RAVA - Resource Aware VNF Agnostic NFV Orchestration Method for Virtualized Networks

Faqir Zarrar Yousaf, Carlos Goncalves, Luis Moreira-Matias and Xavier Costa Perez
NEC Laboratories Europe, Germany
Email: {zarrar.yousaf|carlos.goncalves|luis.matias|xavier.costa}@neclab.eu

***Abstract*—This paper presents the proof-of-concept evaluation of a Resource Aware VNF Agnostic (RAVA) NFV orchestration method that is designed to enhance the Quality of Decision (QoD) of a cloud controller by optimizing the life cycle management decisions that it takes in order to manage the resources in a cloud infrastructure (e.g., a data center). The RAVA method proposes a novel concept of deriving the affinity scores for the plurality of resource units with reference to a specific resource unit for each individual Virtual Machine (VM) instance hosting a Virtualized Network Function (VNF). This affinity score, referred to as Reference Resource Affinity Score (RRAS), will enable the cloud controller to perform precise and efficient resource tailoring or dimensioning; and hence will optimize its decisions and actions related to the management and orchestration of the virtualized resources inside the cloud infrastructure. The motivation behind proposing RAVA is to enhance the Network Functions Virtualization (NFV) Management and Orchestration (MANO) system capabilities towards the realization of a Carrier Cloud, an important vision for the future 5G architecture. The evaluation results presented in this paper are based on an OpenStack based proof-of-concept implementation of the RAVA method.**

I. Introduction

A. Background

Network Function Virtualization (NFV) is fast emerging as a promising technology that leverages the concept of cloud technology and virtualization technique to enable the next generation of mobile communication networks. The next evolutionary phase of mobile networks, also referred to as 5G, imposes stringent functional and operational requirements. In addition to further reduction of delays, increased peak bit rates, higher spectrum spectral efficiency, better coverage, and the support of potentially large number of diverse connectable devices, they are required to be cost-efficient, flexibly deployable, elastic, and above all programmable. Achieving these requirements in a cost/operational effective way are critical to maintain the business sustainability of mobile operators worldwide, mainly in light of the ever-growing mobile data traffic on one hand and the stagnant (rather falling) Average Revenue per User (ARPU) on the other hand.

To this effect, the telecom industry is greatly investing efforts to develop a Network Function Virtualization (NFV) technology by leveraging the recent advancements in cloud computing and virtualization techniques. The main concept behind NFV is to virtualize Network Functions (NFs) by hosting them on Virtual Machines (VM) as Virtualized Network Functions (VNF). These VNFs are instantiated on servers (referred to as physical machines (PM)) inside a Network Function Virtualized Infrastructure (NFVI). An NFVI is, in essence, a data center (DC) network having an array of PMs. A VM hosting a VNF (or a set of them) is, by itself, an abstraction of a PM that is assigned specific slice of the underlying resources such as, but not limited to, processing, memory, input/output (I/O) module and storage. A single PM can host tens to hundreds of VMs as long as the underlying physical resources of the PM are able to satisfy the workload demands of the hosted VNFs. A PM has a Virtual Machine Monitor (VMM) that manages the multiple VMs on it and monitors their respective resource consumption.

The NFV technology is expected to provide a carrier-grade virtualized mobile cloud network, which is envisaged as a key enabler for a flexible, scalable, programmable and elastic 5G systems. There are numerous challenges for achieving this [1], and thus an ETSI Industry Special Group (ISG) on NFV has been formed to standardize various aspects of an NFV enabled network. One of the main immediate challenges is the management and orchestration of potentially hundreds of thousands of VNFs in an NFVI that are chained to realize different Network Services (NS). To meet this complex challenge, the ETSI NFV has developed an NFV Management and Orchestration (MANO) framework [2].

