

Validation of IaaS-based Technologies for 5G-Ready Applications Deployment

Roberto Bruschi
DITEN, University of Genoa, Italy

Fernando Díaz Bravo
Atos SE, Sevilla, Spain

Sergio Mangialardi
S2N National Lab., CNIT, Italy

Franco Davoli
S2N National Lab., CNIT, Italy

Chiara Lombardo
S2N National Lab., CNIT, Italy

Jane Frances Pajo
DITEN, University of Genoa, Italy
S2N National Lab., CNIT, Italy

Abstract— In an attempt to assess the suitability of current Infrastructure-as-a-Service (IaaS) technologies for fulfilling 5G-ready applications requirements, this paper has exploited the CNIT S2N testbed to perform a validation over a complete 5G ecosystem. The testbed supports the telecom network platform developed within the MATILDA Project, which adopts the vanilla versions of the main reference projects/software suites in the 5G field, such as OpenStack and Open-Source MANO (OSM). By including such familiar technologies in the 5G ecosystem, vertical industries would interact with a familiar environment similar to today’s scenarios, which would lower the barrier for creating 5G-ready applications. However, results show that current solutions are blamable for around 99% of the deployment time of applications and network slices, a figure that bumps their lifecycle up to the order of magnitude of minutes proving them still not suitable to fulfill 5G requirements.

Keywords—5G Testbed, Network Functions Virtualization, Edge Computing

I. INTRODUCTION

The integration of vertical industries in the design of the fifth-generation (5G) mobile networks is seen as a cardinal facet to open truly global markets for innovative digital business models. For this reason, the needs of vertical industries should be considered as high-priority drivers of 5G network design by involving them in the 5G ecosystem since the beginning.

Modern cloud technologies and architectures are largely recognized as the foundations of the upcoming 5G ecosystem [1]. These technologies are expected to provide the means for allowing the softwarisation revolution in telecommunication infrastructures, through frameworks such as Network Functions Virtualization (NFV), as well as to act as key enablers for new and more pervasive computing paradigms, like edge computing [2], [3].

One of the most significant added-value aspects contributing to this ever-increasing success can be found in the cloud architecture itself, as it allows a clear and effective splitting of roles among the main actors involved. Moreover, along with the disruptive “as-a-Service” paradigm and the agility of modern computing/networking virtualization technologies, a clear

advantage of embracing cloud technologies is that it lowers the barrier for creating 5G-ready applications, by providing vertical industries with a familiar environment similar to today’s cloud scenarios. However, while the support of applications and network slices interplay is available in almost all the enterprise-level Virtual Infrastructure Managers (VIMs), the suitability of the current cloud operating systems characterizing the Infrastructure-as-a-Service (IaaS) paradigm, as well as existing network orchestrators, needs to be proved in the presence of stringent Key Performance Indicators (KPIs).

In an attempt to evaluate the impact of the current IaaS technologies, and assess their suitability for fulfilling 5G applications requirements, this paper has exploited the CNIT S2N testbed to perform a validation over a complete 5G ecosystem. The testbed supports the telecom network platform developed within the MATILDA Project [4], which allows a coordinated and fully automated 5G-ready vertical applications lifecycle management in a multi-tenant/domain ecosystem. Results show how current technologies deeply affect the times required to deploy 5G-ready applications and network services, to the point that they preclude the fulfilment of the stringent 5G systems requirements [5].

The remaining of the paper is organized as follows. Section II describes the S2N testbed, with details on the supported telecom layer platform in Section III and on the Operations Support System (OSS) in Section IV. Finally, the evaluation is reported in Section V and conclusions are drawn in Section VI.

II. THE CNIT S2N TESTBED

The CNIT S2N testbed is a multi-layered hardware and software facility for the advanced experimentation and demonstration of 5G, edge and cloud computing technologies. It is specifically conceived to host multiple isolated tenant spaces (e.g., as project environments) that can emulate complete 4/5G network environments, as well as to manage and configure their respective physical/virtual resources through a Metal-as-a-Service (MaaS) [6] approach and the software elements through Canonical’s Juju [7] platform.

