Fine-Grained Resource-Aware Virtual Network Function Management for 5G Carrier Cloud

Faqir Zarrar Yousaf and Tarik Taleb

Abstract

This article proposes an approach that enables a cloud infrastructure MANO entity, in a virtualized infrastructure (e.g., a data center) that hosts virtualized networks and services, to derive the affinity scores for the plurality of resource units with reference to a specific resource unit for each individual virtual machine instance. This affinity score, referred to as the reference resource affinity score, will enable a MANO entity to perform precise and efficient resource tailoring or dimensioning, and hence will optimize its decisions and actions related to the operations and management of the virtualized systems inside the cloud infrastructure. The proposed approach is particularly of vital importance for the management of virtual network functions that are chained to create network services as part of the carrier cloud concept, an important vision of the future 5G architecture.

Fifth generation (5G) mobile networks, also referred to as beyond 2020 mobile communications systems, represent the next major phase of the mobile telecom industry, going beyond the current Long Term Evolution (LTE) and International Mobile Telecommunications (IMT)-Advanced systems. In addition to increased peak bit rates, higher spectrum spectral efficiency, better coverage, and the support of potential numbers of diverse connectable devices, 5G systems are required to be cost efficient, flexibly deployable, elastic, and above all programmable. This has become critical for the 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.

Along with recent and ongoing advances in cloud computing and their support of virtualized services, it has become promising to design flexible, scalable, and elastic 5G systems benefiting from advanced virtualization techniques of cloud computing and exploiting recent advances relevant to network function virtualization (NFV). The inherent potential and recent advances in the area of NFV have made it recognized as the key enabler for the realization of a carrier cloud, a key component of the 5G system. There are proposals [1] and commercial solutions [2] available that realize the Evolved Packet Core (EPC) on a virtualized platform. However, there are numerous challenges for making NFV carrier-grade [3], and a European Telecommunications Standards Institute Industry Special Group (ETSI ISG) has been formed to standardize various aspects of an NFV-enabled network, including the NFV management and orchestration (NFV-MANO) framework [4].

According to [3], network functions (NFs) are realized on virtual machines (VMs) as virtualized NFs (VNFs), which are deployed and 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, and each PM can host tens to hundreds of VMs. A VM, by itself, is an abstraction of a PM that is assigned a 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 VM monitor (VMM) that manages the multiple VMs on it and monitors their respective resource consumption.

In an NFVI, multiple VNFs are chained to realize different network services (NSs), and the entire system is managed and orchestrated by the NFV-MANO framework [4] via its three main functional blocks: 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 (VNFIs), such as VNF instantiation, migration, and scaling. The NFVO, on the other hand, is not only responsible for the LCM of the NS but also orchestrates the NFVI resources across multiple VIMs.

Problem Statement

When deploying a VNF, a cloud service provider (CSP) (owner of the NFVI) typically offers a set (or menu) of "flavors" to the VNF provider (VNFP). The term flavor, in OpenStack terminology [5], refers to the available hardware configuration block for a VM. Each flavor has a unique combination of disk space, memory capacity, and priority of CPU time (i.e., processing requirement). The 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

Faqir Zarrar Yousaf is with NEC Laboratories Europe.

Tarik Taleb is with Sejong University and Aalto University.

0890-8044/16/$25.00 © 2016 IEEE

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:

- 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 underutilization of the underlying resources that are "pinned" to the particular flavor. The unutilized portion of the assigned resources is referred to as a "resource black hole" throughout this article.
- 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 [6, 7].
- Such a process does not take into account unforeseen traffic reductions that may result in underutilization of the assigned resources; that is, accumulating resource black holes and also contributing to higher energy consumption.
- Considering the strict resource allocation and isolation policy, it also prevents the sharing and runtime/dynamic (re)allocation and/or (re)organization of the unutilized capacities of the underlying physical resources that are "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 article is thus to propose a remedy to the above limitations by designing a method/system that will:

- Optimize the LCM operations in a large-scale DC
- Enable optimal utilization of the assigned and available resources
- Enable dynamic and precise (re)assignment, (re)distribution, (re)allocation, (re)organization, and sharing of resources among multiple VM instances on the same PM at runtime and at a fine-grained level
- Minimize the occurrences of the above-mentioned costly VM management operations

The overall purpose of the article is to introduce a fine-grained scheme that the resource management entity (e.g., VMM and/or VIM) can utilize for making informed and optimum management decisions in view of changing workload conditions. The utility of the scheme will also be described with respect to the management of VNFs as part of the carrier cloud concept, one of the key visions of the future 5G architecture as explained earlier.