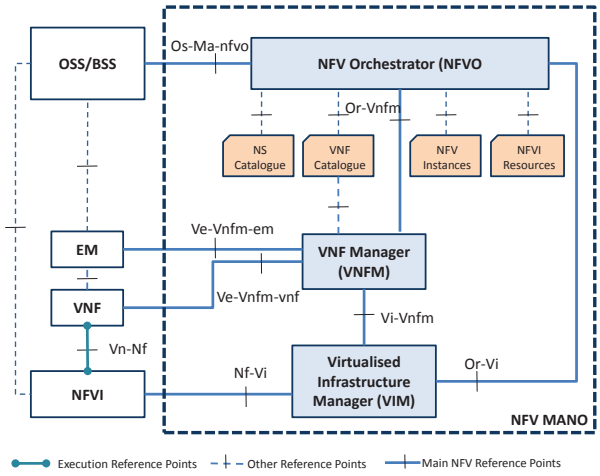


Fig. 1. ETSI NFV Management and Orchestration (MANO) Framework Overview [2].

Figure 1 illustrates the functional architecture of the NFV-MANO framework that has three main functional blocks namely the Virtualized Infrastructure Manager (VIM), the VNF Manager (VNFM) and the NFV Orchestrator (NFVO). The VIM is responsible for the control and management of the NFVI resources on the whole. The VNFM is responsible for the life-cycle management (LCM) operations on the VNF instances, such as VNF instantiation, migration, and scaling.

The NFVO, on the other hand, is not only responsible for the LCM of the NS but it also orchestrates the NFVI resources across multiple VIMs.

B. Problem Statement

When deploying a VNF, a Cloud Service Provider (owner of NFVI) typically offers a set (or menu) of flavors to the VNF provider. The term flavor, in the OpenStack terminology [3], refers to available hardware configuration block for a VM. Each flavor has a unique combination of disk space, memory capacity, and priority of the CPU time (i.e., processing requirement). VNFP will choose the flavor that best matches the functional/operational requirements of the respective VNF. Once selected, the NFV-MANO will coordinate the instantiation of VM(s) on selected PM(s) and assign/allocate resources based on the selected flavors to the VM(s) that will eventually host the VNF(s). Besides the resource availability, the selection of suitable PMs also takes into account specific constraints stipulated in the VNF Descriptor (VNFD) that may be unique to its functional/operational requirements. In essence, the process of hosting/instantiating/deploying VNFs on VMs and also the creation of VMs is, at an atomic level, a resource assignment/management process. However, the static process by which VNFPs choose flavors has inherent limitations leading to performance issues such as:

1) The VNFP may be forced to choose a flavor that may exceed its requirement. This will not only increase the cost of hosting, but may also result in the under-utilization of the underlying resources that are pinned to the particular flavor.
2) Such a process does not take into account the unforeseen traffic/load surges that may have serious impact on the Quality of Service (QoS) of the respective hosted VNF. The system may then be triggered to perform costlier operations of VM migration, cloning, and scaling to meet the traffic demands.
3) Such a process does not take into account unforeseen traffic reductions that may result in under-utilization of the assigned resources.
4) Considering the strict resource allocation and isolation policy, it also prevents the sharing and run-time/dynamic (re)allocation and/or (re)organization of the un-utilized capacities of the underlying physical resources that is pinned to other VMs on the same PM.

Hence, it follows that VM deployment based on rigid assignment of resources may result in non-optimized utilization of the underlying resources. Furthermore, high load surge conditions may prompt the NFV-MANO system to perform specific LCM operations (e.g., VM migration, cloning, or scaling) in order to handle the load conditions. Such management actions are costly and can result in service degradation and non-optimum utilization of the resources. The objective of this paper is thus to propose a remedy to the above limitations by designing a method/system that will:

1) Optimize the LCM operations in a large scale DC.
2) Enable optimal utilization of the assigned and available resources.
3) Enable dynamic and precise (re)assignment, (re)distribution, (re)allocation, (re)organization, and sharing of resources amongst multiple VM instances on the same PM at run-time and at a fine granular level.
4) Minimize the occurrences of the above-mentioned costly VM management operations.

In view of the above objectives, we introduce a fine-granular Resource Aware VNF Agnostic (RAVA) method that can be utilized by the NFVO for making informed and optimum management decisions at run-time in view of changing workload conditions. This paper provides the first proof-of-concept results of the proposed method.

The remainder of this paper is organized as follows. Section II presents the related work, while Section III summarizes the proposed RAVA method. Section IV presents the notion of Quality of Decision (QoD) and evaluates the performance of the RAVA in terms of enhanced QoD based on an experimental test-bed. The conclusions are presented in Section V. For the sake of readability, the list of all abbreviations used in the paper is provided in Table 1.

