Universidade Estadual de Campinas
Faculdade de Engenharia Elétrica e de Computação

Claudio Roberto Bertoldo Junior

Performance Monitoring and Benchmarking of Virtualized Telecom Network Functions

Monografia de Qualificação apresentada à Faculdade de Engenharia Elétrica e de Computação como parte dos requisitos exigidos no programa de Mestrado em Engenharia Elétrica. Área de concentração: Engenharia de Computação.

Orientador: Christian Rodolfo Esteve Rothenberg

Campinas
2015

Contents

Chapter 1

Introduction

Network Functions Virtualization (NFV) [1] is happening due to the service providers' needs in creating an open to innovation and cost-effective environment for implementing their services upon their network functions. One of the main concern about implementing VNF is its performance. In public clouds, for instance, is possible to set various entities, as virtual CPU, memory, disk space, network logical topology, firewall rules, but is difficult to have a thorough control of network performance, although the cloud providers make available options for the instances network quality (between pessimistic and optimistic scenarios). Thus, in business models based on IaaS, where there is a "locked down" environment, a detailed study/performance assessment is needed to forecast how a particular function will behave in this environment, often considered dynamic and closed. The question is to assess whether public clouds (or even private ones) are carrier-grade-ready or at least allow customization/optimization for that. Through [2], for example, is possible to better understand this "locked down" environment of the most popular public cloud infrastructures.

Existing related works are limited to performance evaluation and improvement of generic clouds, many times focused only on web applications (see Sec. 2.2) and having no integration with upper components, such as the Orchestrator in NFV-context. The intention is the same: evaluate infrastructure and/or application performance in order to optimize it or support decision.

This work, in addition to others in NFV field and specifically synergistic to the Next Generation Mobile Networks (NGMNs), moves on together with the conceptual evolution of the technology. Within this context, some challenges are characterized concerning the reliability of the model, due to its prematurity. The proposal aims to address the problems of the "low-developed" and open-to-innovation environment faced by Mobile Network Operators (MNOs), particularly with regard to development agility and functions implementation (deploy and test). If successful, it will contribute to the demystification of the proposed model and to the evolution of the model acceptance and adoption by MNOs worldwide. In this field, the efforts and initiatives of the Industry Specification Group (ISG) for NFV of the European Telecommunication Standard Institute (ETSI)[1] can also be highlighted. Initiatives related to the VBaaS framework, introduced by [3], and the Use Case (UC)/Proof of Concept (PoC) itself deserve attention and will be treated in the summarized literature review. More specifically, this work addresses an issue that is currently being discussed in the academy and industry: the performance of general-purpose applications and network functions running over cloud (public and private), with a view to Quality of Experience (QoE), compliance of required/negotiated SLAs and cost reduction with computing and networking resources, i.e., Capital Expenditure (CAPEX) and

[1] http://www.etsi.org/technologies-clusters/technologies/nfv

Operational Expenditure (OPEX).

1.1 Problem Statement

In this context, there is a need for specific frameworks for evaluating virtualized environments for network functions. This is understandable due to the maturity of the technology and the key metrics definitions moment for assessment and improvement (optimization), still ongoing by the relevant authorities (ETSI etc.). Generally, this assessment seeks to correlate the results obtained in a full or para-virtualized environment with those known from traditional physical domain (baseline). The benefits of knowing that, concerning optimization initiatives and decision support, are numerous. Starting from the premise that the requested workload is estimated during the network design phase, solution architects have the task of designing the infrastructure to support such traffic. The implementation of a VNF to support legacy traffic of a function previously implemented into a black box ("physical-virtualized migration"), where the values of all its Key Performance Indicators (KPIs) at a given time are known, such as throughput effectively supported (related to bandwidth), number of packets processed by unit of time and maximum concurrent sessions, is an example of activity that requires such study, i.e., prior knowledge of the target infrastructure properties. In short, it is possible to determine through measurements and comparisons whether a given infrastructure (public or private) is able to meet the demands imposed by service providers and VNF-Independent Software Vendors (ISVs).

1.2 Value Proposition

The carriers' dilemma in selecting VNF deployment over private (on premises) or public cloud (third-party's property) involves many economic and geographic factors, as well those related to performance, availability and security. In the case of performance, considering that carriers will deploy their own cloud (private), so what is the purpose of performance assessment, whether, by strategic decision, there will not be at least one peer for comparison? The search for optimization in software-level is the answer.

Thus, the value proposition of this work is to contribute to an under-development benchmarking framework for NFVI (public and private clouds) and VNF assessment in order to bring:

VNF-NFVI Correspondence
Knowledge of the correspondence between a given VNF and its target NFVI, concerning resources allocation (required/negotiated SLA versus available resources/infrastructure) during the pre-VNF deployment phase, i.e., to verify during the benchmarking whether an infrastructure meets the VNF performance requirement;

Optimization Identification
Identification of available platform optimization features in virtual processing and networking level, in private clouds mainly, based on obtained values and compared with baselines, enabling the conclusion whether there is or not such optimization(s) implemented (e.g.: shorter Virtual Machine (VM) CPU time slices in the context of vCPU scheduling in Xen[2]/physical CPU sharing [4]; and Transmission Control Protocol (TCP)

[2] http://www.xenproject.org/

performance improvement with Xen through TCP acknowledgments offloading to the driver domain [5]);

Baselining
: In order to support decision about choosing between the private cloud implementation and the public cloud rental (or even the best Point of Presence (PoP), rack or node), complementing benchmarkings based exclusively on economic parameters. Hereupon it is important to highlight the inflexibility of public infrastructures for optimization, that can be configured as an entry barrier for cloud provider in the NFV market;

Benchmarking
: In order to support decision about choosing the best NFVI provider, inside the "test before deploy" context, also complementing benchmarkings based exclusively on economic parameters. It's expected to justify values discrepancies obtained through the underpinning environment characteristics. An example of it is the employment of Xen by Amazon, where its default settings, theoretically immutable, of vCPU scheduling and TCP acknowledgments not being offloaded to the driver domain [5]), do not favor its choice as NFVI for "demanding" functions;

Fault Tolerance
: Support for NFVI fault tolerance mechanisms, generally implemented in the NFV-Management and Orchestration (MANO) component, that correlates performance degradation, such as low throughput and high latency, with failures. It's related to a probe role and not related to an "pre-deployment" benchmarking, but the proposed application can be easily extended to it.

1.3 Expected Results

The expected results of this work are:

1. Development of a suitable methodology for NFVI performance assessment (generic metrics related to computation, network, memory and storage) and also evaluation of services offered by a particular VNF (specific metrics);

2. Design of an architecture, based on the methodology of the previous topic, and implementation of a framework for NFVI and VNF assessment (contribution the VBaaS project). The main idea is to automate the tasks of provisioning and testing/benchmarking different infrastructures and create a knowledge plan that support decision-making.