The remainder of this article is organized as follows. The following section presents the state of the art. Then we describe our proposed fine-grained resource-aware VM management scheme using a number of exemplary implementations, providing its qualitative evaluation, and showcasing its technical benefits. Following that, we discuss how the proposed scheme can be used for the management of VMs for running VNFs composing NSs indicating the advantages, limitations, and challenges. The conclusion and some future work are presented later.

State of the Art

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 an NS. This is because an NS is formed by chaining multiple VNFs, where a VNF may be decomposed into multiple VNF components (VNFCs). 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 [8].

There is a large library of research work that has been conducted for decisions on VM placement, resource allocation, and VM management with the objective of cost savings from better utilization of computing resources and less frequent overload situations. For instance, in [9], performance isolation (e.g., CPU, memory, storage, and network bandwidth), resource contention properties (among VMs on the same physical host), and VMs' behavioral usage patterns are taken into account in decisions on VM placement, VM migration, and cloud resource allocations. In [10], VNFs are placed based on load requirements for each VNF type, and appropriate PMs are selected based on the available resources and inter-VNF constraints. In other research works, optimal placement of VMs on PMs, running specific services, consider electricity-related costs as well as transient cooling effects [11]. Others do autonomic placement of VMs as per policies specified by the DC providers and/or users [12]. Other VM placement strategies consider maximizing the profit under a particular service level agreement (SLA) and a predetermined power budget [13].

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 [14], 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, that is, the utilization value of some specific resource unit or a set of them 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 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 is within the scope of this article. Furthermore, the scope of [14] 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 toward making any virtual infrastructure management decision such as cloning and scaling.

Another related work is reported in [15], 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 "prediction" of the estimated CPU utilization, and for that purpose, the work proposed in [15] performs benchmarking of the CPU utilization with respect to different workloads through repeated experiments. Again, the scope of [21] is limited to making placement decisions of only intercommunicating VMs, based, in turn, on only predicting the estimated utilization of the CPU.

Thus, regardless of the objective, any placement algorithm

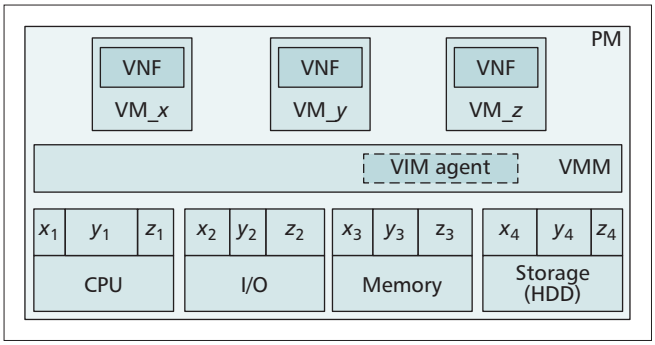

Figure 1. High-level view of a PM hosting multiple VMs with specific resource allocation.

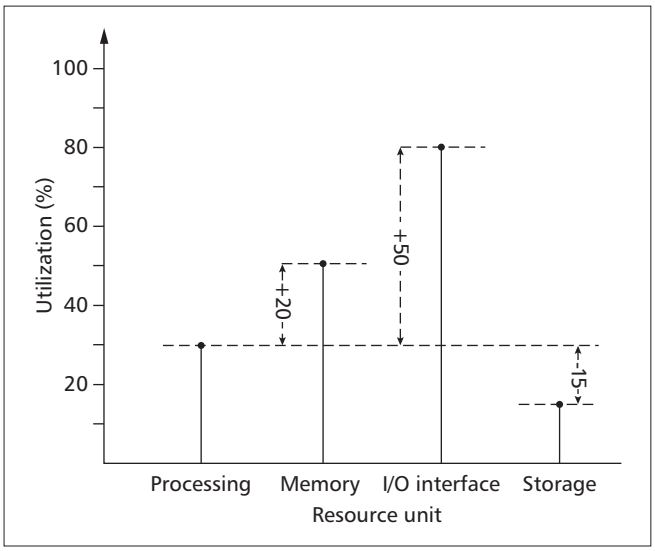

Figure 2. Resource utilization for a specific VM with RRAS with respect to CPU.

