programmable networking mechanisms to provide SLA guarantees for users of virtualized networks. Thus we decide to call this approach ProgNET. To achieve QoS guarantees on virtual networks, ProgNET strives to introduce a novel data link layer mechanism that will solve the data link layer loss problems that are experienced in today's virtual networks. Furthermore, in the loss-less networking environment of ProgNET, new tools and mechanisms will be implemented in order to manage and enforce customer SLAs within the virtualized network substrate. The envisaged SLA mechanism and tools will be designed not only for managing the networking resources of a single provider, but also for managing the resources within federations of providers, where the end to end user services may involve resources drawn from multiple service providers in a transparent way to the user. We will validate and demonstrate ProgNETs novel mechanisms and tools by means of experiments with the ProgNET integrated prototype implementation.

The remaining sections of this paper are organized as follows. Section 2 discuss related work in the area of SLA management and QoS for cloud federations followed by section 3 that explains the proposed architectural approach. Finally, in section 4 the concept and the architecture is concluded.

2. Related work

Contrail[8] is a project funded by the European Commission that has worked on providing SLA management support for cloud federations. In Contrail's Virtual Execution Platform (VEP)[9], a SLA management layer is added to each cloud provider that based on SLA@SOI[10] and integrates additionally the Open Virtualization Format (OVF)[11]. Contrails SLA involve characteristics of a Virtual Machine within the federated cloud, such as the amount of memory and the number of cores of the VM or the location of the VM, e.g. by including in the specification the ability to request VMs at the same physical host when the amount of data transfers between VMs is large. However, apart from providing for SLA support at the resource-level of Virtual Machines within the federation, Contrail does not cater for Quality of Service guarantees in virtual networks that interconnect VMs within a cloud federation. Our work is focused on providing QoS guarantees in interconnected VMs within cloud federations by controlling the SDN approach and the QoS support of the OpenFlow protocol so that the networking resources can be configured in order to reach the requirements of customers. Hence, our approach could be used to augment the Contrail SLA management layer with SLA guarantees for the virtual networks that interconnect Virtual machines within the federation.

Among the commercial solutions for hybrid cloud networks which are Cisco Nexus 1000V [12] and Nexus 6000V [13] InterCloud switches. However, although both switches can be used for establishment of virtual networks across hybrid clouds and thus support easily migration of VMSs and security of inter-cloud network traffic, they do not support SLA guarantees within the hybrid environment. Thus, the end to end flows spanning remote data centers are not controlled with specific QoS configurations. In comparison, our approach supports QoS support to end to end flows within a virtual network that spans across multiple data centers within a cloud federation.

The work described in [14] presents an approach for providing QoS guarantees to virtual networks (VN) that connect virtual machines within a cloud federation. In this work, the carrier network that is used to interconnect remote cloud sites is considered as a Multiprotocol Label Switching (MPLS) transport domain, thus the problem tackled within the paper is how to use MPLSs QoS classes to provide support for isolation of virtual networks within a certain MPLS QoS class. The solution given is a programmable admission control mechanism, transparent to MPLS, by using OpenFlow as the admission controller to the MPLS domain. Thus, edge OF controllers combine information of the state of the requested Virtual networks with QoS information from the MPLS carrier network in order to tag appropriately the packets that belong to each individual VN. The benefit of this approach is that it does not require any modification within the MPLS domain. Compared to [14], ProgNET does not make any assumption on the underlying carrier networks that are employed to interconnect cloud sites within a cloud federation. Thus, we have introduced the abstraction of the *ProgNET WAN Manager* component within ProgNETs layered architecture so that our approach does not depend on a specific inter-domain protocol as well as it does not make any assumptions about the networks that are used to interconnect remote cloud sites. Therefore, in our generic architecture, an implementation of a ProgNET "WAN Driver" could be the one proposed by [14] for an MPLS domain. In addition, ProgNET would enhance the work presented in [14] with the capability of negotiating network level SLAs with the MPLS carrier network provider, which would be needed in cases of network congestions within the MPLS network.

3. Architecture

The OpenFlow protocol and derivatives enable an innovative mechanism to control large networks and high performance networks, as often used in cloud service provider facilities. Moreover, OpenFlow provides the opportunity to create a wide range of new applications and the corresponding use-cases based on providing the network and the related capacity itself as a service. Configuration and behavior of SDN can be controlled by one entity and changed on demand. The properties of the underlying network and its connected servers can therefore be treated as a resource.

