2014 IEEE 22nd International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems

Multi-Criteria Virtual Machines Migration Considering the Reconfiguration of their Logical Topology

P. Kokkinos, T.A. Varvarigou
Department of Electrical and Computer Engineering
National Technical University of Athens
Athens, Greece

A. Kretsis, E.A. Varvarigos
Department of Computer Engineering and Informatics
University of Patras
Computer Technology Institute and Press "Diophantus"
Patras, Greece

***Abstract*—We present a methodology, called communication-aware virtual infrastructures (COMAVI), for the concurrent migration of multiple Virtual Machines (VMs) in cloud computing infrastructures, which aims at the optimum use of the available computational and network resources, by capturing the interdependencies between the communicating VMs. This methodology uses multiple criteria for selecting the VMs that will migrate, with different weights assigned to each of them. COMAVI also selects the computing sites/units where the migrating VMs will be hosted, by accounting for the way migration affects the logical (or virtual) topologies formed by the communicating VMs and viewing this selection as a logical topology reconfiguration problem. COMAVI resolves the maximum possible number of VM resource shortages, while tending to minimize the number of migrations performed, the induced network overhead, the logical topology reconfigurations required, and the corresponding service interruptions. We evaluate the proposed method through simulations, where we exhibit their performance benefits.**

Keywords— virtual machine migration; multi-criteria; virtual topologies; communication-aware

I. INTRODUCTION

One of the means used to improve or balance resource utilization in cloud computing infrastructures (data centers) is the migration of Virtual Machines (VMs). Migration is a resource management operation performed after the initial scheduling phase, in order to cope with the dynamicity of the computing environment.

VMs can migrate between machines located on the same local area network (LAN), or on different ones, over a metropolitan area network (MAN) or a wide area network (WAN). The LAN environment corresponds to a data center, while the MAN/WAN environment to that of a set of interconnected private and public clouds (hosted in data centers) around the world. In a data center scenario, multiple physical machines host several thousand interconnected VMs, with VM initiation, migration and termination being quite dynamic. Generally, dynamic and centrally controlled VMs migrations are easier applicable and more realistic for the case of a data center in a LAN environment, than for the case of a set of interconnected ones over a MAN/WAN.

VM migration consists of a number of smaller subproblems that address the following questions: (i) when VM migrations will be initiated, (ii) which VMs will migrate, (iii) which physical machine each VM will migrate to, and (iv) how migration will be performed so as to minimize service interruption. Additional and equally important subproblems also exist, such as that of reserving the necessary network resources and deciding the route of the migrating VMs and the new routes of the communicating VMs. Usually, related works consider either subproblems (i), (ii), (iii) jointly, or (iv) as a standalone problem, while very few (if any) works consider all the above subproblems together. Also, a number of parameters (topology, network technologies, energy consumption) affect differently these sub-problems. For example, a fast network path between the source and the destination physical machines is required in order to minimize the service interruption perceived by the user or the applications.

In our work, we present a methodology, called communication-aware virtual infrastructures (COMAVI), for the concurrent migration of multiple Virtual Machines (VMs) in data centers, addressing sub-problems (ii) and (iii) mentioned above. COMAVI is based on the multi-criteria approach [1][2] for the selection of the VMs that will migrate. The multi-criteria approach finds a set of solutions (groups of VM candidates for migration) that are Pareto optimal with respect to the criteria considered. Then various optimization policies can be applied so as to rank these solutions and select one of them. Additionally, the COMAVI methodology selects the computing sites (e.g., servers, racks, data centers) where the migrating VMs will be hosted, taking into account the way migration affects the logical (or virtual) topologies formed by the communicating VMs, in a way similar to the virtual topology reconfiguration (VTR) problem that appears in optical networks [3]. Based on the COMAVI methodology, we present a number of algorithms that favor the concurrent move of more than one VMs. We evaluate the proposed algorithms through simulations, and demonstrate their performance benefits. In particular, we show that the proposed algorithms are able to resolve more resource shortages, with fewer VM migrations and network connection interruptions.