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. In contrast, the scope of the scheme proposed herein defines a novel method of rationalizing resource utilization in a specific way, and not in isolation but rather with reference to other RUs.

Proposed Fine-Grained Resource-Aware VM Management

Method Overview

The proposed scheme is applicable to all PM platforms that host VMs. Such PMs are controlled by a VMM that creates, runs, and manages all VMs running in PMs. The VMM ensures fair allocation, utilization, and isolation of the underlying physical resources between multiple VMs as per the allocation scheme specified per VM. This implies that the VMM has the ability to monitor and schedule the utilization of the underlying physical resources among the multiple VM instances hosted on a PM.

The concept of "affinity" is central to the proposed scheme, whereby the term affinity refers to the correlation between different entities, which in our case will be 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. As detailed below, the proposed scheme derives and communicates information depicting the correlation, or affinity, between different RUs with reference to a specified key RU under different workload conditions. This derived information is referred to as the reference resource affinity score (RRAS).

The RRAS provides insight as to how and by how much the utilization of one reference RU will impact the other RUs. The RRAS is thus a vector quantity that expresses the correlation or level of dependence of an individual RU on the reference RU in terms of utilization. The RRAS value for a particular RU indicates to what degree its utilization depends on the utilization of the reference RU. In other words, it refers to what level the utilization of a reference RU will incur the utilization in other RU(s), or how dependent an RU is with respect to the reference RU in terms of utilization. For example, the RRAS value of an I/O module 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 computation/derivation of the RRAS value can be done by a VMM or by a special software agent of the VIM, referred to as a VIM agent, present within or co-located with the VMM. The RRAS values are computed for all VMs in a PM and are presented as RRAS reports. A RRAS report presents fine-grained information that can be incorporated by an internal resource management entity (i.e., VMM) or by an external resource management entity, such as a VIM/VNFM/NFVO, to make informed decisions in terms of optimum resource management under different workload conditions at the server level or infrastructure level, respectively.

The VIM agent computes and maintains RRAS for all VMs on the underlying PM, and the results, which are presented as RRAS reports, can be used locally by the VMM to administer appropriate management actions within the PM such as runtime resource sharing, or send the RRAS reports to VIM to make appropriate management decision(s). The VIM agent can send the RRAS reports to the VIM either periodically, on demand, or whenever a specified utilization threshold is exceeded.

To explain the proposed mechanism more clearly, consider Fig. 1, which illustrates a high-level view of a PM with VMM and hosting multiple VMs, where each VM is hosting a VNF. Each VM is allocated a specific slice of the underlying physical RUs. As an example, we consider the following RUs:

- Processing (e.g., CPU cores)
- Memory (e.g., RAM)
- I/O module (e.g., Gigabit Ethernet network interface card, NIC)
- Storage (e.g., local storage such as HDD).

A VMM not only manages VMs but also ensures that each VM gets its share of the stipulated resource. For the sake of simplicity and explanation, we consider three VMs (i.e., x, y, and z), and the size of the RU (1, 2, 3, or 4) allocated to each VM depends on the hosted VNF requirement. Each VM utilizes its allotted share of RUs in proportion to the incident workload, whereas the VMM ensures isolation between the allocated resources for the different VMs. As an example, Fig. 2 shows the average percent utilization of each respective RU recorded over a specific time period $T = t_n - t_0$, where t_0 and t_n denote the start and end times of a monitoring epoch for a VM, respectively. These values can be derived by the VIM agent for each VM using the resource monitoring function of the VMM. The accuracy of these values depends on the number of samples collected during the monitoring epoch and also on the sampling rate of the VMM's monitoring function.