TABLE I
LIST OF ABBREVIATIONS USED IN THIS PAPER

Abbreviation	Name
AE	Analytics Engine
ARPU	Average Revenue Per User
AS	Affinity Signature
DC	Data Center
DE	Decision Engine
I/O	Input/Output
ISG	Industry Special Group
LCM	Life Cycle Management
MANO	Management and Orchestration
NFV	Network Function Virtualization
NFVI	NFV Infrastructure
NFVO	NFV Orchestrator
NIC	Network Interface Card
NS	Network Services
PM	Physical Machine
QoS	Quality of Service
QoD	Quality of Decision
RRAS	Reference Resource Affinity Score
RU	Resource Unit
VIM	Virtualized Infrastructure Manager
VM	Virtual Machine
VMM	Virtual Machine Monitor
VNF	Virtualized Network Function
VNFC	VNF Components
VNFD	VNF Descriptor
VNFM	VNF Manager

II. RELATED WORK

Successful creation of cloud-based mobile core networks largely depends on how efficiently the underlying VNF LCM operations are performed. The LCM operations such as VNF instantiation, migration, and scaling involve VNF placement across its respective NFVI in a service optimum manner. At the atomic level, this is a resource management problem, which becomes more complex when deploying a NS. This is because a NS is formed by chaining multiple VNFs, where a VNF may be decomposed into multiple VNF Components (VNFC). There are also strict functional relationships among the VNF(s)/VNFC(s) and performance constraints that must be considered when chaining. This gives rise to a multi-dimensional problem making VNF placement more complex [4].

There is a large library of research work that has been conducted for decision on VM placement, resource allocation, and VM management having, as objective, cost savings from better utilization of computing resources and less frequent overload situations. For instance, in [5], performance isolation (e.g., CPU, memory, storage, and network bandwidth), resource contention properties (amongst VMs on the same physical host), and VMs behavioral usage pat-terns are taken into account in decisions on VM placement, VM migration, and cloud resource allocations.

There is also a body of work that proposes different optimization methods that rely on monitoring and somehow responding to the resource utilization metric to trigger the respective optimization method/technique. For instance, in [6], a load balancing scheme is proposed by making VM migration decisions, whereby the proposed scheme suggests the migration decisions to be based on some balancing metric, namely the utilization value of some specific resource unit (RU), or a set of RUs, that the system is able to monitor. It provides details of an iterative greedy method for migrating VMs based on the balancing metric for achieving load balance, but it does not describe any method or system regarding how the balancing metric should be computed/derived and/or how to quantify and/or rationalize the monitored utilization values in making migration decisions, which being the scope of this paper. Furthermore, the scope of [6] is limited to only migration operation whereas the scheme, proposed herein, describes how to rationalize and quantify the utilization values of the plurality of resources and is not limited to making migration decisions but can be used towards making any virtual infrastructure management decision such as cloning and scaling.

Another related work is reported in [7], which proposes a generic model based on resource utilization information for making VM placement decisions of hosting intercommunicating VMs on the same or different PMs. However, the decision model is based on the prediction of the estimated CPU utilization, and for that purpose, the work proposed in [7] performs the benchmarking of the CPU utilization with respect to different workloads through repeated experiments. Again, the scope of [7] is limited to making placement decisions of only intercommunicating VMs, which in turn depends on prior benchmarking of CPU utilization in order to make estimated predictions of CPU utilization. Due to prior benchmarking, this method is neither feasible nor suitable in the context of highly dynamic NFV system where workloads vary dynamically and new VNFs may be introduced and old ones taken out depending on NS requirements.

Thus regardless of the objective, any placement algorithm has to take into consideration, at an atomic level, the utilization of the virtualized resources assigned to the respective VNF(s). In the above cited work, the decisions are made with respect to the utilization of a single resource unit (RU), without taking into consideration its effect on other RUs. Moreover, none of the work takes into account the impact of a management decision on other VNFs.

Mining historical patterns to optimize a decision support system has been exhaustively explored in several applications across different industries. (e.g. transportation).