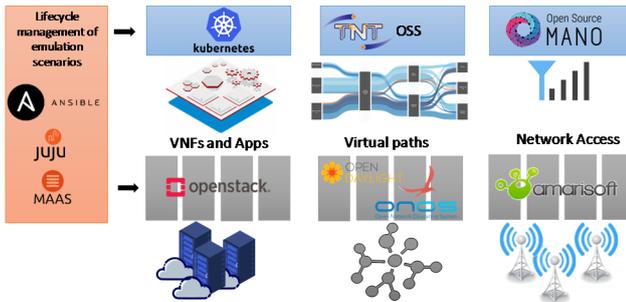


Fig. 1. Main building blocks composing the MATILDA 5G Telecom and Infrastructure Platforms.

The main components of the testbed are shown in Fig. 1. In a nutshell, the testbed has hardware capabilities for setting up 4/5G network environments, complete with User Equipment (UE), radio/wired access, programmable connectivity, security system and (geo-distributed) computing domains, as well as equipment useful for performance evaluations such as traffic generators and power monitors. In addition, multiple Virtual Network Functions (VNFs) have already been (and will be) onboarded to enable 4G/5G mobile radio access networks (RANs), network slices, and a wide range of other services.

The testbed infrastructure is an integration of both general- and special-purpose equipment to realize underlying 4/5G network substrates, which includes:

- 20+ high-end servers, providing approximately 700 cores, 2TB RAM, 100TB of SSD/SAS storage, and with dual-port 10/40/56Gbps Ethernet NICs that supports hardware acceleration, SR-IOV and precision time synchronization (IEEE 1588);
- 10+ SDN switches, providing more than 1000 ports with speeds ranging from 1 to 56Gbps, and a total L2/L3 forwarding speed of more than 5.5 Tbps;
- 1x hardware firewall, providing isolation among tenant spaces with 20Gbps processing speeds;
- 2x professional network traffic generators with 10ns latency measurement precision;
- 3x smart Power Distribution Units (PDUs), able to collect the energy consumption of any hardware elements in the testbed;
- 4+ 5G/LTE cells, providing 1 5G New Radio (NR) base station (gNodeB) or 6 LTE eNodeBs with an Amari SDR device, plus 3 eNodeBs based on commercial Amarisoft products or on open-source projects (e.g., SRS LTE) and USRP B210 NI boards; and
- 10+ heterogeneous UEs, including drones, NB-IoT devices, smartphones/tablets, and high-speed Customer Premises Equipment (CPE).

Based on these infrastructure resources, the testbed platform integrates both proprietary and open-source software to realize and dynamically manage the tenant spaces on top – specifically, the involved 4/5G and application-/slice-specific network services (NSs), (geo-distributed) computing domains and wide-area overlay networks. In this respect, the network

platform layer of the MATILDA framework allows realizing the autonomic management of the lifecycle of 5G network slices and edge computing resources.

MATILDA targets a coordinated and fully automated control of all the resources/services at any layer for allowing the 5G-ready vertical applications lifecycle management in a multi-tenant/domain ecosystem. This entails enabling vertical industries to autonomously manage their application graphs, from their planning (Day-0) and their first deployment (Day-1), through the in-life operations (Day-2), without the actual knowledge of the network infrastructure underneath, but only via a dedicated 5G network slice [8]. The lifecycle of the NFV services (NSs) composing the slice is tightly intertwined with the one of the corresponding application; thus, cooperation of the orchestrators playing at application and network level is critical and has been explicitly addressed via a dedicated convergence layer that will be described in Section IV.DIV.D.

III. THE MATILDA TELECOM LAYER PLATFORM

The scope of the MATILDA Project is to deliver a holistic and innovative 5G framework to undertake the design, development and orchestration of 5G-ready vertical applications (vApps hereinafter) and 5G network services over programmable infrastructures. To this goal, a telecom layer platform has been designed to realize the autonomic management of the lifecycle of 5G network slices and edge computing resources. In accordance with [9], the main stakeholders actively involved in this environment are three: the vertical industry owing the vApp, the telecom service provider delivering 5G services, and the telecom infrastructure provider offering computing and communication facilities.