The rest of the paper is organized as follows. In Section II we report on previous work. In Section III we formulate the communication-aware VM migration problem and introduce the notation to be used. In Section IV we present the COMAVI methodology for the optimized migration of VMs. The performance simulation results are presented in Section V. Finally, Section VI concludes the paper.

II. PREVIOUS WORK

A number of works present schemes that make possible the smooth and transparent migration of VMs across servers in a LAN or in a WAN environment [4][5][6][7][8][9]. In general, Virtual Machine (VM) migration over a WAN is challenging, due to the large network delays and service interruption, along with the need for centralized control

1526-7539/14 $31.00 © 2014 IEEE
DOI 10.1109/MASCOTS.2014.36

and information entities. In [7] the authors present an optimized live migration of VM across WANs by combining replication and scheduling strategies. Of course, VM migration is also supported by actual commercial products, such as VMWare [19].

The above techniques provide solutions for making VM migration possible [subproblem (iv) in Section 1], but usually they do not describe/consider algorithms for making the related decisions [subproblems (ii), (iii)]. The authors in [10] propose a migration strategy based on time-windows in order to overcome the limitations posed by greedy event-based VM migration strategies, where the migration is triggered by a QoS non-conformance event. In [11] the authors define and solve the Traffic-aware VM Placement Problem (TVMPP) problem whose input is the traffic matrix among the VMs and a communication cost matrix among host machines (defined differently for different networking paradigms and architectures) and whose objective is to minimize the aggregate traffic rates handled by every switch, along with the corresponding communication cost. In [12] the authors propose a computationally efficient scheme for incorporating (1) inter-VM dependencies and (2) the underlying network topology into VM migration decisions. Their goal is to minimize the data center network traffic while satisfying all server-side constraints. In [13] a Linear Programming formulation and heuristics are proposed to control VM migration, which prioritize VMs with steady capacity

The notion of logical topologies used in our work is mainly encountered in optical WDM networks as a way to record the establishment or deletion of all-optical paths (lightpaths) between source and destination nodes, under dynamic traffic conditions. The logical topology of a WDM network is the set of all-optical connections or lightpaths established in the network. This is an issue studied intensively in several works [14][15][16][17] in the optical networks domain, including methods for the reconfiguration of virtual topologies based on various optimization techniques. Most works try to minimize the reconfiguration cost (e.g., packet losses, number of additional network resource utilized) or maximize the reconfiguration benefits (e.g., alleviate congestion) or both. In our work, we use the logical topology concept, triggered by the apparent trend of utilizing optical networks not only in core and metro networks, but also in access and local networks, such as those used in data centers, in support of the corresponding computing infrastructures.

III. PROBLEM DEFINITION AND MODEL

A. Problem definition

We consider a computing infrastructure operating over a local area network (LAN), such as that inside data centers or over a metropolitan area network (MAN) / wide area network (WAN), such as that of multiple data centers around a city or around the world. We assume that a centralized authority is responsible for collecting node utilization information or traffic demands at regular intervals, executing the COMAVI algorithms, and performing the corresponding VM migrations.

B. Notation and parameters used

We consider a node i having maximum processing power Pi (measured in processing cores). In the case of a LAN environment it can be a collection of servers, a rack or a pod. On the other hand in a WAN environment, with multiple interconnected data centers, a node can actually be a data center with aggregated computational capacity.

Each VM requesting more capacity than the one offered, causes a “violation” that needs to be resolved by migrating this VM to a new node. If a VM requesting more computational capacity than the one offered by a server can migrate to another server in the same node, then we assume that such a case is not a VM violation, since the migration can be performed with minimal cost (no matter how it is defined) when compared to the case where VM needs to migrate to a different node.

