Social-like Analysis on Virtual Machine Communication Traces

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
Patras, Greece

Abstract— **We apply social network analysis methods on communication traces, collected from Virtual Machines (VMs) located in computing infrastructures, like a data center. Our aim is to identify important VMs, for example VMs that consume more energy or require more computational capacity, bandwidth, etc, than other VMs. We believe that this approach can handle the large number of VMs present in computing infrastructures and their interactions in the same way social interactions of millions of people are analyzed in today's social networks. We are interested in identifying measures that can locate important VMs or groups of interacting VMs, missed through other usual metrics and also capture the time-dynamicity of their interactions. In our work we use real traces and evaluate the applicability of the considered methods and measures.**

Keywords: virtual machines, communication traces, social network analysis, resource management

I. INTRODUCTION

Virtualization has acquired recent popularity as a way to increase resources utilization, enabling efficient and flexible resource management, and reduce cost of ownership. Solutions for dynamically connecting Virtual Machines (VM) and creating virtual infrastructures over physical resources have appeared either in the form of services, namely Cloud computing services (e.g., Amazon Elastic Computing Cloud - EC2) and Cloud computing toolkits (e.g., OpenNebula and Nimbus).

Today, cloud computing environments are supported mostly by data centers. The number and size of the data centers (DCs) constructed around the world is continuously increasing, both due to the need for supporting million-user services (e.g., Facebook, Amazon's AWS, Google), and also the trend towards consolidating small DCs as a way to reduce costs and optimize resource utilization. In general, in a data center, servers are organized in racks that host typically between 20 to 100 servers and each server hosting 100 or more (depending on the hardware) VMs. Though the exact size (number of servers) of the various commercial data centers is not usually reported, estimates place as high as 450.000 servers [3] while even larger ones (in the scale of 10^6 to 10^7 machines) have been envisioned for the future [4]. Multiplying these numbers with the maximum number of VMs per server, which can vary depending on the server hardware; we get a very large number of VMs running in a data center.

Resource management decisions in such large computing infrastructures are of paramount importance affecting to a large degree both their efficiency and their cost. Such management decisions include VM scheduling [5][6] and migration [7]. Moreover, the efficient use of the resources is not only measured in terms of physical resource utilization, but also based on energy usage [8], which also relates to cost. The quality of service perceived by the user utilizing such infrastructures is also important. And while this can be provided for the computational resources, through the use of VMs, it is not so clear for the communication ones. A wide variety of communication-intensive applications run in today's DCs, each involving several active intra and inter data center flows [1]. These communication flows, which tend to change dynamically in a DC environment, significantly impact performance. In particular, a tremendous amount of variability in latency and bandwidth has been reported among VM instances in a cloud service [9]. This is because cloud providers do not offer guaranteed network resources to their users [11]. Instead, a user's compute instances (VMs) share the same network with other users' VMs. Consequently, the latency incurred and the bandwidth achieved by traffic among a user's VMs depends significantly on the network load and the placement of these VMs.

In our work, we investigate methods for analyzing communication traces from DCs so as to identify important VMs, or VMs that correlate in some way. This knowledge could assist several resource management decisions; for example, by migrating in the same rack VMs that communicate with each other a lot, reducing communication overhead, or by triggering the cooling infrastructure in places of the data center where important VMs are operating. A number of other work focus on the profiling of large computing infrastructures such as HPC systems, Grids and recently DCs. In [12] the authors present a thorough analysis of the job inter-arrival times, the waiting times at the queues, the execution times, and the data sizes exchanged at a cluster of the EGEE Grid infrastructure. In [1] authors collect and analyze SNMP statistics, topology, and packet-level traces (some of these traces are also used in this work). They examine the range of applications deployed in these DCs and their placement, the flow-level and packet level transmission properties of these applications, and their impact on network and link utilization, congestion, and packet drops. In [13] the authors study the communication requirements of large-scale HPC applications. All these previous works attempt to capture the characteristics of the traffic in the corresponding

978-1-4673-2798-5/12/$31.00 ©2012 IEEE

infrastructures, while our work uses the same more or less data (traffic matrices) in order to extract information on the relationships among the VMs creating the traffic.