Fig. 2 depicts the main functional blocks composing the telecom layer platform. The *Operations Support System* (OSS) is in charge of managing all functions and operations required for the placement of a vApp over a network slice, as well as maintaining the information on all the data regarding the deployed vApp, network services, available resources, and so on. The *NFV Orchestrator* (NFVO) is responsible for the lifecycle management of the network services, both those composing the base 4/5G services and the ones provided to slices. The *Wide-area Infrastructure Manager* (WIM), based on the Ericsson Network Manager [10], is devoted to manage and monitor the wide-area communication resources, to create overlay networks for vApps and base telecommunication services, as well as to provide information on the resources available in the distributed 5G infrastructure. Finally, the *Virtual Infrastructure Manager* (VIM – one instance per each distributed computing facility), abstracts and exposes computing, storage, and networking capabilities of datacenters within the 5G infrastructures.

As shown in the figure, the OSS and the NFVO act in the telecom service provider domain, while the VIM and the WIM in the infrastructure provider one. Vertical industries can autonomously manage the lifecycle of their application graphs by means of Vertical Application Orchestrators (VAOs). The VAO interacts with the OSS via a specific metamodel, called slice intent, which represents all requirements that should be satisfied during the creation of a slice. It is worth noting that all

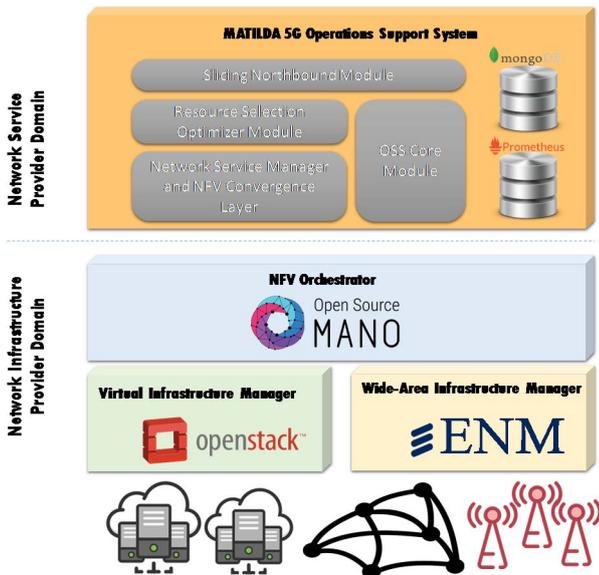


Fig. 2. Main building blocks composing the MATILDA 5G Telecom and Infrastructure Platforms.

of these building blocks and their reference points are fully compliant with the specifications of the ETSI NFV architectural framework [11].

Since the short-term applicability of the MATILDA approach on emerging 5G infrastructures and tools has been one of the key and challenging objectives of the project, the vanilla versions of existing and emerging reference projects/software suites in the 5G field have been adopted as much as possible. In particular, as can be seen in Fig. 2, the NFVO and the VIM have been realized through the latest (unmodified) versions of ETSI Open-Source MANO [12] (OSM) and OpenStack [13] (OS), the wide-area network controller is OpenDaylight [14], and so on. This choice, while still following the layered approach presented in [15], has allowed to gather most of the innovation in the OSS, whose prototype has been conceived and completely developed within the MATILDA Project.

IV. THE MATILDA OSS

The MATILDA OSS has been designed according to a highly modular architecture: all the software services are state-of-the-art cloud-native software, i.e., as stateless services (or more precisely services with a state maintained in an external database), inherently parallelizable, and ready to be integrated with the service-mesh paradigm. As shown again in Fig. 2, the prototype development is composed of a suite of four main software services and two different databases (MongoDB [16] and Prometheus [17]) acting as persistency layer.

In more detail, the *Slicing Northbound* module allows the deployment and lifecycle management of the application graph components by supporting the communication between the VAO and the OSS. The *OSS Core* module offers OSS external interfacing towards the VIMs and the WIM(s), since it includes the VIM and WIM convergence layers. The *Resource Selector Optimizer* (RSO) module provides the optimization and reinforcement policies/algorithms to be executed in the OSS. Finally, the *NFV Convergence Layer* (NFVCL) module

provides a level of abstraction for the flexible and high-level management of the complete lifecycle orchestration of network services, Virtual Network Functions (VNFs) and Physical Network Functions (PNFs) instantiated in the 5G infrastructure.