In contrast to the above mentioned work, the presented RAVA schema encloses a novel method of inferring resource usage by mining such historical trends. Moreover, it also takes into consideration the impact of the decisions on other VNFS during run-time.

III. RAVA - Conceptual Overview

The conceptual details of RAVA has already been provided in [8], which we will summarize here in this section.

The concept of affinity is central to the RAVA orchestration method, whereby the term affinity refers to the correlation between different RUs. The affinity value, or affinity score, is a vector quantity that indicates the level or degree of dependence of one or more RUs on a reference RU (RRU). This method derives and communicates information depicting the correlation, or affinity, between different RUs with reference to a specified RU under different workload conditions. This derived vector is referred to as Reference Resource Affinity Score (RRAS).

RRAS provides an insight on how and by how much the utilization of an RRU will impact the utilization of other RUs. RRAS thus expresses the correlation, or the level of dependence, of an individual RU on the reference RU in terms of utilization. For example, the RRAS value of an I/O resource with reference to CPU indicates the degree of its utilization dependence on the CPU utilization. A high RRAS value would indicate a strong affinity, whereas a small value will indicate weaker affinity or dependence. The RRAS value computation is done by the cloud management and orchestrator entity, for example an NFVO, for all VMs deployed in an NFVI at run-time. The RRAS values enables the NFVO server to make informed decisions in terms of optimum resource management during run-time under different workload conditions regardless of the type of VNF, hence resource aware and VNF agnostic.

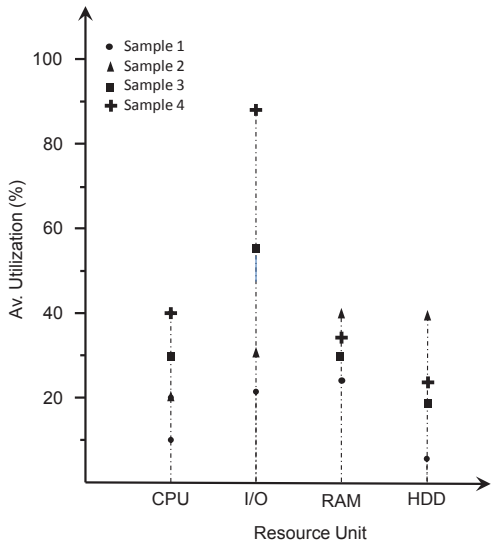


Fig. 2. Resource Utilization samples for a specific VM.

As an example, assuming a linear relationship between the utilization of the two dependent RUs, we use the Pearson product-moment correlation coefficient (PPMCC) as a measure of linear correlation between the RU and the RRU. The

Pearson coefficient (r) is calculated over the samples of average percentage utilization of the different RUs during a specified evaluation epoch $t_{eval} = n * t_{mon}$, where t_{mon} is the monitoring epoch during which the system monitors (x) samples of the absolute utilization (μ) of the different RUs, and calculates the average percentage utilization ($\overline{\mu} = \frac{\sum_{i=1}^{x} \mu_i}{x}$) for the RUs. The values of n and x is a design choice. Thus over a single t_{eval} we will have n number of samples,where each sample is an average utilization of an RU during t_{mon}. A Pearson co-efficient (r) is then derived over these samples *with reference to a RRU* and the derived value of r is the RRAS. A conceptual notion of this can be explained from Figure 2 depicting samples of $\overline{\mu}$ for each of the respective RU for a VM during t_{mon}. The RRAS value, which is the Pearson co-efficient r, is then computed for the different RUs with respect to the CPU, which is the RRU. Table II shows a single RRAS report instance corresponding to the example utilization values shown in Figure 2. The RRAS report depicts the RRAS for each RU with reference to a specific RRU based on the samples of average resource utilization.