Each Virtual Machine VM_n is characterized by its requested computational/processing rate P_{VM_n}. The set of VMs with which VM_n is interconnected (in order to exchange intermediate results) that are hosted on the same node is denoted by $I_{in}(n) = \{VM_{in_1}, VM_{in_2}, \cdots\}$, while the set of communicating VMs that are hosted on different nodes is denoted by $I_{out}(n) = \{VM_{out_1}, VM_{out_2}, \cdots\}$; the cardinalities of these sets are denoted by $|I_{in}(n)|$ and $|I_{out}(n)|$, respectively. The traffic rates of the data exchanged between VM_n and its communicating VMs is denoted by $B(n) = \{B_1, B_2, \dots, B_{|I(n)|}\}$. We let S_n be the size of data transferred upon the migration of VM_n. In addition, the traffic rates B_i of the i-th connection can also be considered separately for internal (B_i,*in*) and external (B_i,*out*) VM flows. Based on the above, each VM_n is assigned a cost/information vector V_n:

$$V_n = \left\{P_{VM_n}, \textstyle\sum_{VM_j \in I(n)} B_j, S_n, |I_{in}(n)|, |I_{out}(n)| \right\}, \tag{1}$$

that records the VM’s information regarding its requested computational capacity P_{VM_n} and its total requested bandwidth $\sum_{VM_j \in I(n)} B_j$. Other parameters, such as the migration cost e.g. in terms of service interruption time, flow priorities and dependencies [18] could also be easily considered, using the presented multi-criteria approach.

IV. COMAVI METHODOLOGY AND ALGORITHMS

In what follows we present COMAVI methodology (Fig. 1) and the corresponding algorithms. Different COMAVI algorithms can also be derived by changing appropriately the optimization criteria and related cost functions.

Within the proposed general methodology, several different choices can be made. In particular, different criteria can be considered by the multi-criteria algorithm (Step-1) and different optimization policies can be applied (Step-2) for identifying the VMs that are selected for migration in order to alleviate the capacity violations. Also, different criteria can be used for the selection of the nodes the VMs will migrate to (Step- 3) and finally different choices of the corresponding virtual topologies can be made.

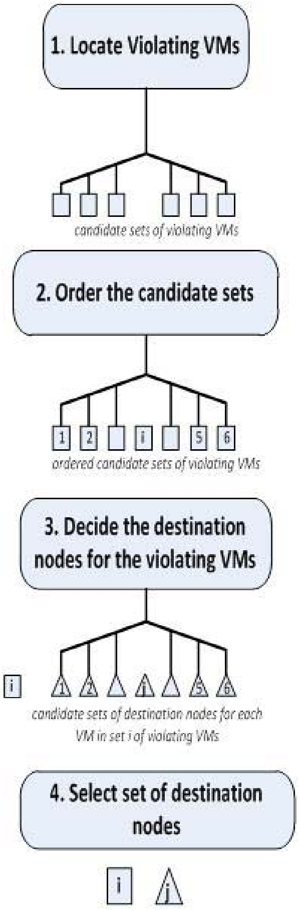


Fig. 1. The COMAVI methodology

COMAVI is based on the following procedure (Fig. 3), which will be detailed in the subsections that follow:

Step-1. Locate sets of VMs, whose migration will alleviate the capacity violations present at each node. Then, using the multi-criteria approach, reduce the number of these sets by selecting the Pareto optimal (non-dominated) sets.

Step-2. Order the candidate Pareto optimal sets using an optimization policy.

Step-3. For the first candidate set decide where its constituent VMs will migrate to. Several different sets of destination nodes can be considered. If it is not possible to find nodes where the VMs can migrate, then select the second best candidate set of violating VMs and repeat this step.

Step-4. For each set of destination nodes, create the corresponding virtual topologies and select the one that optimizes a certain objective function (e.g., minimizes the number of reconfigurations).

A. Locate violating Virtual Machines

The first step is to identify sets of VMs, whose migration will remedy the capacity violations present at each node. There are multiple such sets, since a particular violation can be alleviated by migrating either a single large (e.g., in terms of the requested computational capacity) VM or multiple smaller ones. Each VM is assigned the information vector presented in Eq. (1) and the multi-criteria approach is used in order to find the Pareto optimal such sets, that is, the sets that cannot be eliminated from consideration by replacing them with other solutions that improve an objective without worsening another one.

In particular, at intervals of duration *T*, we check the physical nodes and links to identify resource shortages. A node *i* faces computational power shortage (a "violation") when the hosted VMs request total computational power larger than the computational capacity of that node:

$$\sum_{\mathrm{VM_n \in node\ i}} \mathrm{P_{VM_n}} > \mathrm{P_i}$$

In this case, node *i* cannot provide to its hosted VMs the required computational power.