A. The Slicing Northbound Module

The Slicing Northbound Module interacts with the VAOs for the deployment and lifecycle management of the application graph components over the created network slice. It provides mechanisms for requesting the creation of an application-aware network slice according to the set of computational, networking and Quality of Service (QoS) requirements reported in the slice intent. Upon the successful creation and the provision of the required network slice, it supports the lifecycle management of the application graph components.

B. The OSS Core

The OSS Core module, the central building block of the OSS, is in charge of managing the logic and the sequence of actions of the other blocks. It has a highly modular, asynchronous, and multi-threading architecture to facilitate future extensions, like for instance the support of new VIM and WIM interfaces. In fact, it includes the VIM and WIM convergence layers, as well as an interface for triggering the setup of base network services directly driven by the Telecom operator. The OSS Core has been realized as a cloud-native stateless service: all the status and configuration information regarding the slice intents, the materialized slices, and the base network services are stored in specific “collections” in the MongoDB persistency layer. When a new request is received, the OSS Core Module parses it and selects the proper workflow of (sequential) actions to be requested to the other registered OSS modules, or to external architectural building blocks.

C. The Resource Selector Optimizer

The RSO module is in charge of providing optimization and reinforcement policies/algorithms to be executed in the OSS. Three submodules have been provided in the RSO to cope with the required operations for the slice creation procedure. The Graph Reduction submodule aggregates application components according to a set of link QoS constraint thresholds, generating a reduced graph composed of “macro nodes”. The vApp components in the same macro-nodes will be treated as an inseparable set in the following placement and deployment operations/actions. The Utilization Forecasting submodule periodically collects performance metrics for each VIM registered in the OSS, that are the utilization of vCPU, RAM and disk available in a pre-determined observation period, to predict their behavioral trends for a given time horizon. Finally, the Placement Optimization submodule exploits both the slice resource requirements and the metrics forecasted by the previous submodule to select the most suitable combinations of VIMs where to place the macro nodes.

D. The NFV Convergence Layer

The NFV Convergence Layer (NFVCL) provides the level of abstraction required for the flexible and high-level lifecycle management of instantiated NSs, VNFs and PNFs. The internal architecture of the NFVCL module is depicted in Fig. 3. Northbound, it communicates with the OSS Core to interact with the RSO during the resource negotiation phases preceding the

materialization of a network slice, and with the Slicing Northbound module for the resulting allocation of the network resources in the VIMs. Southbound, the NFVCL interacts with the NFVO to drive the lifecycle management of NSs and P/VNFs. While the NS Descriptor (NSD) specified by ETSI NFV is composed of a pre-determined number of VNFs making up the NS graph, which cannot be modified without destroying and recreating the NS/VNF instances, MATILDA has designed a new, generalized structure that has been defined as “network service blueprint”. Blueprints have a template structure describing the logical architecture of the NS graph in terms of types of VNFs, their inter-connections, and the virtual networks to be used towards the outside. Within this template structure, single VNFs or sub-sets of the template graph can be annotated in order to be replicated in multiple NS instances (NSIs). Along with the blueprints, a number of software plugins, called VNF Configurators, are made available to provide all information regarding the Day-2 operation primitives for each VNF. Further, two submodules, namely the NFV Service Mapper (NFVSM) and the NFV Service Setup (NFVSS), act as interfaces towards the OSS Core module and the NFVO, respectively.

V. EVALUATION

The goal of the following evaluation is to assess the suitability of the existing IaaS-based technologies for supporting 5G-ready vertical applications.

A. Testbed Deployment Setup

For the results reported in the following sub-sections, the S2N testbed has been deployed as shown in Fig. 4. A wide-area network interconnects three VIMs and two eNodeB PNFs. VIMs 1 and 3 are supposed to be “edge” VIMs, in the sense that they are placed geographically closer to the eNodeBs, while VIM 2 is placed in the “core” of the Telecom infrastructure.