3. Execution of performance benchmarkings (data collection and comparison), through the developed methodology and framework (PoC), over private cloud (deployed testbed), correlating the obtained outcomes for NFVI and VNF metrics (over VMs or even containers) and then comparing between different virtualized infrastructures.

4. Last but not least, results analysis and sharing with the academic community and interest groups in NFV performance benchmarking and NFV for NGMN, thus contributing to the model demystification for early adopters service providers. Therefore, the aim of this study is to compile this information with the final goal of contributing incrementally with the evolution of technology as a whole, regardless of the frameworks and enabler technologies employed. Eventually, new features can be raised and proposed, while focusing on performance evaluation and enhancements.

Chapter 2

Literature Review

2.1 Background

NFV is a trend that aims to leverage an increasingly innovation-friendly environment, compared to the current scenario, being able to allow new functions to be quickly tested and integrated to those already existing and allowing customization in software-level of several network elements. From an economic standpoint, this paradigm introduces a new approach compared to existing middleboxes that integrate hardware and software into a single solution (appliance). Service Providers will have the flexibility to separate hardware and software, share Information Technology (IT) infrastructures (through virtualization), use open source software and COTS hardware solutions. Considering the current scenario, where service providers' profit margin is more and more decreasing and the demand for data traffic is increasing, mainly due to the rise of Over-The-Top (OTT) services, it's necessary to develop creative solutions that contribute to the investments optimization, that exceed the users' expectations and that has a median learning curve. The consulting company "Mind Commerce" estimates that NFV global investments will grow up at a Compound Annual Growth Rate (CAGR) of 83.1% between 2015 and 2020. The revenues will reach $ 8.7 Billion by the end of 2020[1]. "Dell'Oro Group" believes that the market can represent US$ 2 billions in equipment sales by 2018[2]. "Research Doyle" estimates a more optimistic scenario, in which the technology will reach a market of US$ 5 billions by 2018, including software, servers and storages[3].

Currently, we notice the evolution of the fourth generation of mobile networks (Long Term Evolution (LTE)/System Architecture Evolution (SAE)) on global scale and the first steps of fifth one. Like other technologies of mobile networks standardized by 3rd Generation Partnership Project (3GPP), the 4G's topology is basically segmented into wireless access network (Evolved Universal Mobile Telecommunications System Radio Access Network (eUTRAN)) and core network (EPC), entirely Internet Protocol (IP)-based. Complementary, still inside mobile networks context, the IMS aims to enable Voice over LTE (VoLTE). The high complexity of both system's O&M (considered "closed") and the low capacity for customization and innovation turn them the most biased in order to implement its functions in a virtualized environment. "Infonectics" consulting consulted several carriers about the beginning of NFV applications. 59% answered that they intend to deploy virtualized EPC (vEPC) by 2016 or later, and several also reported about plans for deploying virtualized IMS (vIMS) to support VoLTE. [6]. Finally, it is concluded that, as well as EPC, the implementation of IMS functions

[1] Mind Commerce, "NFV Market: Business Case, Market Analysis and Forecasts 2015 - 2020", 2015.

[2] C. Matsumoto, "NFV Market Size: How's $2B for a First Guess?", 2014.

[3] L. Doyle, "Forecasting the NFV Opportunity", 2013.

in virtualized environments brings several benefits to MNOs, such as flexibility, scalability and cost savings.

In the NFV context, the computing and networking performance of the NFVIs is considered the biggest technical barrier in the migrating process from physical functions to virtualized ones, i.e., the search for carrier-grade performance is one of the biggest challenges of the technology. Cloud and Communication Service Providers aims at optimizing hardware resources in view of high capacity systems that are cost-effective, consume low energy and are physically compact.

In order to optimize or just to deploy virtualized network functions, such as those that constitute an IMS, it is essential to know the characteristics of the virtualized environment (NFVI). Achieving benchmarkings for choosing the best provider/platform or just to support the decision to deploy a private cloud is considered one of the most important stages of the technology proposed roadmap. Sub-sizing means bottleneck insertion and, on the other hand, super-sizing means waste of resources (financial, mainly). These benchmarkings are based on workload testing (stressing) over the network/interfaces (generation of varying packets sizes and the their transmission via different protocols, for example) and computational resources.

2.2 Related Work

There are several related works, from industry and academia, that aims to analyze metrics, methods and performance benchmarking frameworks for cloud infrastructures (automated cloud test), such as Cloud WorkBench [7], CloudBench [8], Expertus [9], Cloud Crawler [10], CloudCmp [11], C-Meter [12], Cloud-Gauge [13], CloudSuite [14] and Yahoo YSCB [15], besides of the recent Google Perfkit Benchmarker[4]. These frameworks are not specific for NFV performance benchmarking. They are generally test-oriented for web-based applications.

The most related work as a whole is [16], which adopts the open-source Clearwater vIMS[5] VNF for evaluating. A so-called "monitoring engine" was developed in order to monitor the system's performance (application and infrastructure). Differently from the work proposed here, that has the same VNF, but just as an UC, the author has proposed a framework fully based on Clearwater Project. Rather than employing the Clearwater's "All-in-One" approach (used here and detailed throughout this document), the "A-Node-per-VM" approach was adopted. Concerning the UC, the methodology and tools are very similar.

The scope of this work:

- Relates to others cited due to the similar objective of evaluating "as-a-Service" applications and the computational infrastructure that supports them, through the design and implementation of monitoring and benchmarking frameworks based on cutting-edge tools;

- Differentiates from the most part of the cited ones because they are explicitly web-oriented, i.e., instead of targeting network applications (or generic ones), such as data forwarding functions, they aim to evaluate web applications' performance, correlating with cloud computing infrastructure's performance (processing, database etc.).

The tables below (2.1 and 2.2) introduce works related to the motivation, methodology, similar frameworks (but web-oriented), virtualized infrastructure performance and other UCs.