Next, the COMAVI methodology attempts to identify the VMs whose migration can remove the computational resource shortages. At this point we are not interested in checking whether these migrations will create new resource shortages ("violations") or not. The procedure "visits" each violating node i and finds all possible sets of VMs, whose removal from node i can resolve the resource's computational capacity violation:

$$\mathrm{G_i} = \begin{Bmatrix} \mathrm{G_{i,1}} = (\mathrm{VM_{i,1,1}}, \mathrm{VM_{i,1,2}}, \cdots) \\ \mathrm{G_{i,2}} = (\mathrm{VM_{i,2,1}}, \mathrm{VM_{i,2,2}}, \cdots) \\ \vdots \\ \mathrm{G_{i,k}} = (\mathrm{VM_{i,k,1}}, \mathrm{VM_{i,k,2}}, \cdots) \end{Bmatrix}.$$

Gi will be referred to as the group of candidate migrating sets from node i. For example, for candidate migrating set $\mathrm{G_{i,1}}$:

$$\sum_{\mathrm{VM_n \in node\ i}} \mathrm{P_{VM_n}} - \sum_{\mathrm{VM_j \in G_{i,1}}} \mathrm{P_{VM_j}} \le \mathrm{P_i} \tag{2}$$

Each candidate migrating set of VMs is characterized by an information vector that is produced by the information vectors of its constituent VMs [see Eq. (1)], by applying appropriate associative operations. For example, the information vector of the candidate migrating set $\mathrm{G_{i,1}}$ is calculated as follows:

$$\mathrm{V_{G_{i,1}}} = \begin{Bmatrix} \sum_{\mathrm{VM_j \in G_{i,1}}} \mathrm{P_{VM_j}}, \\ \sum_{\mathrm{VM_j \in G_{i,1}}} \sum \mathrm{B(j)}, \\ \sum_{\mathrm{VM_j \in G_{i,1}}} \mathrm{S_j}, \\ \sum_{\mathrm{VM_j \in G_{i,1}}} |\mathrm{I_{in}(j)}|, \\ \sum_{\mathrm{VM_j \in G_{i,1}}} |\mathrm{I_{out}(j)}|, \\ |\mathrm{G_{i,1}}| \end{Bmatrix}, \tag{3}$$

that is, by adding the respective parameters of the VM information vectors. Also, $\mathrm{V_{G_{i,1}}}$ is extended by including the number $|\mathrm{G_{i,1}}|$ of its constituent VMs.

Since the number of possible migrating sets (VM combinations) increases exponentially with the number of VMs located at a node, and can be as high as 2^{n} (this corresponds to the unusual case where all VMs reside at a single node), it may be difficult to find the group Gi of all possible migrating sets for all network nodes i. To alleviate this problem, the candidate sets of VMs is reduced by applying domination relations. A possible migration set is said to be dominated by another one, when it is inferior to it with respect to all the parameters of interest. In particular, we say that information vector $\mathrm{V_{G_{i,1}}}$ dominates information vector $\mathrm{V_{G_{i,2}}}$, if $\mathrm{V_{G_{i,1}}}$ is better than $\mathrm{V_{G_{i,2}}}$ with respect to all the cost parameters. The term "better" is interpreted differently based on the parameters of interest. For the parameters defined in Eq. (1), information vector $\mathrm{V_{G_{i,1}}}$ dominates vector $\mathrm{V_{G_{i,2}}}$ if the following conditions hold:

$$\mathrm{P_1 \le P_2}, \mathrm{B_1 \le B_2}, \mathrm{S_1 \le S_2}, |\mathrm{I_{in_1}}| \ge |\mathrm{I_{in_2}}|, |\mathrm{I_{out_1}}| \le |\mathrm{I_{out_2}}|, |\mathrm{G_1}| \le |\mathrm{G_2}| \tag{4}$$

If Eq. (4) holds, information vector $\mathrm{V_{G_{i,2}}}$ can be discarded from further consideration, since the set $\mathrm{V_{G_{i,2}}}$ of VMs is inferior (as a possibility, for migration) to the set