SLAs are established and managed between cloud service provider and consumers and are an instrument to describe a service and its quality. This issue is of crucial importance for companies whose success depends on the advertised QoS that the service provider delivers. However, SLAs are mostly formal language written by lawyers that is static and just published on the providers web sites. This might be enough to cover the majority of customers, but it is often an obstacle for companies that would like to rent computing and storage resources with additional representations and warranties. SLAs for network services do not exist in today's cloud landscape, because they are hard to describe in programmatic fashion, and guarantees are not advertised. This architecture will enhance IaaS with mechanisms for automated negotiation and creation of SLAs for network services that deliver QoS guarantees **(a) between VMs in a single data-center cloud (b) to external cloud services located anywhere in the internet (c) for interconnections between several heterogeneous cloud environments**.

In order to negotiate, create, and evaluate SLAs in a machine-readable form, we will use the WS-Agreement standard and its extension WS-Agreement Negotiation, as protocol and language. Thus, customers will be able to define their requirement in a standardized way and request services dynamically from an independent service provider. Furthermore, the ProgNET approach will extend cloud middlewares with OpenFlow capabilities to apply negotiated and guaranteed quality constrains to the underlying cloud SDN substrate.

The combination of WS-Agreement (Negotiation) and OpenFlow offers the opportunity for new use-cases and business models as well. The approach presented in this paper demonstrates new capabilities based on a shared virtual

Ethernet circuit overlay. The basic idea is to enable customers to query and book available network routes between local and external hosts including guaranteed network characteristics. In order to deploy a requested QoS-based network, the customer and the cloud provider will negotiate service qualities which will be applied when both come to a common agreement. To achieve this goal, we designed a layered architecture that is depicted in the figure below.

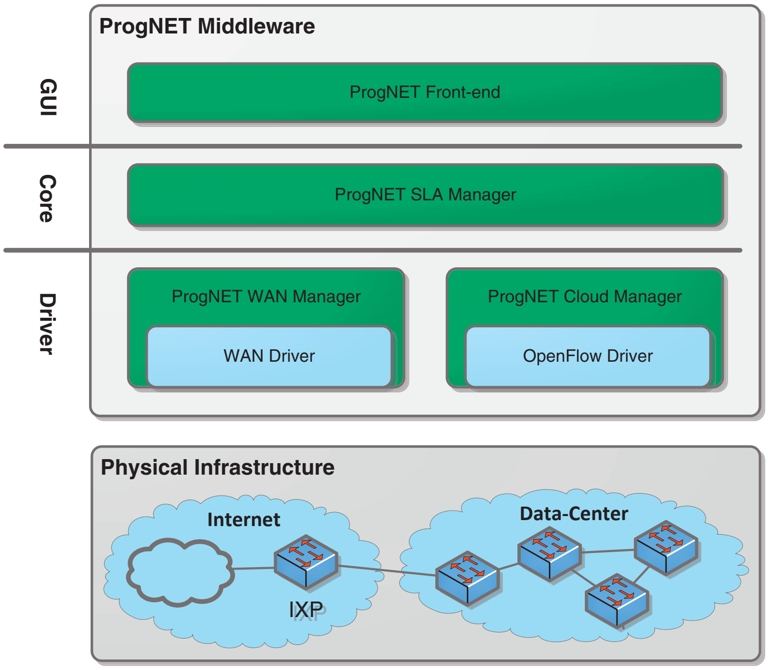

Fig. 1. ProgNET layered architecture

The ProgNET stack is a layered architecture consisting of multiple layers building up on each other. On the top layer the **ProgNET Front-end** is the graphical user interface that allows customers in a human-friendly way to specify network requirements, browse for different network services and to negotiate them in order to create a SLA and establish an appropriate network environment. Furthermore, this component provides also monitoring for detecting SLA violations and notification mechanisms to inform the customer and the provider about inadequate behaviors of guaranteed resources.

The **ProgNET Front-end** however is not managing any network resources directly, but just aspects regarding the establishment of network services based on SLAs. All properties related to network resources are delegated to the **ProgNET SLA Manager**, which collects information about the underlying network from the **ProgNET Cloud Manager** and the **ProgNET WAN Manager**. Based on this information the **ProgNET SLA Manager** provides functionalities to negotiate and create aggregated SLAs and then appropriately establish QoS-based network environments in a single data-center and to external services. Furthermore, this **ProgNET SLA Manager** enables federated networking between several autonomous clouds with heterogeneous infrastructures.

ProgNET is designed modularly, thus each component can be exchanged with any other third-party component. In the case of a commercial service provider, the front-end will be the same as for booking all other traditional cloud services. In particular, all properties related to computing or other traditional cloud resource aspects are covered by third-party components as illustrated in the figure 2, because the focus of this novel architectural approach is on network services and not on already existing cloud service federations. In fact, several projects had elaborate virtualization, stage, and federated cloud services over the last years. In contrast, we are not covering them again, but focus consequently on the missing part in today's cloud landscapes.