[4]https://github.com/GoogleCloudPlatform/PerfKitBenchmarker

[5]http://www.projectclearwater.org/

Table 2.1: Related Work Taxonomy (Methodologies and Benchmarking Frameworks)

Area	Work	Approach	Gap(s)	Useful
Methodology	[17]	**Compilation of methods, modern architectures and recent implementations that enable cloud testing.**	**It does not introduce some new benchmarking frameworks (chronological mismatch).**	**Exploited for insights concerning the architecture design here proposed.**
Methodology	[18]	**Introduces a set of metrics that supports decision-making about the best cloud provider, based on cost-benefit analysis.**	**It does not make an analysis about the candidate benchmarking tools.**	**Reference for the evaluation metrics compilation proposed here for NFVI.**
Methodology	[19]	**Methods and implementations for cloud testing clearly focused on web-based apps.**	**Purely theoretical, it is clearly focused on web-based apps.**	**Interesting comparative table on the different types of tests and their respective focuses.**
Benchmarking Framework	[7]	**Introduces an implementation itself and a case study oriented to storage performance analysis in public cloud.**	**It is only focused on storage perfomance of a public cloud provider.**	**Architecture and open-source code for reusing and getting insights.**
Benchmarking Framework	[8]	**Concern with the correspondence between application and infra and optimization goals.**	**Not identified.**	**It brings a high-level metrics approach, such as VM provisioning throughput and latency.**
Benchmarking Framework	[9]	**Highlights the benefits of cloud testing automation.**	**Unavailable code to evaluate the framework.**	**It addresses clearly the separations of application versus platform and deployment versus run time.**
Benchmarking Framework	[10]	**Focused on an architecture for describing and automatically executing application performance tests in IaaS.**	**Unavailable code to evaluate the framework.**	**Introduces a so-called "Crawl" Domain-Specific Language (DSL), an engine called "Crawler" (based on RESTful web service) and an UC.**

Table 2.2: Related Work Taxonomy (Virtualized Infrastructure Performance and Use Case)

Area	Work	Approach	Gap(s)	Useful
Virtualized Infrastrucuture Performance	[20]	Benchmarking analysis (load generator vs. System Under Test (SUT)): Physical Machine (PM) vs. PM, PM vs. VM and VM vs. VM.	It is limited to an UC with only one virtualization technology involved.	This study is a root cause analysis that justifies the poor performance of a given scenario.
Virtualized Infrastrucuture Performance	[4]	Proves the relationship between network performance and CPU utilization.	It is also limited to an UC with only one virtualization technology involved.	The impact in performance of a non-NFV-optimized environment.
Use Case	[21]	NFVI reliability evaluation and benchmarking, based on fault injection.	Focused only on NFVI reliability (availability/resilience) evaluation and benchmarking.	Its Use Case exploits the open-source VNF vIMS Clearwater over COTS server and VMware ESXi hypervisor[a].
Use Case	[22]	Methods and implementations for cloud testing clearly focused on web-based apps.	Unavailable oficial documentation (no details about its implementation).	It also exploits the open-source VNF vIMS Clearwater.
Use Case	[23]	Overview on the IMS technology, the SUT (vIMS) and Test System, used to start several scenarios and then to stress the SUT.	Not focused on virtualization. All implementations were performed over Bare Metal nodes.	Introduction of the IMS Bench SIPp tool[b].

[a] http://www.vmware.com/products/vsphere-hypervisor
[b] http://sipp.sourceforge.net/ims_bench

Chapter 3

Methodology

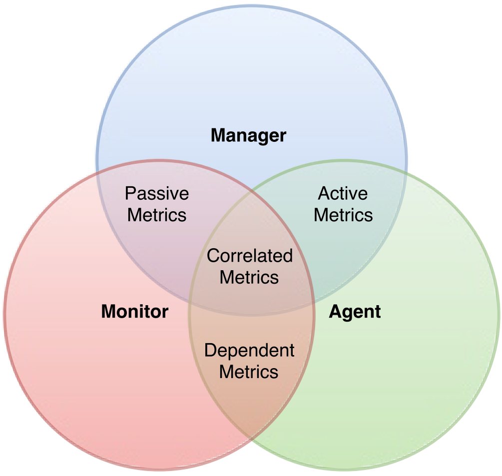
The methodology to achieve the expected results of the project as a whole is divided into three parts: (i) definition of metrics and tools that are core of the proposed framework, (ii) design and implementation of the framework itself, and (iii) execution of an UC/PoC to test and prove its effectiveness. The design and implementation phase is summarized in the VBaaS framework development itself. Implementation englobes Operation System (OS) compilation (images), API development, their integration and testing etc. The evaluation phase (PoC) is summarized in ensuring the efficiency and effectiveness of this framework through a UC, which additionally aims to correlate performance results in VNF and NFVI and eventually triggering optimization. The specific assessment (through specific metrics and tools) is hardly reused because each function has its standard features, but the proposal is to highlight the concept of VNF benchmarking by the methodology independently of the services provided by it.

3.1 Design and Implementation

The framework architecture is based on performance metrics to be collected from VNF Under Test (VUT) and NFV Under Test (NUT). The framework is a Virtualized Network Function Forwarding Graph (VNF-FG), as well as VUT itself. The VUT has specific metrics (Layer 5-7), plus generic metrics ("conventional" ones) that comprise computation, network and storage resources. The metrics and KPIs definition for performing benchmarkings should take into consideration two relevant domains:

- Instance itself (generic metrics), being it bare metal or full or lightweight virtualized and it is related to machine in-loco (CPU, memory and storage) and network resources (Layer 2-4);
- Application, i.e., VNF that covers a wider range of specific metrics (Layer 5-7) dependent of the variety of network functions candidates to the virtualization (e.g. SIP Proxy and SAE-GW).

A full benchmarking should include the generic metrics (conventional ones, i.e., application/VNF-independent), besides of having the ability of crossing them with a view at forming KPIs that will support decisions. With the architecture, metrics and default tools definition, the framework implementation covers the development of the three main building blocks (Manager, Monitor and Agent, detailed hereafter and shown on Figure 3.1 and 3.2) with the same design patterns.



Credits: R. V. Rosa, co-author of [1] and [3]

Figure 3.1: VBaaS Building Blocks' Relationship

Manager and Agent will be one or more lightweight Linux instances (VMs or eventually containers), equipped with several generic metrics evaluation tools (and specific ones, pulled from a local attachable repository), running over an abstracted NFVI, managed and orchestrated by an abstracted MANO. Monitor will be a plugin attached to the NUT due to its goal of collecting in-loco metrics.

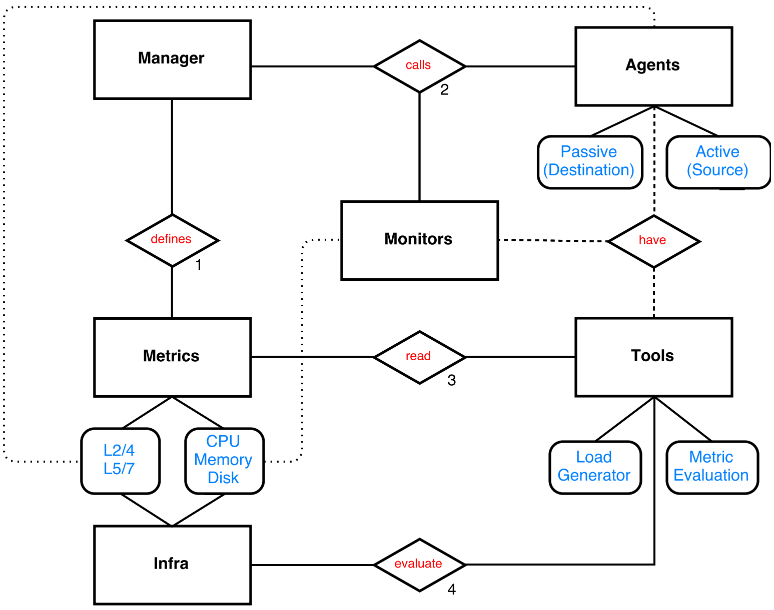


Figure 3.2: VBaaS Building Blocks' Actions and Properties

Although this work is going to be focused only on Monitor building block, Manager and Agent are also documented here.