$V_{G_{i,1}}$ with respect to all parameters: it requires more computational and communication resources, meaning that the migration of the corresponding VMs will cause more load to the new hosting nodes and the links they will use for their interconnection, the size S of data transferred during migration is larger, and more service/connection interruptions (related to parameters $|I_{in}|, |I_{out}|, |G|$) will occur due to the migration; thus making $G_{i,2}$ an inferior candidate set of migration than set $G_{i,1}$ for any reasonable objective function. If the number of candidate migration sets remains large even after applying the domination relation, it is possible to apply a less strict domination relation to further prune this group; for example, by removing all VM migration sets whose total number of interrupted connections is larger than a predefined threshold.

B. Select a candidate group of violating VMs

Next, the selected sets of violating VMs are ordered. The ordering is performed by applying an objective function to the corresponding information vector of each candidate set, which represents the importance given to each of the cost parameters. The VM set with the smallest metric value among all the candidate migration sets in all the nodes is the one first considered for migration.

The following minimization functions are considered. Objective function (5) favors the migration of VM groups from nodes with small processing requirements, expecting to increase the chances of finding destination nodes with the available resources for running the migrated VMs.

$$f(G_{i,k}) = \sum_{VM_j \in G_{i,k}} P_{VM_j} \tag{5}$$

Objective function (6) favors moving groups of VMs with few high-bandwidth external flows.

$$f(G_{i,k}) = \frac{\sum_{VM_j \in G_{i,k}} |I_{out}(j)|}{\sum_{VM_j \in G_{i,k}} \sum B_{out}(j)}. \tag{6}$$

Objective function (7) favors migrating groups of VMs with few external flows of high bandwidth and a lot of internal flows with small bandwidth.

$$f(G_{i,k}) = \frac{\sum_{VM_j \in G_{i,k}} |I_{out}(j)|}{\sum_{VM_j \in G_{i,k}} \sum B_{out}(j)} + \frac{\sum_{VM_j \in G_{i,k}} \sum B_{in}(j)}{\sum_{VM_j \in G_{i,k}} |I_{in}(j)|}. \tag{7}$$

Using objective function (8), we attempt to jointly take into account all the above parameters and considerations.

$$\begin{aligned} f(G_{i,k}) = \sum_{VM_j \in G_{i,k}} P_{VM_j}{}^{2} + \sum_{VM_j \in G_{i,k}} \sum B(j) \\ + \sum_{VM_j \in G_{i,k}} |I_{out}(j)| \\ + \sum_{VM_j \in G_{i,k}} |I_{in}(j)| + |G_{i,k}|. \end{aligned} \tag{8}$$

Other objective functions can also be defined (in the proposed mechanism) based on the interests of the resource infrastructure provider and/or the users.

C. Find sets of candidate destination nodes

In this step, the COMAVI methodology finds candidate destination nodes for the violating VMs selected in the previous step. Such destination nodes can be chosen, using various criteria (available capacity, cost, energy, etc.). In the simulation results to be reported in Section V, a predefined number D of candidate destination nodes for all migrating VMs (of all nodes), is selected, in order to bound the algorithms' execution times. These nodes have the necessary computational (e.g., CPU cores) capacity for hosting and serving the migrating VMs. Optimally, the methodology should consider all possible combinations of migrating VMs and destination nodes, but this would increase exponentially the algorithm's running time. If it is not possible to migrate a selected (from Step-2) set's VMs without increasing the total number of violations, the second best candidate VM set (from Step-2) is considered and the process is repeated.

D. Select a set of destination nodes

The purpose of this stage is to select a single set of destination nodes, among all the candidate ones, so as minimize the required networking changes and the induced network overhead (that is minimize the probability of network link overload) triggered by the VM migrations. At the same time this step attempts to improve indirectly the overall network efficiency of the communicating VMs, after their migration to their new hosting nodes completes, which is also directly depends on the resource related decisions (e.g., routing) taken. For each candidate set of destination nodes (where VMs could migrate), COMAVI creates a logical topology and selects one of them (and the corresponding set of destination nodes) by applying a selection criterion:

- Select the virtual topology with the smallest number of links:
  $$\min \sum |I'_{out}|, \tag{9}$$
  where $|I'_{out}|$ is the number of inter-connections originating from one of the selected (in Step-2) VMs, after all migrations have been completed.
- Select the virtual topology that results in the smallest number of connection changes from the original virtual topology prior to the VMs migration. This policy attempts to directly minimize the number of topology reconfigurations (additions and deletions of virtual links) performed and the related changes in the networking settings and devices.
- Select the virtual topology that results in the smallest total requested bandwidth by the intercommunicating VMs:
  $$\min \sum \sum B'_{out}, \tag{10}$$
  where $\sum B'_{out}$ is the total bandwidth requested for all inter-connections originating from a particular VM after all the migrations have been performed. This metric can be used both for the DCR and the SCR communication models.

V. SIMULATION EXPERIMENTS

We evaluated the proposed COMAVI methodology through a number of simulation experiments performed in the Matlab environment. In particular, the following COMAVI based algorithms have been considered, each using a different objective function (for Step-2) and logical topology selection policy (for Step-4):

- comavi-1: Uses the objective function of Eq. (8) and selects the logical topology that results in the smallest total bandwidth requested by the inter-communicating VMs, as in Eq. (10).
- comavi-2: Uses the objective function of Eq. (8) and selects the logical topology with the smallest number of links, as in Eq. (9).
- comavi-3: Uses the objective function of Eq. (7) and selects the logical topology that results in the smallest number of connection changes in comparison to the original logical topology prior the VMs migration.
- comavi-4: Uses the objective function of Eq. (7) and selects the logical topology that results in the smallest total bandwidth requested by the inter-communicating VMs, as in Eq. (10).

For comparison purposes we have also implemented a simple greedy sequential migration policy, which attempts to migrate each VM one by one, to a new physical node. The migration of a VM is performed only if the total number of violations after the migration is smaller than the initial one.

The main metrics of interest are:

- the number of violations present before and after the migrations occur. We focus on computational resource violations, where the processing capacity requested by the hosted VMs is larger than the corresponding physical node's computational capacity,
- the number of migrations performed, indicating the network overhead incurred, and
- the number of logical topology changes, indicating the service interruptions and the reconfigurations of the network settings and the corresponding devices.

Regarding the selected metrics, we should note that as mentioned in Section I, COMAVI does not consider how migration will be performed e.g., the order with which migrations will be performed, the networking paths that will be utilized etc.

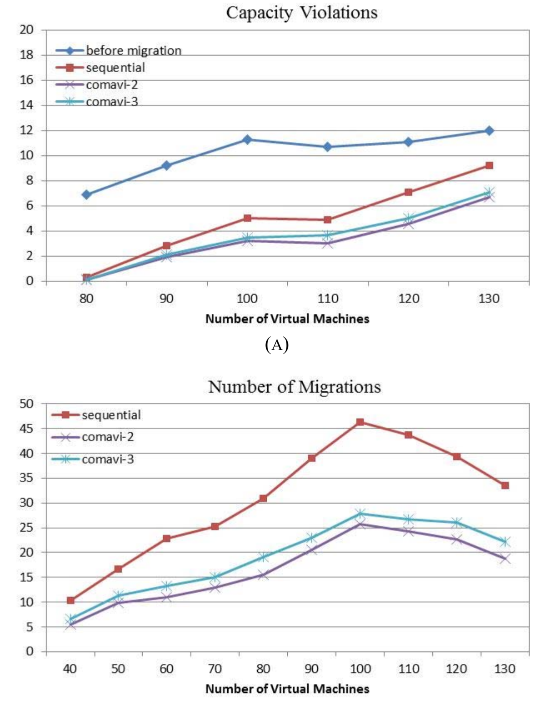
In each simulation experiment VMs number is static that is no new VMs arrive or old ones expire. Moreover, all static parameters (e.g., computational capacity, bandwidth, storage requirements etc.) are selected from a range of values using uniform distribution.