On top of this infrastructure, the NFVO instantiates a Public Land Mobile Network (PLMN) as base network service. The PLMN in Fig. 4 is composed of a number of NSs, whose VNFs are highlighted with the light blue color. One service, in the core VIM 2, includes a single monolithic VNF implementing the EPC functionalities. This VNF has two main network interfaces, one towards the 4G network and one towards the public Internet. On each edge VIM, a further NS is created for managing and configuring the eNB PNF and for providing S1 bypass capabilities [18]. The S1 bypass has two main network interfaces, one towards the wide-area network to interconnect with the EPC and the eNBs, and one towards a virtual layer-2 network internal to the VIM.

An OpenStack project is fully dedicated to the vApp and directly controlled by the VAO via a connection to the public Internet. Different vApps in the same VIM will be assigned to different and isolated OpenStack projects, highlighted with a sketched orange box in the figure. vApp components might be placed on edge VIMs close to the attach point network, if they have particular performance requirements, or in other VIMs, like the ones in the infrastructure core. In that case, a further NS (whose VNFs are colored in green in Fig. 4) should be created and deployed to provide connectivity among the vApp components in the different VIMs.

B. vApp Planning and Deployment Times

In order to compute the time required after the reception of a slice intent to deploy the corresponding vApp, several measurements have been performed on the MATILDA testbed for both the OSS and OpenStack. Specifically, we have considered four application graphs composed of a number of VMs between four and seven. They have been deployed over the three VIMs shown in Fig. 4, without any constraints on their proximity requirements.

In the OSS, when an application graph deployment request is received, the RSO composes an aggregated graph based on the proximity requirements of each application component. Upon reception of such graph, the WIM computes a list of candidate VIMs that fulfil the application requirements, which is forwarded again to the RSO to select one of the solutions. This decision drives the initialization of an OpenStack project, in each involved VIM, which entails creating the required tenant spaces shown in Fig. 4.

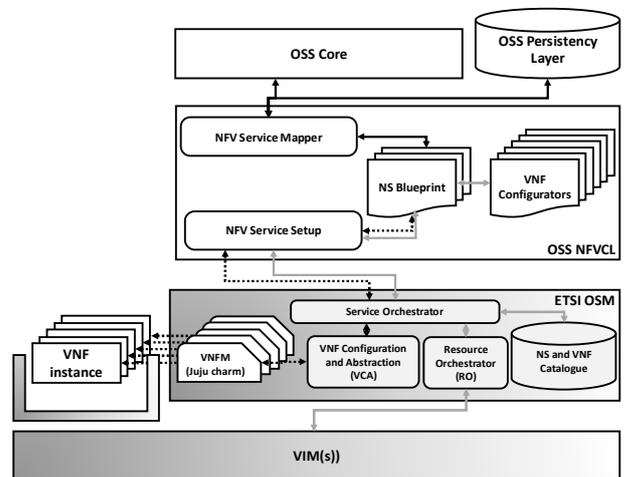


Fig. 3. The internal architecture of the NFV Convergence Layer and its interactions with external entities.

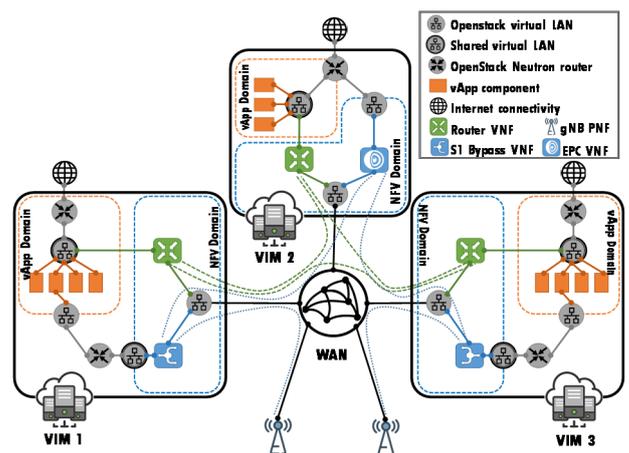


Fig. 4. Testbed deployment for the evaluation, including a base network service and a network slice, produced by the MATILDA Telecom Layer Platform.