3.2 Use Case and Proof of Concept

Before telling what this PoC addresses, is better to understand what the expected outputs are with the benchmarking results raised by the proposed framework:

- VNF (specific) and NFVI (generic) metrics values correlation in order to better understand the Infra and then to perform human or machine-triggered optimization and then to reevaluate (decision support);
- Compare different virtualized infrastructures on top of the proposed testbed (private cloud) in concern flexibility and performance itself;
- Select the environment where the incoming VNF fits better based on its performance history, required/negotiated SLA etc.

The assessment phase (proposed UC), besides of demonstrating the VBaaS framework efficiency as a whole, aims to evaluate/correlate the performance results between VNF and NFVI. The NFVI, with the exception to its management and orchestration entities and interfaces with other systems (e.g. Operation Support System (OSS) and Business Support System (BSS), that can be abstracted hereupon, reaffirming what has been outlined previously about disregarding NFV overlying functions, such as MANO), has consolidated metrics that are already employed since the emerging of cloud computing. The application/VNF, in turn, suggests specific metrics, which in the majority of cases are variations of throughput and latency of transactions (sessions) that it performs. I.e., the crucial point in the methodology is the clarity of the purpose to evaluate the testbed, composed by two main domains:

- NUT - generic assessment, where methodology/framework can be reused for independent assessments, independently of the function(s) over it;
- VUT - specific assessment. As an UC, this work will adopt the integrated vIMS function (choice detailed above), thus evaluating specific metrics of the technology.

For choosing the VNF, considering the aforementioned synergy of this work with Next Generation Mobile Networks (NGMNs) and the performance evaluation for different NFVI for determined NFV and the subsequent correlation results, Clearwater vIMS was selected:

- Metaswitch's Clearwater IMS project (a.k.a. Clearwater vIMS): an open source project developed by Metaswitch Networks. As quoted in related work section, this framework is protagonist in the ETSI NFV ISG PoC#1 [22]. It has been widely employed in PoCs of several NFV projects, such as [21].

Testbed

The UC will explore a Private Cloud as NFVI in order to evaluate the VBaaS framework and the VNF-FG (Clearwater) behavior. The document [24] will be exploited as Reference Architecture with a view to implementing an environment (hardware/software) optimized for NFV. This architecture, in turn, suggests state-of-the-art implementations, such as the Open vSwitch accelerated by Intel® DPDK[1], the deployment over OpenStack [2] and the integration with OpenDayLight[3]. The greatest advantage of having a high performance platform is the flexibility to run different virtualization and cloud management solutions and to modify kernel, hypervisor, and virtual switches settings, in view of the environment optimization for NFV. Since OpenStack is turning the de facto VNF VIM and due to its compatibility to AWS,

[1]https://github.com/01org/dpdk-ovs
[2]https://www.openstack.org/
[3]https://www.opendaylight.org/

it is going to be widely explored. The release deployed on Ubuntu Server 14.04.03 is the codenamed “Kilo” (“2015.1.1”, the last stable version when OpenStack was chosen), with KVM[4] as hypervisor.

Based on the VNF Architecture proposed by ETSI, independent of the NFVI-provider (and placement), the NUT and the VBaaS framework is placed as shown in Figure 3.3, with OpenStack as VIM, Clearwater vIMS as NUT and OSS/BSS, Orchestrator and VNF Managers abstracted.

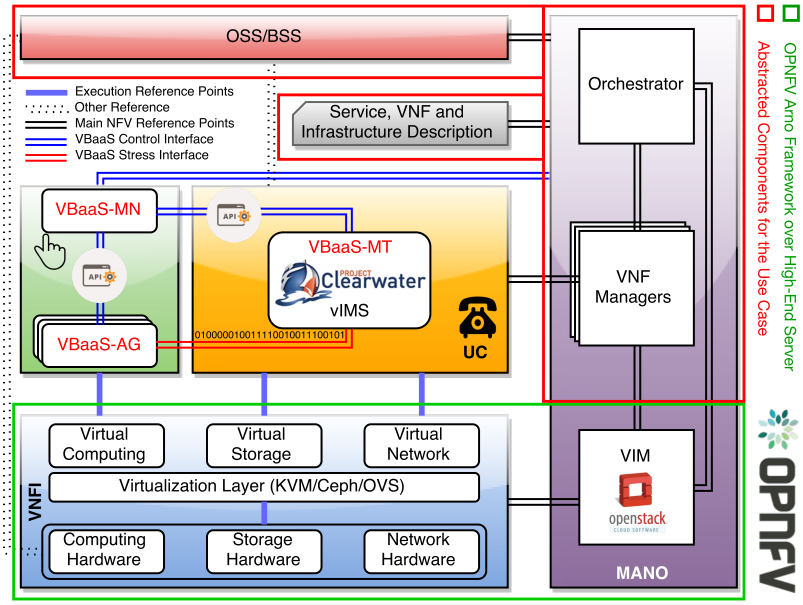


Figure 3.3: UC Based on VNF Architecture by ETSI

Due to the Tiny flavor’s RAM size incompatibility with Clearwater vIMS minimum requirements, it is not going to be considered for the tests. The candidate NUTs (physical and virtualized hardware) for the UC are shown on Table 3.1:

Table 3.1: Candidate NUTs for the UC

Hardware[a]			**OpenStack Flavor**				
CPUs[b]	**Clock (GHz)**	**RAM (GB)**	***Tiny***	***Small***	***Medium***	***Large***	***XLarge***
12	2.1	24	X	✓	✓	✓	✓
32	2.5	64	X	✓	✓	✓	✓

[a] Both servers follow the same CPU/RAM ratio rule of OpenStack’s flavors, i.e., 2 GB of RAM per CPU
[b] The numbers above are multiplied by two due to the Intel® Hyper-Threading feature

In order to make things easier, from now on the server with 12 CPUs will be called “Smaller” and the server with 32 CPUs will be called “Larger”.

[4] http://http://www.linux-kvm.org

Chapter 4

Partial Results

Specially in this project's milestone, the top-down approach is being adopted, i.e., the expected results from the UC is being collect before the functional release of the framework that enables this collection in a fully-automated process.