Based on these individual scalar values, the VIM agent derives the RRAS report for each VM, an exemplary sample of which is shown in Table 1. Table 1 provides the RRAS

report corresponding to the example utilization values of a specific VM, shown in Fig. 2. Based on the absolute average utilization values, the VIM agent computes the RRAS for each RU with reference to each other RU. Any suitable approach can be employed to compute the RRAS values. In this article, as an example, we take the difference between the average utilization values of the RUs with the reference RU, and consider this difference as the RRAS value (Table 1). In Table 1, the RU of each row is a reference with respect to which the RRAS for other RUs are computed.

Reference resource unit	Absolute average utilization (%)	Reference resource affinity score			
		Processing	Memory	I/O module	Storage
Processing	30	—	+20	+50	–15
Memory	50	–20	—	+30	–35
I/O					
Module	80	–50	–30	—	–65
Storage	15	+15	+35	+65	—

Table 1. RRAS report for a specific VM on a PM.

Being vector quantities, the RRAS values show how much the utilization of the reference RU is impacting the other RUs. A positive RRAS value of a particular RU indicates that the utilization of the reference RU will result in increased utilization of the specific RU well above the level of utilization of the reference RU. On the other hand, a negative value indicates that the utilization of the reference RU will result in decreased utilization of the particular RU well below the level of utilization of the reference RU. This will help to enable the VIM to precisely determine the influence of a reference RU on the other RUs. For instance, with reference to a processing RU (e.g., CPU), it is observed from Fig. 2 that 30 percent utilization of CPU will correspond to 50 percent of the utilization of the memory RU, 80 percent utilization of the I/O module RU, and 15 percent utilization of the storage RU. Based on these individual utilization values, the RRAS values can be derived. One simple method for deriving RRAS values could be to determine the absolute difference between the percentage utilization of the respective RUs and the reference RU, as considered in Table 1. For example, the first row in Table 1 indicates the RRAS values computed with reference to the processing RU (i.e., the CPU). In this case, the memory unit will have an affinity score of +20, I/O module +50, and storage –15. Similarly, the second row in Table 1 shows the affinity of the individual RUs with respect to the memory.

The notion of deriving an affinity score indicates how strongly an individual RU correlates with the reference RU under specific workloads. Thus, from the table, it can be interpreted that the I/O module has the "strongest" affinity with the CPU with an RRAS value of +50, while the storage has the least affinity, which is –15. 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.

The VIM agent periodically generates RRAS reports by observing the RU values over a specific period of time (T). The VIM agent may be instructed by the VIM to derive an RRAS report with reference to a specific RU, a subset of RUs, or all RUs. The VIM agent (or preferably the VIM) can also store and maintain 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 records of the RRAS report enables the VIM or VIM agent to derive the affinity signature (AS) of RU(s) with respect to a reference RU for the VMs. An AS provides the VIM or VIM agent information about the long-term affinity of an RU with a reference RU for a VM. Time series models can be used to improve the accuracy of an AS. An AS is a plot of successive RRAS values against the specific utilization values of a reference RU. If there are multiple RRAS values for a certain utilization level (or utilization range) of the reference RU, the average RRAS value is then considered for the AS.

The information provided by the RRAS reports and AS can be manipulated by deriving statistics such as affinity trend as

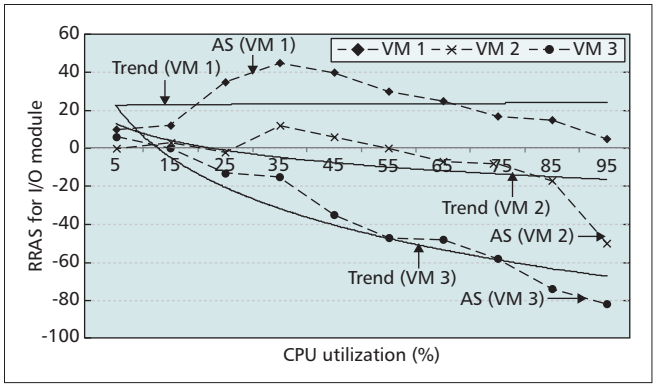

Figure 3. Affinity signature for I/O module RU with reference to CPU RU based on RRAS reports.