Results in Fig. 5 show the distribution of the execution times ascribable to the OSS (left bar, stack of the RSO and WIM execution times) and to OpenStack at this phase, which completes the Day-1 operations of the vApp lifecycle. Execution times are plotted over the application graph and the number of VIMs available during the test, and are reported in milliseconds for the OSS and in seconds for OpenStack.

The OSS execution times grow with the size of the application graph because of the contribution of both RSO phases. In fact, it can be seen that the execution time ascribable to the WIM is constant with the application graph size and only presents negligible variations with the number of VIMs. On the other hand, since OpenStack performs the same operations regardless of the application size, and each VIM creates its project independently, the execution times are not correlated neither with the application graph size nor with the number of available VIMs.

These tests highlight how the execution times ascribable to OpenStack represent over 99% of the total, bringing the total deployment time in a range that is suitable for the IaaS context, but is definitely not acceptable for the 5G vertical environment, in which service agility is seen as a non-negotiable.

In order to better understand the execution times trends, the tests have been repeated for Application Graph 2 ten times, and the average results are reported in Fig. 6. The results representation is based on quartiles: a box is drawn between the 1st and 3rd quartiles, a line along the median (2nd quartile), two additional lines indicating the minimums and maximums outside the 1st and 3rd quartiles, respectively, and the labeled value reports the mean values.

C. NSs Onboarding, Deploying and Operation

Lifecycle operations involving the NFV services are not parallel to the ones on vApps shown in the previous section, but they are rather nested, as the Day-0 operations triggered at the NFVO level start at the completion of the Day-1 operations described in the section above.

The results plotted in Fig. 7 represent the execution times measured for Day-0/1/2 operations on a number of NSIs, from one to five, which have been deployed over the S2N testbed. The considered NSs are virtual routers (i.e., the green icons in Fig. 4) based on the VyOS project that provide auxiliary network functionalities on a per vApp slice basis, such as routing protocols, firewall, DNS, DHCP, NAT, etc.

Results represent the time ascribable to each individual operation that takes place upon reception of a slice intent to put into operation the network slice. In more details, the NFVSM identifies the required NSs and selects the most suitable blueprint fulfilling the QoS requirements (operation indicated as 0-1 in Fig. 7). Starting from the information in the blueprint, the NFVSS is responsible to properly generate NSD packages (operation 0-2) and onboard them on the NFVO, which concludes Day-0 operations (0-3).

In Day-1 operations, the NFVSS asks the NFVO to instantiate the NSIs (1-1) and waits for the completed instantiations. Day-1 operations end when all VNF images and

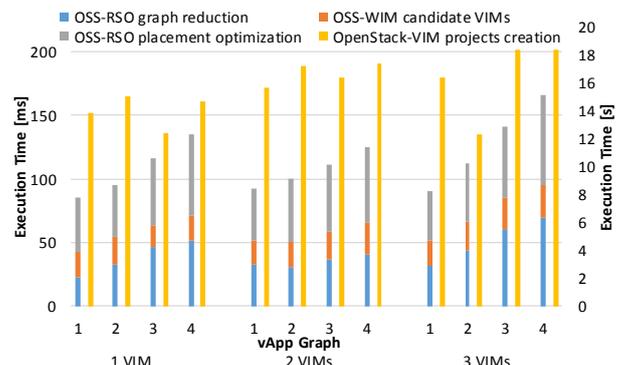


Fig. 5. Total execution times for Day-1 operations ascribable to the OSS (plotted on primary vertical axis) and to OpenStack (on secondary axis).

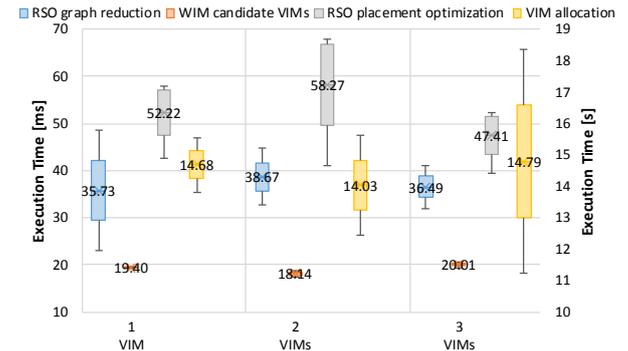


Fig. 6. Average execution times for Day-1 operations ascribable to the OSS modules and to OpenStack. The box plots report the average results obtained by deploying Application Graph 2 ten times.