4.1 Workflow for Partial Results Collection

The flowchart below (Figure 4.1) shows the methodology employed to collect the partial results through a non-fully-automated way, i.e., without taking advantage of the high-level automation introduced by the VBaaS Framework yet.

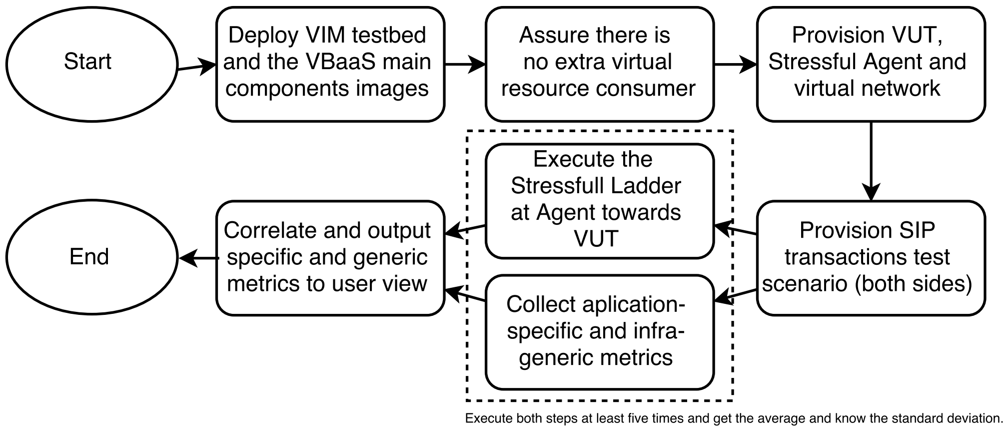


Figure 4.1: Partial Results Methodology

The first step is independent of the next ones and once the testbed and the VBaaS main components images are deployed, they will be useful until the end. The next steps are executed with minimal automation support, what includes manual execution, absence of fine-grained synchronization and so on. Assuring there is no extra virtual instance consuming resources (such as an "out-of-the-circle" concurrent VM) is important to better understand the results in a shared resources environment. The provisioning phase (VUT, "Stressful Agent" and virtual network) is performed manually through VIM's Graphical User Interface (GUI) or CLI. Provisioning the test scenario includes bulk provisioning of "ghost" subscribers in vIMS side and installing and configuring the application load generator in the "Stressful Agent" side. Then, through a "Stressful Ladder", generating SIP Registration Transactions , by SIPp tool, and created by a different instance (XLarge) at same physical node, the "All-in-One" Clearwater vIMS (VUT) process this load while collecting its vCPU usage and the vIMS Edge Proxy process' (Bono) RAM demand for after-correlation. Then VMs and virtual network's capacity are evaluated.

4.2 Data Collection

Follow the results of generating the theoretical "Stressful Ladder" towards the vIMS instance (each OpenStack default flavor, except the "tiny" as explained previously in Section 3.2) and then crossing with the collected CPU and RAM by Bono usage at VUT on both nodes (Larger and Smaller).