TABLE II
RRAS REPORT SNAPSHOT FOR A SPECIFIC VM

Reference Resource Unit	Average Utilization (%) Samples	CPU	I/O	RAM	HDD
CPU	[10, 20, 30, 40]	-	0.98	0.4	0.05
I/O	[20, 35, 55, 90]	0.98	-	0.34	-0.01
RAM	[25, 40, 30, 35]	0.4	0.34	-	0.94
HDD	[17, 40, 20, 25]	0.05	-0.01	0.94	-

Being vector quantities, the RRAS values show how much the utilization of an RRU is impacting the other RUs. A high positive or high negative RRAS value of a particular RU indicates a strong correlation of its utilization with reference to the RRU. On the other hand, a low positive or negative value indicates a weak correlation of a RU with an RRU. Moreover, a high positive RRAS indicates a "strong affinity" while a high negative RRAS indicates a "weak affinity". This will help enable the NFVO to precisely determine the influence of a RRU on the other RUs. For instance, with CPU as a RRU, it is observed from Table II that the I/O RU has a very strong correlation and thus a strong affinity with the CPU, while the correlation of Memory and HDD RU with the CPU is very weak. In other words, the I/O module will experience a higher degree of utilization than the storage with respect to CPU utilization. This could be indicative of a VNF that may perform packet forwarding and routing. In other words, the NFVO can determine the VNF's functional/operational profile by observing the RRAS values.

The NFVO maintains the past RRAS reports for a specific RU or a set of RUs. The period of history can range from minutes to hours or even days, depending on the policy. Such historical/past record of the RRAS report enables the NFVO to derive Affinity Signature (AS) of RU(s) with respect to a RRU. An AS is a plot of the successive RRAS values for a RU, which can then be manipulated by deriving statistics such as affinity trend, as will be seen in Section IV. The AS, and the affinity trend, provides the NFVO the information about the *long term* affinity of a RU with a RRU for a VNF. This information will enable the NFVO to make informed and optimum management decisions, for example selecting the best possible PM to which a target-VNF should be migrated or scaled to. This will thus potentially improve NFVO's Quality of Decision (QoD). The notion of QoD is explained in Section IV.

One issue that may occur with our stated approach is that linear regression will give a more precise trend if the moving average of our process is monotonically increasing or decreasing. Otherwise we need more complex models to assess the trends, stationarity and seasonality of the samples. Since this paper presents the first results of the proof-of-concept, we are at present working towards developing more complex models and considering more elaborate scenarios.

IV. PERFORMANCE ANALYSIS

In this section we provide the first evaluation of the RAVA NFV management and orchestration method. The evaluation results are based on a proof-of-concept test-bed that has been developed using the OpenStack platform [3]. In general, RAVA method is designed to enhance the QoD of a cloud controller responsible for making life-cycle management decisions.

The main objectives of the evaluation results are to demonstrate the potential capability of RAVA towards enhancing the QoD of a cloud controller when it performs lifecycle management actions, like migration of a target-VNF (or Target-VM) to a suitable target-PM. The QoD is measured in terms of the following two mutually dependent criteria:

1) How resource efficient the management action is. The resource efficiency is in turn measured in terms of:
   - Whether both the long term and short term resource requirements of the target-VM will be fulfilled in the target-PM.
   - How non-intrusive a management action has been for other VMs that are already provisioned in the target-PM. That is, to what extent will the target-VM affect the performance of other VMs in the target-PM in terms of resource availability.
2) Number of times the management action has to be executed before the most-suitable PM is determined to live migrate/scale the target-VM to.