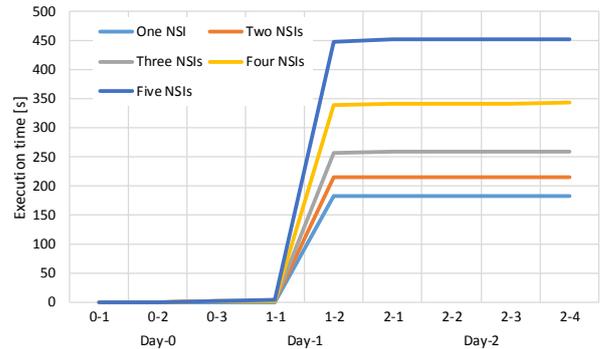


Fig. 7. Execution times for Day-0/1/2 operations on NS instances.

their corresponding VNF Manager (VNFM, in the form of a Juju charm) are up and running (1-2).

Upon successful fulfilment of Day-0/1 operations, the NFVSS retrieves the proper VNF Configurators (2-1) and sends the list of Day-2 primitives to the Juju charm of each VNF composing the service. In the results, start (2-2) and stop (2-3) primitives have been configured for each NS. Day2 operations end once these primitives have been successfully run for each service (2-4).

In the same way as in the previous results, the number of VIMs does not impact the execution time of the Day-0/1/2 operations, so it is not considered in Fig. 7. Since the instantiation of the NSs in the VIMs is sequential, times increase with the number of instances. However, the operation involving

OSM, e.g., the instantiation of the NSIs (1-1), accounts for almost all the execution time, keeping the whole lifecycle in the order of magnitude of minutes.

This behavior is highlighted in Fig. 8, which reports the total execution time (the bars in the graph) and the contributions of the NFVCL and OSM (the lines in the graph). The total execution time grows from 3 to 7.5 minutes with the number of VyOS instances, of which OSM is responsible for 98-99.5%.

VI. CONCLUSIONS

The integration of vertical industries in the design of the fifth-generation (5G) mobile networks should be considered with high priority in order to open truly global markets for innovative digital business models. For this reason, one of the main tools to enable verticals to deploy and orchestrate 5G-ready applications is to include cloud technologies in the upcoming ecosystem, as it lowers the barrier for creating 5G-ready applications, by providing vertical industries with a familiar environment similar to today's scenarios.

However, the suitability of the current technologies characterizing the Infrastructure-as-a-Service (IaaS) paradigm, such as cloud operating systems as well as existing network orchestrators, needs to be proved in the presence of stringent Quality of Service (QoS) levels.

In this paper, the CNIT S2N testbed, a multi-layered facility for the advanced experimentation and demonstration of 5G, edge and cloud computing technologies, has been exploited to perform a validation over a complete 5G ecosystem including the most common reference projects/software suites in the 5G. The testbed supports the telecom network platform developed within the MATILDA Project, a highly modular architecture specifically conceived to host multiple isolated tenant spaces that can emulate complete 4/5G network environments.

The evaluation has been carried out by analyzing the execution times for deploying applications and network slices, and in particular the contributions ascribable to the MATILDA platform components and to IaaS technologies (OpenStack and OSM). Results show how the algorithms operating in the OSS scale with the number of instances and have negligible execution times, while both OpenStack and OSM are responsible for around 99% of the deployment of vApps and NSs, bumping their lifecycle up to the order of magnitude of minutes. This range is suitable for the IaaS environment, but current technologies are still not suitable to fulfill 5G requirements such as ultra-low latency and ultra-high reliability.

ACKNOWLEDGMENT

This work has been supported by the Horizon 2020 5G-PPP Innovation Action MATILDA (Grant Agreement no. 761898) and by the Horizon 2020 Innovation Action SPIDER (Grant Agreement no. 833685).