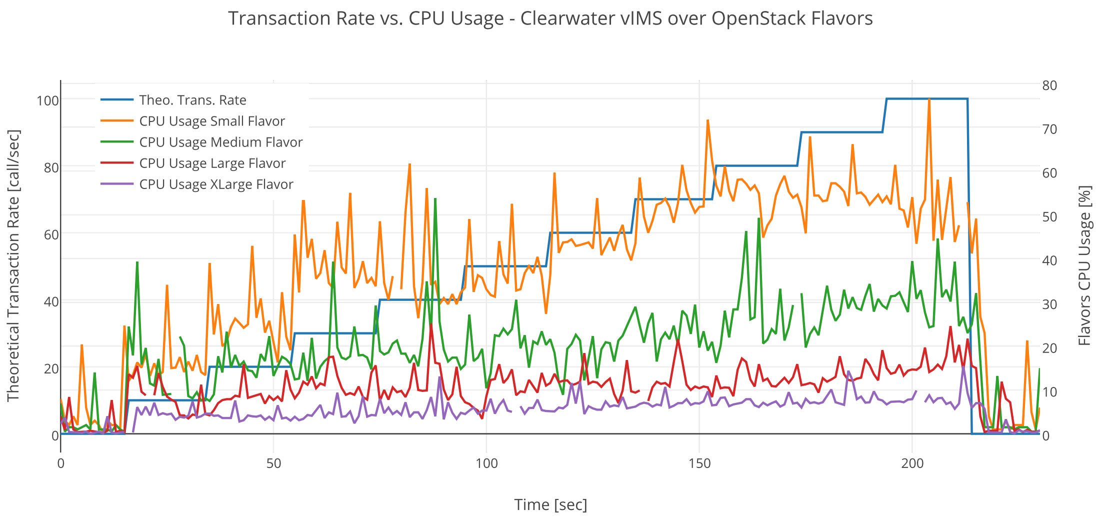


Figure 4.2: CPU Usage - vIMS over OpenStack Flavors over Larger Node

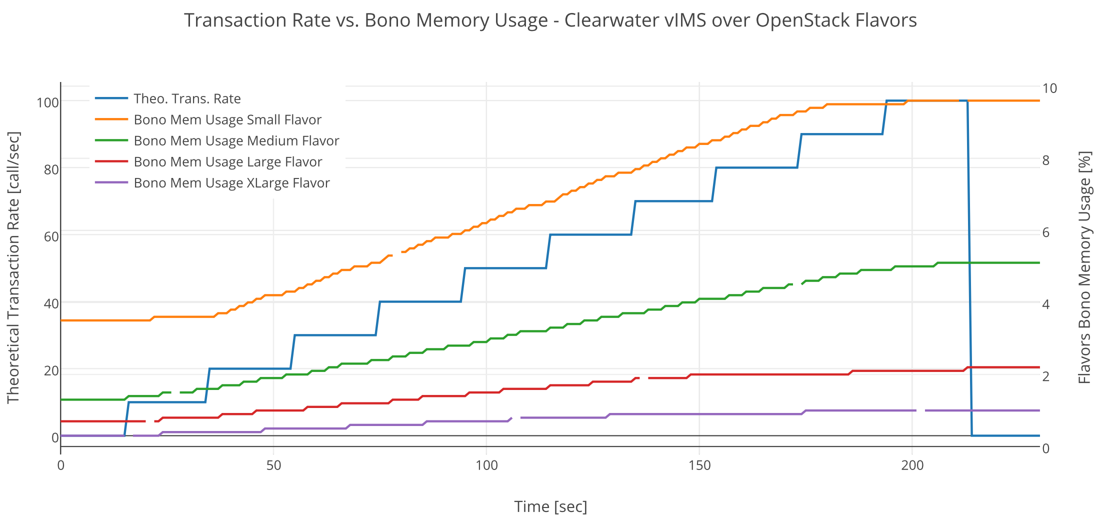


Figure 4.3: Bono Memory Usage - vIMS over OpenStack Flavors over Larger Node

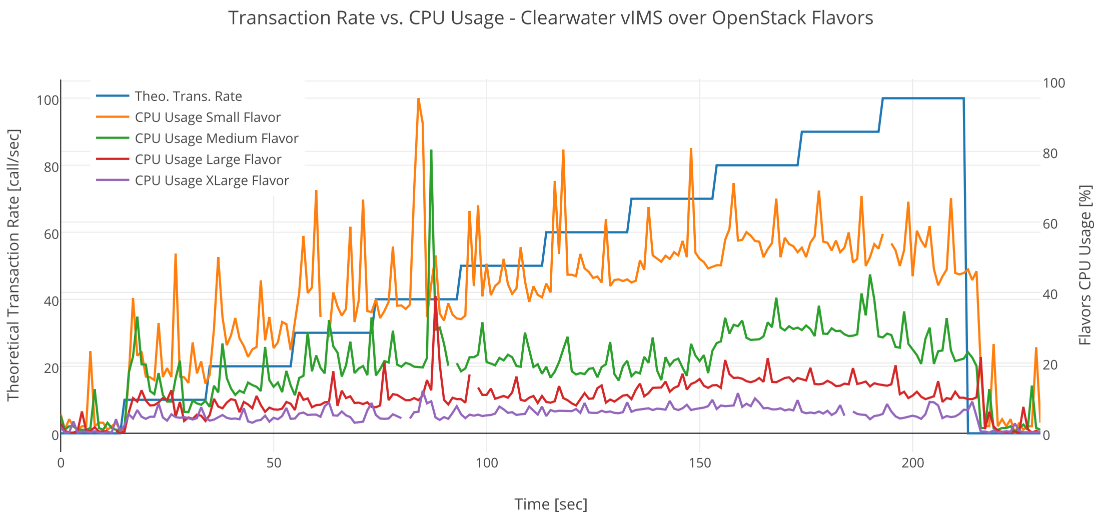


Figure 4.4: CPU Usage - vIMS over OpenStack Flavors over Smaller Node

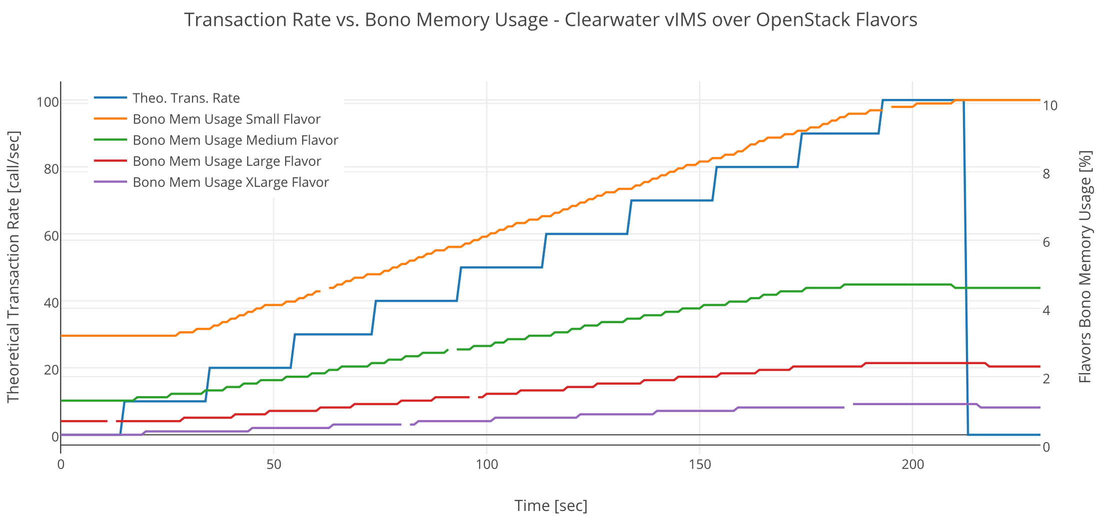


Figure 4.5: Bono Memory - vIMS over OpenStack Flavors over Smaller Node

From this testing scenario definition, it is possible to determine some theoretical limits, such as the maximum number of generated transactions and run time and the test efficiency, as follows:

$$TransactionsAmountLimit = \sum_{n=1}^{p} Ri * Fd * n \tag{4.1}$$

Where p (rate_max Rm divided by rate_increase Ri) is the number of steps of the "Stressful Ladder" and Fd is the increase_interval.

The amount of time spent by the stress testing is represented by:

$$RunTime = \frac{Rm}{Ri} * Fd = p * Fd \quad (4.2)$$

Generic Metrics Collection for Pre-Benchmarking

In order to find a correlation between generic and specific metrics, avoiding specific metrics collection (pre-benchmarking) and enabling conclusions about the VNF considering only NUT's performance, Sysbench[1], a generic benchmarking tool, was ran into a OpenStack's instance configured with a vCPU quantity greater than the quantity of pCPUs of its bare metal host. An expected result, that could not be found in literature, is the saturation when the amount of VM's threads are equal to threads capacity of bare metal host.

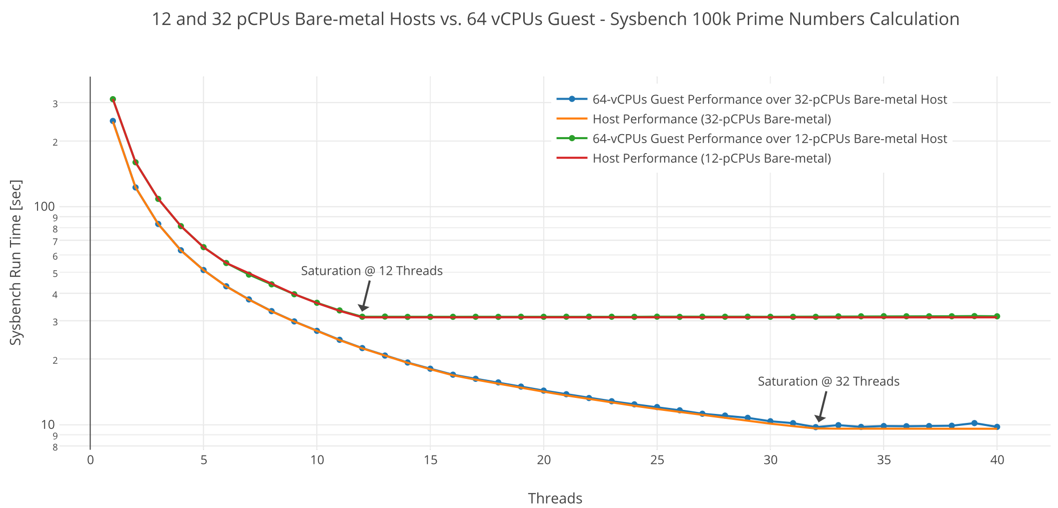


Figure 4.6: Processing Performance of Smaller and Larger Hosts vs. 64 vCPUs Guest