seen in Fig. 3. The figure shows the AS of the I/O modules for three VMs with reference to the processing RU (i.e., CPU). The dotted lines indicate the AS of the respective VMs, while the solid lines indicate the logarithmic trend of the affinity of I/O with CPU. As evident from the figure, the I/O module of VM 1 shows a consistently strong long-term affinity with CPU. This is also indicated by the almost constant trend on the positive axis. This shows that the utilization of the I/O module is highly dependent on the CPU utilization as high CPU utilization will incur high utilization of the I/O module. On the other hand, the long-term affinity of VM 2 and VM 3 is not very strong, as indicated by the decaying trend of the AS in the negative axis. This shows that the utilization of the I/O resources is not significantly impacted with CPU utilization. This could imply non-I/O-intensive VNFs hosted on VM 2 and VM 3, while VM 1 hosts an I/O-intensive VNF.

The AS of different RUs can be combined to determine the short-term and long-term demands and behavior of a VNF. The information thus derived from the AS can be utilized by the NFV-MANO functions (e.g., VNFM and NFVO) for making short-term and long-term informed and optimized decisions in performing LCM operations such as, but not limited to:

- VM deployment and instantiation
- VM migration
- VM cloning
- VM scaling, horizontal and/or vertical
- Runtime dynamic resource provisioning

To further refine the RRAS reports and the AS graph, the time of day (ToD) can be also considered to indicate the times at which the affinity is strongest or weakest. This can help the system in optimum management of resources at different ToDs. As a use case, consider a routing VNF the AS of which indicates a strong affinity of the I/O RU and weak affinity of the memory RU during specific ToDs when the routing VNF is involved in routing user plane (U-plane) data traffic.

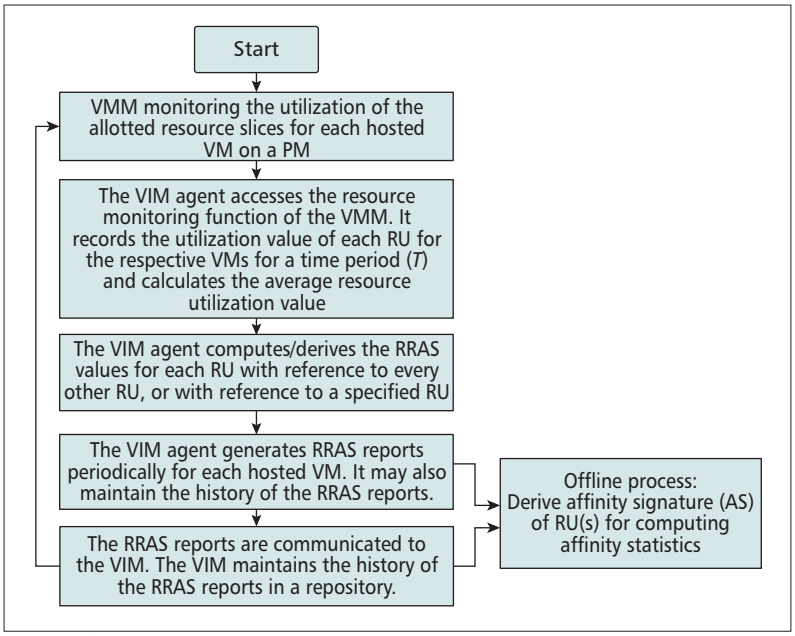

Figure 4. Overview of the functional steps of the proposed scheme.