REFERENCES

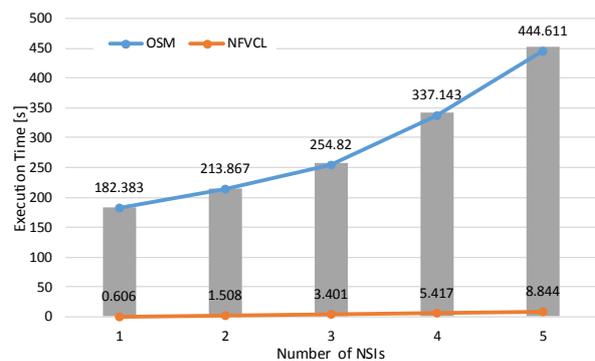


Fig. 8. Total execution times for Day-0/1/2 operations on NS instances, and contributions ascribable to the NFVCL and to OSM.

- [1] D. Szabó, F. Németh, B. Sonkoly, A. Gulyás, and F. H.P. Fitzek. "Towards the 5G Revolution: A Software Defined Network Architecture Exploiting Network Coding as a Service," In Proc. of the 2015 ACM SIGCOMM Conf., pp. 105-106, Aug. 2015.
- [2] D. Soldani and A. Manzalini, "Horizon 2020 and Beyond: On the 5G Operating System for a True Digital Society," in IEEE Vehicular Technology Magazine, vol. 10, no. 1, pp. 32-42, March 2015.
- [3] ETSI GS MEC 002, "Mobile Edge Computing (MEC); Technical Requirements", version 1.1.1, March 2016. URL: http://www.etsi.org/deliver/etsi_gs/MEC/001_099/002/01.01.01_60/gs_MEC002v01010101p.pdf
- [4] MATILDA - A Holistic, Innovative Framework for Design, Development and Orchestration of 5G-ready Applications and Network Services over Sliced Programmable Infrastructure, URL: <http://www.matilda-5g.eu/>
- [5] "Service requirements for next generation new services and markets", 3GPP TS 22.261 version 15.5.0 Release 15, URL: https://www.etsi.org/deliver/etsi_ts/122200_122299/122261/15.05.00_60/ts_122261v150500p.pdf.
- [6] MAAS-Metal as a Service, URL: <https://maas.io/>
- [7] Juju, URL: <https://jujucharms.com/>.
- [8] X. Foukas, G. Patounas, A. Elmokashfi, M. K. Marina, "Network Slicing in 5G: Survey and Challenges," IEEE Commun. Mag., vol. 55, no. 5, pp.94-100, May 2017.
- [9] 3GPP, "Study on management and orchestration of network slicing for next generation network," TR 28.801, version 15.0.0, Sept. 2017.
- [10] The Ericsson Network Manager, <https://www.ericsson.com/en/portfolio/digital-services/automated-network-operations/network-management/network-manager>.
- [11] ETSI "Network Functions Virtualisation (NFV), Management and Orchestration," ETSI GS NFV-MAN 001 V1.1.1, URL: http://www.etsi.org/deliver/etsi_gs/NFV-MAN/001_099/001/01.01.01_60/gs_nfv-man001v010101p.pdf
- [12] Open Source MANO, URL: https://osm.etsi.org/wikipub/index.php/Main_Page.
- [13] OpenStack, URL: <https://www.openstack.org/>.
- [14] OpenDaylight, URL: <http://www.opendaylight.org/software>.
- [15] 5G PPP Architecture Working Group, 5G empowering vertical industries, February 2016, URL: https://5gpp.eu/wp-content/uploads/2016/02/BROCHURE_5PPP_BAT2_PL.pdf
- [16] MongoDB, URL: www.mongodb.com/.
- [17] Prometheus, URL: <https://prometheus.io/>
- [18] R. Bruschi, G. Lamanna, C. Lombardo, S. Mangialardi, "Extending the TRIANGLE testbed towards Mobile Edge Computing", ISWCS 2018, Lisbon, Portugal, 2018.
- [19] VyOS - an Open Source Linux-based Network OS, URL: <https://vyos.io/>.