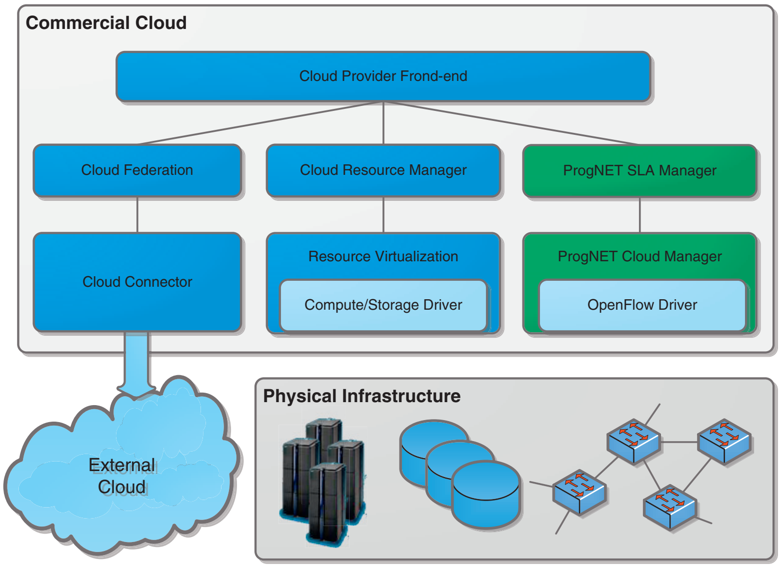

Fig. 2. ProgNET in a commercial Cloud Environment

The existing Cloud Front-end and the specific ProgNET Front-end can also coexist and be used as graphical interfaces to the cloud. The **ProgNET SLA Manager** is used for requesting network services. Thus, an interface is required that is suitable for different networks, enabling federated networking and SLA negotiation for network services.

The **ProgNET SLA Manager** is waiting for requests from customers and negotiates the conditions for the cloud network environment. The ProgNET back-end consisting of the **ProgNET Cloud Manager** and the **ProgNET WAN Manager** which are providing information about the network utilization and available capacities to the **ProgNET SLA Manager**. If both the customer and the provider agree to the negotiated conditions, a SLA is created that serves as a formal contract. Afterwards, the network connections will be established between the VMs/entities according to the conditions of the agreement based on the underlying network topology and the remaining available.

The key components to manage the local and global available network capabilities and their particular QoS allocation are the **ProgNET Cloud Manager** and the **ProgNET WAN Manager**. These components are building the backbone of the entire architecture and the corresponding API related software integrations. Especially the **ProgNET Cloud Manager** entity will introduce innovative mechanisms to allocate local network overlays based on SDN and connect them with QoS enforcements and guarantees. Therefore, this architecture addresses several mechanisms based on the centralized control plan in SDN substrates and the opportunity to enforce QoS in data-centre SDN access and legacy aggregation/core networks as well as for comprehensive SDN networks. For non-SDN enabled networks in data-centers, this architectural approach enforced QoS with legacy mechanisms based on MPLS and VLAN tagging at the OpenFlow network edge to control and manage the forwarding process. For pure SDN substrates (most likely in a single data-center) this architecture consider all opportunities for QoS enforcements based on OpenFlow capabilities, e.g. meter tables, of the latest OpenFlow standards like OF 1.3 and beyond [15]. This will redefine the QoS integration and usage of today's networks, because this means that ProgNET is creating a new granularity for the allocation and an almost deterministic behaviour as provided by real time Ethernet.

In contrast to the **ProgNET Cloud Manager**, the **ProgNET WAN Manager** is an abstraction for the particular external carrier or IXP related network and directly expose the relevant QoS capabilities as external service. Besides of exposing SLA-based networking capabilieties to the outside world, the **ProgNET WAN Manager** checks external routes by discovering external **ProgNET WAN Manager**. To provide a reasonable solution ProgNET considers also solutions based on direct circuits for Cloud Data-Centres through provider networks to directly connect to the IXP, which is exposing and integrating a service architecture based on this abstraction layer to expose and sell this QoS.

4. Conclusion

The presented architecture describes an initial approach for a layered network cloud federation management stack, which enables the integration of network federation to heterogeneous cloud middlewares. This concept addresses the federation of SDN based cloud networks by an orchestration of SDN and traditional network substrates. Moreover, the highest layer presents an SLA based interface for the network resources allocation based on QoS parameters which are basically enforced by the bottom layer. This is equivalent to the ”get what you pay” concept already used for e.g. CPU or memory allocation for virtual hosts.