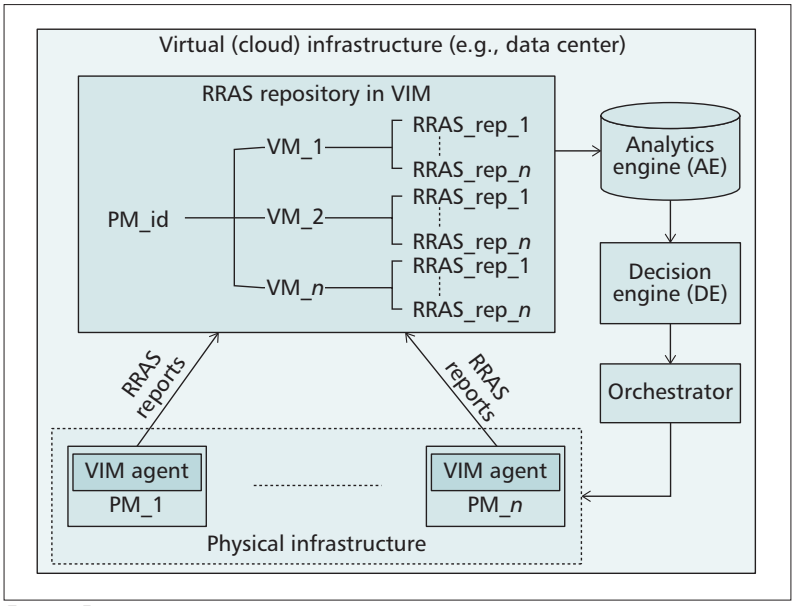

Figure 5. Conceptual overview of VIM containing the RRAS repository.

Considering the fact that for machine-to-machine (M2M) applications the control plane (C-plane) traffic exceeds the U-plane traffic, thus imparting a higher demand on memory RU (for maintaining states) than on I/O RU, the system may then decide to reroute the M2M traffic over this routing VNF to maximize the utilization of available resources. Additionally, the AS information can also be used for power savings, that is, VNFs indicating weaker affinity to allotted resources can be consolidated on fewer servers, thereby preserving resources and energy.

As another use case, the RRAS reports and AS graph can be used by a VMM in making local management decisions, such as the (re)allocation of underutilized resources (i.e., resource black holes) from one VM to another within the same PM. For example, with reference to the AS graph shown in Fig. 3, since the I/O module slice allocated to VM 3 shows a lower affinity with the reference CPU slice, the VMM can then re-allocate/redistribute a portion of the I/O module resource slice fromVM 3 to the I/O-intensive VM 1 in order to enhance its I/O performance.

Such local resource management actions can be a temporary measure until the NFV-MANO system is able to make long-term resource arrangements for the needy VM (i.e., VM 1 in our case) that is running low on allocated resources. One possible management action is for the VNFM to instantiate VM 1 on a different PM that can guarantee resources on a long-term basis. Once instantiated, the resources that were taken away from VM 3 can be released back to it. This has the advantage of keeping VM management operation transparent from VM users while minimizing service disruptions. The re-allocation of portions of the RU with lower affinity can also be done on a long-term basis, depending on the underlying policy. The AS of RUs can also be used in performing cloud infrastructure resource analytics. It can be utilized by a suitable machine learning algorithm in formulating effective future actions. The proposed scheme can accordingly be used to enforce efficient task scheduling as well as load balancing.

System Overview

Figure 4 provides an overview of the functional steps of the proposed scheme, while Fig. 5 shows the conceptual overview of the RRAS repository in VIM. As illustrated in Fig. 5, the PMs send the RRAS reports, which are stored in the RRAS repository in the VIM, via the VIM agent toward the VIM. The RRAS repository is a data structure that maintains the history of the RRAS reports for each PM within the NFVI. The reports (characterized by RRAS-rep-ID, where the ID can be an integer value or ToD) within the repository can be referenced based on a unique VM-id, which in turn can be referenced based on the PM_id. The RRAS reports are analyzed by the analytics engine (AE) for deriving AS and other necessary statistics, which are then fed to the decision engine (DE). Based on the output of the AE, the DE makes a specific LCM decision, which is communicated to the orchestrator for carrying out necessary actions to enforce the decisions. For example, the AE, based on the RRAS reports, can derive an AS similar to the one shown in Fig. 3. The AE then pushes the derived AS to the DE, which can compute necessary statistics on the AS. Based on such statistics, the DE will determine a specific action or set of actions. The action space computed by the DE may include LCM operations such as, but not limited to, VM migration, cloning, vertical/horizontal scaling (scale-in or scale-out), and resource re-allocation/redistribution. When determining the appropriate action or set of them, the DE may also take into consideration the operator's policy depending on the type of AF and the SLA. With specific reference to Fig. 3, considering the strong long-term affinity of the I/O RU with the CPU for VM 1, the DE may decide to migrate and deploy VM 1 on a different PM that can offer higher I/O capacity than the existing PM and can also take care of unexpected throughput surges. The DE will convey its preferred decision to the orchestrator, which is then responsible for enforcing it by managing the process of locating the appropriate PM and then migrating VM 1 to its new location with minimum service interruption. To ensure minimum service disruption, the