From the chart above (Figure 4.6), it can be concluded:

- In the testbed deployed, the results concerning the equality of both guests and bare metal hosts' curves were unexpected and it proves the OpenStack/KVM's efficiency in terms of vCPU-pCPU resources allocation (virtualization overheads minimization) for the workload generated;

- Unless the specification is to reserve a large number of pCPU time slices (related to the OpenStack's Overcommiting[2]) fot a given VNF in order to ensure a considerable performance when the concurrent load is relatively high, instantiating VNFs with custom flavors with a number of vCPUs greater than the number of pCPUs makes no sense due to the saturation;

- Once guest and bare metal host's curves for each node are overlapped, there is no concurrent workload, i,e., there is no other VNF on the same host consuming processing (may even be instantiated, but without significant consumption). This analysis triggers a new

[1]https://github.com/akopytov/sysbench
[2]http://docs.openstack.org/openstack-ops/content/compute_nodes.html#overcommitting

assumption about the determinism of NFVI's capacity concern concurrent workload on the same physical host, that affects all collection so far concerning VNF-specific metrics and computation ones, collected with no concurrent workload.

Another interesting result was the difference between Sysbench run time when calculating a given amount of prime numbers (50k for instance) for both Small and Large servers and comparing the processing performance between a VM and its bare metal host (Smaller and Larger nodes), using the same amount of threads, equal to the vCPU quantity of the he VM under test. In this case, the test was configured with eight threads in order to compare with the XLarge Flavor's performance (both results for both servers in red on table 4.1 reinforce the affirmation of VIM and hypervisor's efficiency).

Table 4.1: OpenStack Flavors over Smaller and Larger Nodes - Sysbench 50k Prime Numbers Calculation Run Time [sec]

Hardware			**OpenStack Flavor**				*Bare Metal*
CPUs	Clock (GHz)	RAM (GB)	*Small*	*Medium*	*Large*	*XLarge*	*(8 Threads)*
12	2.1	24	122.8	62.34	31.5	17.02	16.98
32	2.5	64	95	48.06	24.66	12.86	12.72

The results, comparing different flavors at the same node, after collecting a first dataset is easily predicted due to its proportionality. It is important from the operator's point of view, to know the limitations (max. thresholds), in terms of application metrics, of its available virtual resources.

4.3 Results Analysis and Lessons Learned

An important pain point raised through these experiments is the trade-off about having an implemented dynamic resource allocation algorithm (that is positive and the essence of virtualization) and then the difficult of benchmarking (results with different levels of workload from different VNFs sharing the same physical resources) vs. having a static resource allocation algorithm implemented (e.g. CPU Pinning[3], that is often under or overdimensioned) and then be able to not worry about concurrent load.

In short, through the collected values for generic (SIP Registration Transactions) and specific metrics (CPU and SIP Proxy's process RAM Usage) is possible to determine whether the NUT is suitable for the target VNF. The test were performed with no concurrency on the NFVI, so, as shown through the generic benchmarkings, the performance of a VM when its vCPU quantity is greater than or equal to the pCPU quantity, is equal to the performance of its bare metal host. It means that in presence of concurrent load (e.g. another VNFs running on the same host) and concerning the OpenStack's default computation resource allocation algorithm (dynamic), the capacity of a NFVI to host a VNF is directly affected. This phenomenon does not occur whether there was an algorithm for static allocation of computation resources. I.e., the dilemma of having or not a deterministic environment touches the dynamic vs. static allocation trade-off. In other words, the non-determinism occurs due to the unpredictable concurrent load. If the goal is really to guarantee predicable performance after VNF deployment, techniques such as CPU Pinning should be considered.

[3]https://specs.openstack.org/openstack/nova-specs/specs/juno/approved/virt-driver-cpu-pinning.html

Chapter 5

Project Plan

5.1 Activity Calendar

For this research project the activities were planned according to below (Table 5.1):

Table 5.1: Monthly Working Plan

			14							15												16	
	Activities	**Status**	06	07	08	09	10	11	12	01	02	03	04	05	06	07	08	09	10	11	12	01	02
01	Bibliographic Review/Project Definition	Completed	■	■	■	■	■	■	■	■	■	■	■										
02	Benchmarking Tools Learning/Evaluation	Completed					■	■	■	■	■	■	■										
03	Use Case Definition	Completed										■	■										
04	Candidate VNFs/VNF-FGs Evaluation	Completed										■	■										
05	VBaaS Architecture Definition	Completed											■	■	■								
06	Qualification Monograph Writing	Completed											■	■	■	■	■	■	■				
07	VBaaS Framework Prototype Development	In progress												■	■	■	■	■	■	■			
08	Paper Writing and Submission (EADCA)	Completed														■	■						
09	Testbed Setup	Completed															■						
10	VBaaS Framework Prototype Smoke Test	Completed																■	■				
11	Proof of Concept/Experimental Evaluations	Completed																■	■				
12	Presentation for Qualification Phase	In progress																■	■	■			
13	Results Analysis	In progress																■	■	■			
14	Master Thesis Writing	In progress																■	■	■	■	■	
14	Paper Writing and Submission (External)	To do																			■	■	
15	Master Thesis Presentation	To do																				■	■

Chapter 6

Conclusions and Future Work

Through the lessons learned, concerning the challenges of non-determinism of performance in a NFV environment, the next steps comprise in seeking solutions to estimate concurrent workload on the same physical host in order to reproduce the real world and to consider it into the VNF-specific and NFVI-generic results correlation. Additionally, the need of proviosining a specific test scenario for each network function or function chain benchmarking is an issue that can be remains open, allowing the operators/developers to implement their own test scenario according to the target VUT, as well as is being done through the proposed UC.

As a whole, this work, through its assessment phase and UC, besides of demonstrating the under-development VBaaS Framework efficiency, aims to evaluate/correlate the performance results between VNF (specific) and NFVI (generic) in order to:

1. Better understand the Infra and then to perform human or machine-triggered optimization and then to reevaluate;

2. Compare different cloud providers, including the internal one (private cloud) in concern to cost, flexibility, performance itself etc. (decision support);

3. Support selection of the environment where the incoming VNF fits better, based on its performance history, required/negotiated SLA etc.