Altogether the described architecture presents a reasonable software stack approach for the integration of network based cloud federation over heterogeneous network substrates and cloud middle-ware software. In the next steps we will validate and demonstrate ProgNETs novel mechanisms and tools by means of experiments with the ProgNET integrated prototype implementation.

References

1. Stanik, A., Hovestadt, M., Kao, O.. Hardware as a service (haas): Physical and virtual hardware on demand. In: *Proceedings of the 4th IEEE Intl. Conference on Cloud Computing Technology and Science*; CloudCom 2012. IEEE publishers; 2012, .
2. Korner, M., Stanik, A., Kliem, A.. An approach for qos constraint networks in cloud environments. In: *Network of the Future (NOF), 2013 Fourth International Conference on the*. 2013, p. 1–3. doi:10.1109/NOF.2013.6724522.
3. Sherwood, R., Gibb, G., Yap, K.K., Appenzeller, G., Casado, M., McKeown, N., et al. Can the production network be the testbed? *In USENIX Symposium on Operating Systems Design and Implementation (OSDI)* 2010;.
4. Sherwood, R., Gibb, G., Yap, K.K., Appenzeller, G., Casado, M., McKeown, N., et al. Flowvisor: A network virtualization layer. *Technical Report Openflow-tr-2009-1, Stanford University* 2009;.
5. Salvadori, E., Corin, R., Broglio, A., Gerola, M.. Generalizing virtual network topologies in openflow-based networks. In: *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*. Dec., p. 1–6. doi:10.1109/GLOCOM.2011.6134525.
6. Corin, R., Gerola, M., Riggio, R., De Pellegrini, F., Salvadori, E.. Vertigo: Network virtualization and beyond. In: *Software Defined Networking (EWSDN), 2012 European Workshop on*. 2012, p. 24 –29. doi:10.1109/EWSDN.2012.19.
7. Koerner, M., Almus, H.. Hla - a hierarchical layer application for openflow management abstraction. In: *Proceedings of the Fourth International Conference on Network of the Future (NoF'13)*. Pohang, Korea; 2013, p. 1–4.
8. Carlini, E., Coppola, M., Dazzi, P., Ricci, L., Righetti, G.. Cloud federations in contrail. In: Alexander, M., D'Ambra, P., Belloum, A., Bosilca, G., Cannataro, M., Danelutto, M., et al., editors. *Euro-Par 2011: Parallel Processing Workshops*; vol. 7155 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. ISBN 978-3-642-29736-6; 2012, p. 159–168. URL: `http://dx.doi.org/10.1007/978-3-642-29737-3_19`. doi:10.1007/978-3-642-29737-3_19.
9. Harsh, P., Jegou, Y., Cascella, R., Morin, C.. Contrail virtual execution platform challenges in being part of a cloud federation. In: Abramowicz, W., Llorente, I.M., Surridge, M., Zisman, A., Vayssiére, J., editors. *Towards a Service-Based Internet*; vol. 6994 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. ISBN 978-3-642-24754-5; 2011, p. 50–61. URL: `http://dx.doi.org/10.1007/978-3-642-24755-2_5`. doi:10.1007/978-3-642-24755-2_5.
10. Theilmann, W., Yahyapour, R., Butler, J.. Multi-level sla management for service-oriented infrastructures. In: Mhnen, P., Pohl, K., Priol, T., editors. *Towards a Service-Based Internet*; vol. 5377 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. ISBN 978-3-540-89896-2; 2008, p. 324–335. URL: `http://dx.doi.org/10.1007/978-3-540-89897-9_28`. doi:10.1007/978-3-540-89897-9_28.
11. Crosby, S., Doyle, R., Gering, M., Gionfriddo, M., Hand, S., Hapner, M., et al. Open virtualization format (ovf) specification. *Standards and Technology; no DSP0243 in DMTF Specifications; Distributed Management Task Force* 2009;.
12. Cisco. 2014. `http://www.cisco.com/c/en/us/products/collateral/switches/nexus-1000v-switch-vmware-vsphere/data_sheet_c78-722606.pdf`.
13. Cisco. 2014. `http://www.cisco.com/c/en/us/products/collateral/switches/nexus-6001-switch/data_sheet_c78-726128.pdf`.
14. Khan, A., Kiess, W., Perez-Caparros, D., Triay, J.. Quality-of-service (qos) for virtual networks in openflow mpls transport networks. In: *Cloud Networking (CloudNet), 2013 IEEE 2nd International Conference on*. 2013, p. 10–17. doi:10.1109/CloudNet.2013.6710552.
15. Foundation, O.N.. Openflow switch specification / version 1.4.0. 2013. `https://www.opennetworking.org/`.