orchestrator may vertically scale up the I/O resource of VM 1 by temporarily re-allocating the unutilized portion of the I/O RUs assigned to VM 2 and VM 3 to VM 1 until the VM 1 migration process is completed. All the supplementary tasks that may arise due to VM 1 migration, such as state/context transfer, and adjusting routing rules within DC to ensure that flows get diverted to the new location of VM 1, are also handled by the orchestrator. With reference to the NFV-MANO system, the AE/DE can be part of the NFVO/VNFM.

Scheme Utility for Virtualized Mobile Network Management

The method/system described can be used toward the optimized management of virtualized mobile networks, such as the virtualized EPC (vEPC) system [2], which offers a complex ecosystem of multiple VNFs/VNFCs interconnected to deliver a variety of NSs. The vEPC system consists of several virtualized instances of mobility management entity (vMME), serving gateway (vSGW), and packet data network gateway (vPGW). Furthermore, each VNF may be further decomposed into a number of VNFCs. For example, a vMME may consist of a service load balancer (SLB) VNFC and a mobility management processor (MMP) VNFC. Similarly, a vS/PGW may be decomposed into a vS/PGW-C VNFC and a vS/PGW-U VNFC separately handling control plane (C-plane) and user plane (U-plane) traffic, respectively [2].

In view of the varying traffic load and changing service requirements, the method proposed herein will enable the VIM to have a fine-grained view of how much each VNF/VNFC is consuming their respective resources, and at the same time the system will be able to assess the impact of a VNF/VNFC on others. For instance, there could be a case where a vS/PGW-C and vS/PGW-U are collocated on the same PM, and it becomes noticeable that the memory and CPU utilization have increased with reference to I/O RU, indicating MTC traffic. In such a situation the VIM, in view of RRAS, can take any one of multiple available options. It can instantiate a new vS/PGW-C VNFC to handle the excess C-plane traffic. Otherwise, it may migrate the vS/PGW-U VNFC to another PM having the requisite resources (i.e., memory and CPU). Additionally, in view of AS, the VIM can make preemptive management decisions in order to ensure uninterrupted service.

Conclusion

In this article, we propose a scheme that renders greater precision to the ability of a local virtual infrastructure management entity (e.g., VIM agent) or a global virtual infrastructure management entity (e.g., VIM) in exercising virtual infrastructure management decisions by computing RRAS vector values for every VM hosted on a PM. The RRAS values are based on the resource utilization levels monitored by the VMM over a specific time period. RRAS values enable the cloud infrastructure management entity (e.g., NFV-MANO system) to get a fine-grained and precise view of the degree of affinity (or correlation) of an RU or a plurality of RUs with a reference RU. The scheme enables cloud providers to perform precise and fine-grained resource tailoring, ensures against the rigid pinning of resources to specific VM instances, and minimizes resource black holes through efficient re-allocation of local resources among VMs. The scheme can be used for efficient LCM operations on VMs for instantiating and managing (during runtime) VNFs and creating NSs to enable the carrier cloud concept, one of the key visions of the future 5G architecture.

At present we are planning to implement the proposed method as a module in our OpenStack testbed and develop test cases for the various use cases described in this article, and analyze its performance under realistic traffic and system conditions.

Acknowledgment

Part of the research work presented in this article is conducted as part of the Mobile Cloud Networking project, funded by the European Union Seventh Framework Program under grant agreement N[318109].