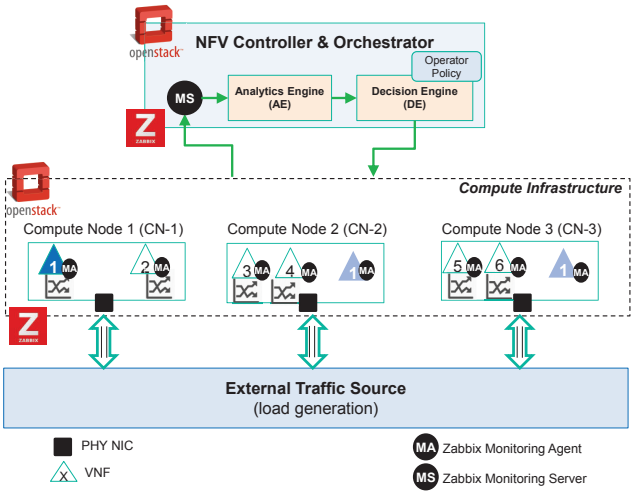


Fig. 3. Testbed setup

TABLE III
METRICS COLLECTED BY RAVA

Metric Name	Description
type	Node type (i.e., VM or Physical Host)
id	Node Identifier
cpu_cores	Number of CPU cores
cpu_idle	CPU idle time (in %age)
cpu_util	CPU utilization (in %age)
net_in	Ingress network traffic (in bps)
net_out	Egress network traffic (in bps)
net_speed	Link speed of primary NIC (in Mbps)
net_util	Network utilization (in %age)
mem_total	Total memory (bytes)
mem_available	Available memory (in bytes)
mem_util	Memory utilization (in %age)

A. Testbed description

Figure 3 depicts the experimental setup used for evaluating the QoD performance of RAVA management and orchestration scheme. The testbed consists of an OpenStack platform with 3 compute nodes, a controller node and an external traffic generator. The controller and the compute nodes run OpenStack 2015.1.1 (Kilo) release and each compute node has VMs that are configured as TCP/UDP servers, while the traffic generator is configured to have multiple clients sending TCP/UDP traffic towards the respective servers using iPerf. The 3 compute nodes share the same hardware specifications (Intel i5-3320M 2.60 GHz CPU, 8 GB RAM, 128 GB SSD, Intel Ethernet 1 Gbps) while the controller node has similar ones (Intel i5-3472U 1.80 GHz CPU, 8 GB RAM, 128 GB SSD, Intel Ethernet 1 Gbps), and all VMs have the same flavor (virtual hardware template; 1 virtual CPU core, 1 GB RAM, 50 GB disk). A Zabbix monitoring system [9] is configured to monitor and collect necessary performance metrics for each compute node and for each VM in the system. Table III provides the list of metrics that are monitored and periodically collected by RAVA from the monitoring system and the cloud controller. The Analytics Engine (AE) and Decision Engine (DE) functional blocks inside the controller implement the RAVA method logic using Python. The AE analyzes the RRAS reports for deriving AS and other necessary statistics, which are then fed to the DE. Based on the output of the AE and in view of the operator's policy, the DE makes relevant LCM decision for the VNF(s). These decisions are then translated into OpenStack commands and hence executed for the specified VNF(s).

B. Results Analysis

In order to demonstrate RAVA capabilities, we created a scenario where a highly loaded I/O intensive VNF (target-VM) in compute node 1 (CN-1) is throttling the network I/O of existing VMs co-located in the same compute node (i.e., CN-1). The target-VM could be live migrated to one of the two available candidate compute nodes (i.e., CN-2 or CN-3) as both have the required I/O resources available to host the target-VM. At the time when the target-VM needs to be migrated, we observe the average network load on CN-2 is less than that on CN-3. This is depicted in Figure 4.

But at the time, the I/O utilization of VMs in CN-2 will have a strong correlation (or affinity) with the CPU utilization as opposed to the VMs in CN-3. Under normal scheduling

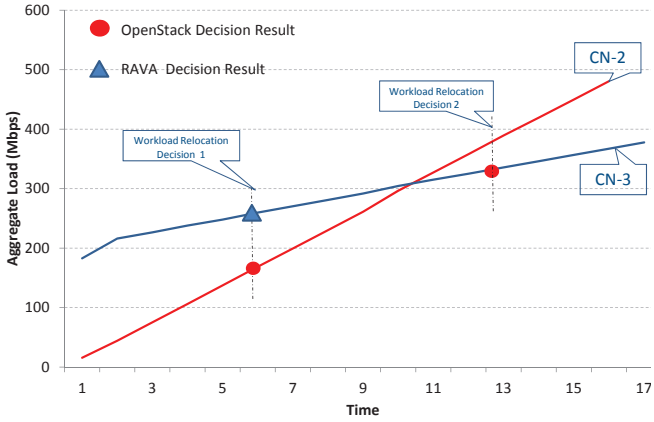
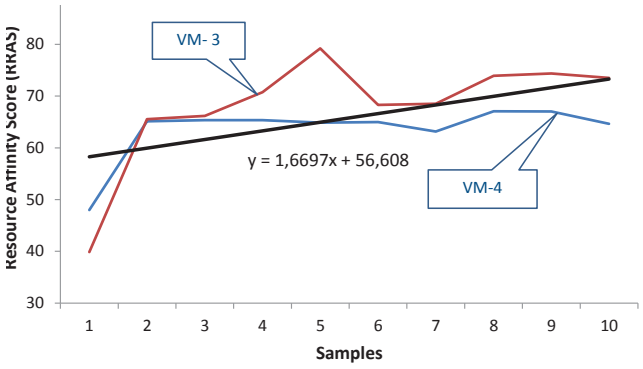