The NFVI has consolidated metrics that are already employed since the emergence of cloud computing. The application/VNF, in turn, suggests specific metrics, which in the majority of cases are variations of throughput and latency of transactions (sessions) that it performs. One of the most relevant challenges of this ongoing work is the diversity of specific performance metrics for VNF assessment due to the diversity of network functions candidates to the virtualization (such as those one that compound an IMS), demanding from framework its adaptability for evaluating entities still unknown. The choice of vIMS as an UC was broadly motivated by [22], [21], [25] and [23].

As future work, the study of elastic and parallelized VNFs assessment can be considered (e.g.: stressing a cluster/VNF-FG and then observing elastic triggers working concerning new physical infrastructures - nodes, racks or even data centers).

Finally, after evaluating and formatting the results, they will be shared with academic community and interest groups in NFV. Herewith we hope to contribute to the demystification of the proposed model and the evolution of the acceptance and adoption of the model by MNOs globally. The evaluation of the project results will be through qualifying examination and dissertation, cited above, according to the guidelines established by the School of Electrical and Computer Engineering at the University of Campinas for the graduate program level master.

Bibliography

[1] R. Rosa, M. Siqueira, E. Barea, C. Marcondes, and C. Rothenberg, "Short course on NFV. Network Function Virtualization: Perspectivas, Realidades e Desafios," (Florianópolis, Brazil), XXXII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), May 2014.

[2] B. P. Rimal, E. Choi, and I. Lumb, "A taxonomy and survey of cloud computing systems," in *NCM 2009 - 5th International Joint Conference on INC, IMS, and IDC*, pp. 44–51, 2009.

[3] R. V. Rosa, C. E. Rothenberg, and R. Szabo, "VBaaS: VNF Benchmark-as-a-Service," in *Fourth Edition of the European Workshop on Software Defined Networks (EWSDN)*, (Bilbao, Spain), 2015.

[4] R. Shea, F. Wang, H. Wang, and J. Liu, "A deep investigation into network performance in virtual machine based cloud environments," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 1285–1293, IEEE, Apr. 2014.

[5] A. Kangarlou, S. Gamage, R. R. Kompella, and D. Xu, "vSnoop: Improving TCP Throughput in Virtualized Environments via Acknowledgement Offload," in *2010 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–11, IEEE, Nov. 2010.

[6] P. Lynch, M. Haugh, L. Kurtz, and J. Zeto, *Demystifying NFV in Carrier Networks*. 2014.

[7] J. Scheuner, P. Leitner, J. Cito, and H. Gall, "Cloud WorkBench - Infrastructure-as-Code Based Cloud Benchmarking," *2014 IEEE 6th International Conference on Cloud Computing Technology and Science (CloudCom)*, pp. 246 – 253, 2014.

[8] M. Silva, M. R. Hines, D. Gallo, Q. Liu, K. D. Ryu, and D. Da Silva, "CloudBench: Experiment automation for cloud environments," in *Proceedings of the IEEE International Conference on Cloud Engineering, IC2E 2013*, pp. 302–311, 2013.

[9] D. Jayasinghe, G. Swint, S. Malkowski, J. Li, Q. Wang, J. Park, and C. Pu, "Expertus: A generator approach to automate performance testing in IaaS clouds," in *Proceedings - 2012 IEEE 5th International Conference on Cloud Computing, CLOUD 2012*, pp. 115–122, 2012.

[10] M. Cunha, N. Mendonca, and A. Sampaio, "A declarative environment for automatic performance evaluation in IaaS clouds," in *IEEE International Conference on Cloud Computing, CLOUD*, pp. 285–292, 2013.

[11] A. Li, X. Yang, S. Kandula, and M. Zhang, "CloudCmp: Comparing Public Cloud Providers," *Proceedings of the 10th annual conference on Internet measurement - IMC '10*, p. 1, 2010.

[12] N. Yigitbasi, A. Iosup, D. Epema, and S. Ostermann, "C-Meter: A framework for performance analysis of computing clouds," in *2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGRID 2009*, pp. 472–477, 2009.

[13] M. A. El-Refaey and M. A. Rizkaa, "CloudGauge: A dynamic cloud and virtualization benchmarking suite," in *Proceedings of the Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises, WETICE*, pp. 66–75, 2010.

[14] M. Ferdman, A. Adileh, O. Kocberber, S. Volos, M. Alisafaee, D. Jevdjic, C. Kaynak, A. D. Popescu, A. Ailamaki, and B. Falsafi, "Clearing the clouds: a study of emerging scale-out workloads on modern hardware," *SIGARCH Comput. Archit. News*, vol. 40, pp. 37–48, 2012.

[15] B. F. Cooper, A. Silberstein, E. Tam, R. Ramakrishnan, and R. Sears, "Benchmarking cloud serving systems with YCSB," in *Proceedings of the 1st ACM symposium on Cloud computing - SoCC '10*, pp. 143–154, ACM Press, 2010.

[16] S. Hiray and M. Vutukuru, "Network Function Virtualization," 2014.

[17] X. Bai, M. Li, B. Chen, W. T. Tsai, and J. Gao, "Cloud testing tools," in *Proceedings - 6th IEEE International Symposium on Service-Oriented System Engineering, SOSE 2011*, pp. 1–12, 2011.

[18] Z. Li, L. O'Brien, H. Zhang, and R. Cai, "On a Catalogue of Metrics for Evaluating Commercial Cloud Services," in *2012 ACM/IEEE 13th International Conference on Grid Computing*, pp. 164–173, IEEE, Sept. 2012.

[19] J. Gao, X. Bai, and W. Tsai, "Cloud testing-issues, challenges, needs and practice," *Software Engineering: An International Journal*, vol. 1, pp. 9–23, 2011.

[20] D. B. Khedher and J. Alomari, "Performance Evaluation of Virtualization Solutions for Telecommunication Applications," no. Vmm, pp. 2–7.

[21] D. Cotroneo, L. De Simone, A. K. Iannillo, A. Lanzaro, and R. Natella, "Dependability evaluation and benchmarking of Network Function Virtualization Infrastructures," in *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, pp. 1–9, IEEE, Apr. 2015.

[22] "CloudNFV™ unites the best of Cloud Computing, SDN and NFV." 2013.

[23] D. Thiß en, J. Miguel, and E. Carl, "Evaluating the Performance of an IMS / NGN Deployment," *Proceedings of the 2nd Workshop on Services, Platforms, Innovations and Research for new Infrastructures in Telecommunications. SPIRIT*, 2009.

[24] "Intel® Open Network Platform Server Reference Architecture: SDN and NFV for Carrier-Grade Infrastructure and Cloud Data Centers." 2014.

[25] G. Carella, M. Corici, P. Crosta, P. Comi, T. M. Bohnert, A. A. Corici, D. Vingarzan, and T. Magedanz, "Cloudified IP Multimedia Subsystem (IMS) for Network Function Virtualization (NFV) -based architectures," *2014 IEEE Symposium on Computers and Communication*, pp. 1–6, 2014.