References

[1] T. Taleb *et al.*, "EASE: EPC as a Service to Ease Mobile Core Network," to appear, *IEEE Network*.
[2] White Paper TE-524262, "NEC Virtualized Evolved Packet Core vEPC," Oct. 09, 2014; http://www.nec.com/en/global/solutions/nsp/nfv/doc/vEPC WP.pdf.
[3] T. Taleb, "Toward Carrier Cloud: Potential, Challenges, & Solutions," *IEEE Wireless Commun.*, vol. 21, no. 3, June, 2014. pp. 80–91.
[4] ETSI GS, "Network Function Virtualization (NFV) Management and Orchestration, NFV-MAN 001 v. 0.8.1, Nov 2014.
[5] OpenStack — Open Source Software for Creating Public and Private Clouds, http://www.openstack.org, Mar., 2015.
[6] A. Ksentini, T. Taleb, and F. Messaoudi, "A LISP-Based Implementation of Follow Me Cloud," *IEEE Access*, vol. 2, Oct. 2014. pp. 1340–47.
[7] T. Taleb and A. Ksentini, "Follow Me Cloud: Interworking Federated Clouds and Distributed Mobile Networks," *IEEE Network*, vol. 27, no. 5, Sept./Oct. 2013. pp. 12–19
[8] F. Z. Yousaf *et al.*, "Cost Analysis of Initial Deployment Strategies for a Virtualized Mobile Core Network Functions," to appear, *IEEE Commun. Mag.*
[9] G. Somani, P. Khandelwal, and K. Phatnani, "VUPIC Virtual Machine Usage Based Placement in IaaS Cloud," CoRR abs/1212.0085 (2012).
[10] F. Z. Yousef *et al.*, "SoftEPC Dynamic Instantiation of Mobile Core Network Entities for Efficient Resource Utilization," *Proc. IEEE ICC 2013*, Budapest, Hungary, June 2013
[11] K. Le *et al.*, "Reducing Electricity Cost through Virtual Machine Placement in High Performance Computing Clouds," *Proc. Int'l Conf. High Performance Computing, Networking, Storage and Analysis*, Seatle, WA, Nov. 2011.
[12] C. Hyser *et al.*, "Autonomic Virtual Machine Placement in the Data Center," HP Labs, HPL-2007-189, Feb. 2008.
[13] W. Shi and B. Hong, "Towards Profitable Virtual Machine Placement in the Data Center," *Proc. 4th IEEE Int'l Conf. Utility and Cloud Computing*, 2011.
[14] G. Smirnov, K. Hu, and D. Kaeli, "Systems and Methods for Determining Placement of Virtual Machines," U.S. Patent No. 8,099,487 B1, Jan. 17, 2012.
[15] S. Sudevalayam and P. Kulkarni, "Affinity-Aware Modeling of CPU Usage for Provisioning Virtualized Applications," *Proc. IEEE Int'l Conf. Cloud Computing*, July 2011.

Biographies

FAQIR ZARRAR YOUSAF [M] (zarrar.Yousaf@neclab.eu) is working as a research scientist at NEC Laboratories Europe Ltd, Heidelberg, Germany. He received his Ph.D. from Dortmund University of Technology, Germany, in April 2010. His current research interest is SDN/NFV, and he is actively involved in the ETSI NFV standardization activities. He has served as a TPC member for various conferences and journals.

TARIK TALEB [M] (talebtarik@ieee.org) is currently a professor at the School of Electrical Engineering, Aalto University, Finland. He has worked as a senior researcher and 3GPP standards expert at NEC Europe Ltd. Prior to his work at NEC, until March 2009, he worked as an assistant professor at the Graduate School of Information Sciences, Tohoku University, Japan, in a lab fully funded by KDDI. He received his B.E. degree in information engineering with distinction, and his M.Sc. and Ph.D. degrees in information sciences from Tohoku University in 2001, 2003, and 2005, respectively. His research interests lie in the field of architectural enhancements to mobile core networks (particularly 3GPP's), mobile cloud networking, mobile multimedia streaming, and social media networking. He has also been directly engaged in the development and standardization of the Evolved Packet System. He is a member of the IEEE Communications Society Standardization Program Development Board and serves as Steering Committee Chair of the IEEE Conference on Standards for Communications and Networking.