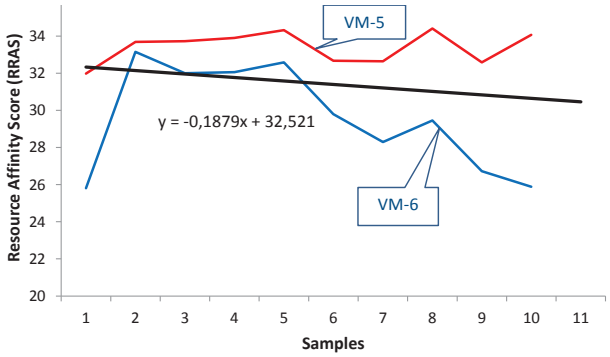
Fig. 4. Load comparison between CN-2 and CN-3

policy rules employed in OpenStack, triggering a migration of target-VM will lead Nova Scheduler (OpenStack Computing scheduler component) to select CN-2 as node to migrate the target-VM to. Such decision is based on the assessment of the resource allocation ratio per compute node, dispatching the migration request to the less allocated compute node. This means the cloud controller overlooks the actual computational load posed by virtual resources, e.g., one or a set of VMs may be exhausting the network I/O capacity while still being within its allocated bandwidth limits. The cloud platform is not able to determine and *predict* the strong correlation of the I/O resource unit with the CPU resource unit for the VMs in CN-2. This will eventually overload the compute and network resources of CN-2. This is depicted in Figure 4 where the load utilization of CN-2 will increase beyond CN-3 soon afterwards. Thus a normal controller logic will need to execute the migration of target-VM once more to CN-3. In other words, the QoD of the controller was not optimum.

However, RAVA method will be able to determine and predict this strong correlation between the CPU utilization and the network I/O resource utilization of the VMs for the entire compute node pool. RAVA will also be able to determine that the VMs in CN-3 do not have such a strong affinity between CPU and I/O resource units. Thus, the controller will choose CN-3 over CN-2 as a host for the target-VM to live migrate to. This is illustrated in Figure 5 which shows the Affinity Signature (AS) for the VMs in CN-2 (Figure 5(a)) and CN-3 (Figure 5(b)) respectively. As described before, the AS is a plot of the successive RRAS values, a RAVA metric that is utilized by the controller for making management decisions. The RAVA method will compute the linear regression expression for the average AS for both CN-2 and CN-3, which determines the degree of affinity between I/O and CPU RUs for the two candidate compute nodes based on the slope and y-intercept values. This is shown as a straight line in Figure 5(a) and Figure 5(b), depicting the straight line equation. As noted, CN-2 has an increasing linear trend with a greater value of slope and y-intercept values when compared with CN-3. This indicates a very strong affinity of the I/O RU with the CPU, as opposed to CN-3 which has a slightly decreasing trend indicted by the negative slope and a smaller y-intercept value,



(a) CN-2



(b) CN-3

Fig. 5. Affinity Signature for the candidate compute nodes.

thereby indicating a weaker affinity between the respective RUs. Thus the controller will select CN-3 as a target-PM where to live migrate the target-VM because CN-3 will have resources available that will serve the resource requirements of target-VM on a long term basis without any adverse impact on the performance of existing VMs in CN-3 (i.e., VM5 and VM6 respectively). This will ensure to keep the loads on CN-2 and CN-3 within preferred limits. Due to the strong correlation of I/O resource unit with CPU resource unit, the utilization of both I/O and CPU resource units of the VMs in CN-2 will continue to increase as opposed to CN-3 where it will remain almost constant after the target-VM has been migrated to CN-3. Thus RAVA method clearly demonstrates enhancing the QoD of a cloud controller, such as NFVO.