A. Effect of the Number of Virtual Machines (VMs)

Fig. 2a shows the number of violations at the nodes, before and after the various migrations policies are executed as a function of the total number of Virtual Machines (VM), in a network of 20 nodes, with interconnection load equal to 0.4. We observe that the COMAVI algorithms resolve more violations than the greedy sequential policy, with comavi-2 exhibiting slightly better performance than comavi-3 algorithm. In all cases, as the number of VMs increases, the number of capacity violations also increases. As the number of VMs increases further, the network becomes saturated and any benefits from the migrations start to diminish.

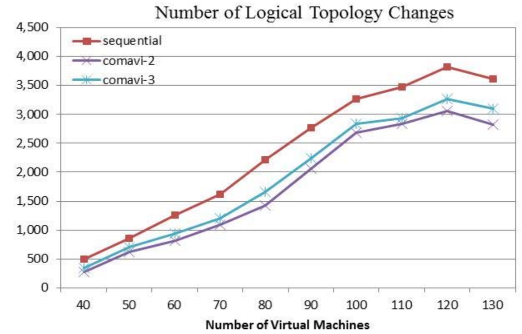
Fig. 2b shows the number of migrations performed as a function of the total number of VMs. We observe that the COMAVI policies perform considerable fewer VM migrations and cause fewer service interruptions than the greedy sequential policy, even when the number of VMs in the network is quite large. Also, the COMAVI policies seem to adjust to the fact that after a certain point the network becomes VM-saturated, and they reduce the number of VM migrations performed. These results demonstrate the importance of performing concurrently the migration of multiple VMs. Fig. 2c illustrates the number of reconfigurations in the logical topology, that is, the total number of virtual connections between nodes that were added or deleted in order to serve the VMs in their new locations. Again, we observe that the COMAVI algorithms lead to a smaller number of reconfigurations and consequently to fewer changes in the network settings. The slightly better performance of the comavi-2 algorithm is because, under the applied simulation settings, the selection of VMs that alleviate the particular violations and utilize a smaller number of CPU cores is more important than the selection of VMs based only on their connection and bandwidth requirements.

We also measured the total bandwidth requested (in Mbps) by communicating VMs located in different nodes. Note that each communicating pair of VMs requested bandwidth is uniformly distributed between 0 and 10 Mbps. We observe that using comavi-2 algorithm the total bandwidth requested by inter-communicating VMs is smaller than when using the comavi-3 or the greedy sequential policy.



(A)

(B)



(C)

Fig. 2. (a) Number of capacity violations at nodes, before and after the various migrations policies are applied, (b) Number of VM migrations performed, (c) Number of logical topology changes, in relation to the total number of running VM.

B. Effect of the Virtual Machines (VM) Interconnection Load

Fig. 3 shows the number of computational capacity violations against the VM migrations when the VMs' interconnection load increases, in a network of 20 nodes and 100 VMs. We observe that the number of violations and the number of VM migrations is not significantly affected by the VM interconnections load, in contrast to the number of logical topology reconfigurations that are greatly affected. This was reasonable to expect, since the load of the VM interconnections affects only the network and not the computing requirements of the VMs. In all cases, the COMAVI policies exhibit better performance than that of the greedy sequential policy [more cross-points near (0,0)], with comavi-2 policy performing better than comavi-3.

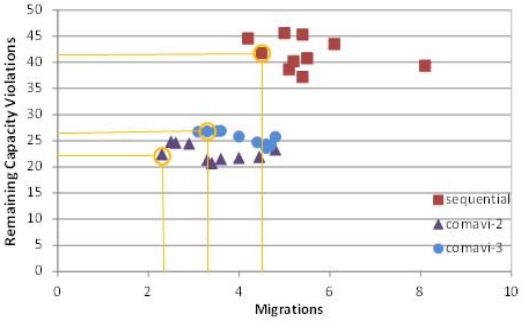


Fig. 3. Number of capacity violations at nodes, before and after the migrations policies are applied against the number of VM migrations performed; in the figure we have marked (with orange circles) the results from all policies, when the interconnection load is 90%.

Other algorithms (e.g., comavi-1, comavi-4) following the COMAVI framework were also evaluated; however the performance results obtained do not add something significant in the discussion (at least in the under the considered settings), so they are omitted for the sake of brevity.