V. Conclusion

In this paper, we have presented the first results of a proof-of-concept implementation of the proposed RAVA management and orchestration method that is designed to enhance the QoD of a cloud management and orchestration entity (e.g., NFVO) by optimizing the LCM decisions that it takes on the VNFs at run-time. We introduce the notion of QoD, and for the purpose of evaluation we developed an experimental test-bed in an OpenStack environment that enabled us to evaluate RAVA's performance in the context of QoD. The novel concept behind RAVA is the computation of RRAS vector values on a per VM basis hosted on PMs at run-time. The RRAS values are based on the correlation between the utilization of different RUs with a reference RU. Based on successive RRAS values, an Affinity Signature (AS) is derived that enables the cloud controller to get a fine granular and precise view of the degree of affinity (or correlation) of a RU or a plurality of RUs with a reference RU, thereby enabling it to make more informed and thus optimized management decisions. The scheme has the potential to enable cloud providers to perform precise and fine-granular resource tailoring by efficient allocation of resources among VMs. The scheme is thus used for efficient LCM operations on VMs for instantiating and managing (during run-time) VNFs to ensure a sustained and reliable carrier cloud operations. This work serves mainly as a high-level proof-of-concept on the potential impact of taking informed decisions on this context. The present model suffers some limitations (e.g., when Moving Average components of the signal does not increase/decrease monotonically) which will be tackled in the future work. As future work we propose to explore the time series analysis models [10] to properly assess the trends/seasonalities and the stationarity of the signal of correlated values, which are key to guarantee the robustness of our resource usage prediction schema (e.g., [11] [12]).

Acknowledgment

The research work presented in this paper is conducted as part of the Mobile Cloud Networking project, funded from the European Union Seventh Framework Program under grant agreement no[318109].

References

[1] T. Taleb, "Towards Carrier Cloud: Potential, Challenges, and Solutions", in IEEE Wireless Communications Magazine, Vol. 21, No. 3, pp. 80-91., June 2014.
[2] ETSI GS, "Network Function Virtualization (NFV) Management and Orchestration", NFV-MAN 001 v0.8.1, Nov 2014
[3] OpenStack Open Source Software for Creating Public and Private Clouds, http://www.openstack.org, Oct 2015
[4] F. Z. Yousaf, P. Loureiro, F. Zdarsky, T. Taleb and M. Liebsch, "Cost analysis of initial deployment strategies for virtualized mobile core network functions," in IEEE Communications Magazine, vol. 53, no. 12, pp. 60-66, Dec. 2015.
[5] G. Somani, P. Khandelwal, and K. Phatnani, "VUPIC Virtual Machine Usage Based Placement in IaaS Cloud", CoRR abs/1212.0085 (2012)
[6] G. Smirnov, K. Hu, and D. Kaeli, "Systems and Methods for deter-mining placement of virtual machines", Patent No, US 8,099,487 B1, Jan 17, 2012
[7] S. Sudevalayam and P. Kulkarni, "Affinity-Aware Modeling of CPU Usage for Provisioning Virtualized Applications", in Proc. IEEE Intl Conf. on Cloud Computing, Jul. 2011
[8] F. Z. Yousaf and T. Taleb, "Fine-grained resource-aware virtual network function management for 5G carrier cloud," in IEEE Network, vol. 30, no. 2, pp. 110-115, March-April 2016.
[9] Zabbix - The Enterprise Class Monitoring Platform, http://www.zabbix.com, Oct 2015.
[10] J. Cryer, K. Chan, "Time Series Analysis with Applications in R", New York, NY, USA: Springer-Verlag, 2008
[11] Moreira-Matias, L., Gama, J., Ferreira, M., Mendes-Moreira, J., Damas, L.: "Online Predictive Model for Taxi Services". In: Advances in Intelligent Data Analysis XI, LNCS vol. 7619, pp. 230-240. Springer Berlin / Heidelberg (2012)
[12] L. Moreira-Matias and F. Alesiani, "Drift3Flow: Freeway-Incident Prediction Using Real-Time Learning," 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Las Palmas, 2015, pp. 566-571.