VI. CONCLUSIONS

We presented the communication-aware virtual infrastructures (COMAVI) methodology for coordinating the concurrent migration of Virtual Machines (VMs) in computing infrastructures, so as to improve or balance resource utilization. COMAVI uses a multi-criteria approach for selecting the VMs that will migrate, while also selects the computing sites where the migrating VMs will be hosted, by considering the way migration affects the logical (or virtual) topologies formed by the communicating VMs, before and after the migration. We carried out a number of simulation experiments to evaluate the performance of the proposed algorithms, showing that they are able to resolve the maximum number of resource capacity violations, while tending to minimize the number of migrations required, the network overhead induced, and the logical topology and network settings reconfigurations.

ACKNOWLEDGMENT

This work is implemented within the framework of the Action «Supporting Postdoctoral Researchers» of the Operational Program "Education and Lifelong Learning" (Action's Beneficiary: General Secretariat for Research and Technology), and is co-financed by the European Social Fund (ESF) and the Greek State. This work has also been partially supported by the European Commission through the FP7-ORBIT project under grant agreement nº 609828.

REFERENCES

[1] S. Chen, K. Nahrstedt, On Finding Multi-Constrained Paths, IEEE International Conference on Communications (ICC), 1998.
[2] T. Korkmaz, M. Krunz, Multi-Constrained Optimal Path Selection, IEEE INFOCOM, 2001.
[3] A. Gençata, B. Mukherjee, "Virtual-topology adaptation for WDM mesh networks under dynamic traffic", IEEE/ACM Transactions on Networking, Vol. 11, No. 2, pp. 236-247, 2003.
[4] D. Arora, et. al, "Online Strategies for Intra and Inter Provider Service Migration in Virtual Networks", Computing Research Repository (CoRR), 2011.
[5] R. Bradford, et, al, "Live wide-area migration of virtual machines including local persistent state", International Conference on Virtual Execution Environments (VEE), pp.169-179, 2007.
[6] T. Wood, et. al, "CloudNet: A Platform for Optimized WAN Migration of Virtual Machines", University of Massachusetts Technical Report TR-2010-002, January 2010.
[7] S. Bose, et. al, "Cloud Spider: Combining Replication with Scheduling for Optimizing Live Migration of Virtual Machines across Wide Area Networks", IEEE CCGRID, 2011.
[8] A. Sundararaj, et. al, "Dynamic Topology Adaptation of Virtual Networks of Virtual Machines", 7th Workshop on Languages, Compilers and Run-time Support for Scalable Systems, 2004.
[9] F. Travostino, et al., " Seamless Live Migration of Virtual Machines over the MAN/WAN", FGCS, 2006.
[10] S. Bose and S. Sundarrajan, "Optimizing Migration of Virtual Machines across Data-Centers", IEEE ICPP, 2009.
[11] X. Meng, et. al, "Improving the Scalability of Data Center Networks with Traffic-aware Virtual Machine Placement", IEEE INFOCOM, 2010.
[12] V. Shrivastava, et. al, "Application-aware virtual machine migration in data centers", IEEE INFOCOM, , pp.66-70, 2011.
[13] T. Ferreto, et. al, "Server consolidation with migration control for virtualized data centers", Future Generation Computer Systems, 2011.
[14] R. Ramaswami and K. Sivarajan, "Optical Networks: A Practical Perspective", 2nd ed. San Mateo, CA: Morgan Kaufmann, 2001.
[15] R. Ramaswami and K. Sivarajan, "Design of logical topologies for wavelength-routed optical networks", IEEE Journal on Selected Areas in Communications, Vol. 40, pp. 840-851, 1996.
[16] Phuong Nga Tran, U. Killat, "Dynamic reconfiguration of logical topology for WDM networks under traffic changes," IEEE Network Operations and Management Symposium (NOMS), pp. 279-286, 2008.
[17] R. Duran, et al., "Efficient reconfiguration of logical topologies: Multiobjective design algorithm and adaptation policy", ICBCNS, 2008.
[18] H. Bazzaz, et. al, "Switching the optical divide: fundamental challenges for hybrid electrical/optical datacenter networks", ACM Symposium on Cloud Computing, 2011.
[19] VMware: http://www.vmware.com/