The analysis of communication traces is not an easy task, considering the huge number of VMs in a data center, their communication interactions and their dynamicity, as it has been exposed in [1]. Actually, this analysis resembles the social network analysis in which quantitative methods and computational tools are used to identify and answer social science questions, such as who is most influential person [14][15] (correspondingly, which is the most important VM), who is friend with who [16] (correspondingly, which VM interacts over time highly and constantly with which other VM), identify events on the social world [17] (correspondingly, identify important events in the DC). The answers to the corresponding questions for the DCs can be very useful in making resource management decisions (e.g., migrate a VM or increase the fan speed).

In this work we evaluate the applicability of a number of social network analysis methods for analyzing communication traces, collected from large computing infrastructures so as to identify important and/or interacting over time Virtual Machines (VMs). We expect that by analyzing these communication patterns we will be able to identify the VMs or groups of VMs that are important, or bandwidth hungry, or energy inefficient, or exhibit some specific property. We believe that this can be a new and highly interesting field of research in the data center community, considering also the fact that exact and detailed information for all VMs is actually difficult to collect and process. In this way many of the research efforts (methods etc) for social analysis can find their way and analogy in data center analysis / profiling, building related tools.

The remainder of the paper is organized as follows. In Section 2 we describe the communication-related VM data used and the resulting network graphs. In Section 3 we present various social network analysis measures and discuss their applicability for the analysis of data center communication traces. The application of some of these measures in real traces is presented in Section 4. Finally, Section 5 concludes the paper.

II. MODELING

We assume as input a trace where each line indicates a communication between two VMs (e.g., identified by their IP), the time of the communication and the total transmitted data. One could also use an aggregated form of such data, where each line represents a connection between two VMs, its start time, the total transmitted data and its duration. For example, in our analysis our connection data have the following format:

<timestamp> <source> <destination> <data size>

This information can be modeled as a weighted network graph $G=(N,E)$, where the nodes in N are the VMs-IP and the links in E indicate VMs exchanging data. The weight $w_{i,j}$ of an edge-link $e_{i,j} \in E$, represents the total data transmitted between the corresponding VMs, VM_i and VM_j. In what follows, we present a number of different definitions of weight $w_{i,j}$, based on whether the communication traces are aggregated over the whole period of observation, or particular slots are considered.

Analyzing the communication traces statically, by aggregating all the communication information for the given time period, is generally easier. For example, in this case the weight $w_{i,j}$ of the edge $e_{i,j}$ is equal to:

$$w_{i,j} = \sum D_{i,j}^{[0,t_{end}]}, \tag{1}$$

where $D^{[0,t_{end}]}$ is the total data transmitted in the period $[0,t_{end}]$ of observation between VM_i and VM_j. However, at the same time important information is lost, since the dynamicity of the communication is not captured. To this end we are interested in performing the network analysis using longitudinal networks. These networks evolve over time by the addition and removal of nodes, and the forming, strengthening, weakening, and ultimately, the severing of ties. In this case, the weight $w_{i,j}$ of the edge $e_{i,j}$ depends on the particular time slot of observation $[t_1, t_2]$. In particular, weight $w_{i,j}$ can be calculated either by aggregating all the past information (data transmitted) for the period $[0, t_2]$:

$$w_{i,j}^{[t_1,t_2]} = \sum D_{i,j}^{[0,t_2]}, \tag{2}$$

where $D_{[0,t_2]}$ is the total data transmitted in $[0, t_2]$ between VM_i and VM_j. Or by aggregating only the information (data transmitted) related to this time slot.

$$w_{i,j}^{[t_1,t_2]} = \sum D_{i,j}^{[t_1,t_2]}, \tag{3}$$

where $D_{[t_1,t_2]}$ is the total data transmitted in the period of observation $[t_1, t_2]$ between VM_i and VM_j.

III. MEASURES

Social network analysis uses a number of metrics and methods. In what follows we present some of the most important ones and discuss where they can be applied in the VM communication patterns scenario.

A. Centrality

Centrality refers to a group of metrics that aim to quantify the "importance" or "influence" (in a variety of senses) of a particular node (or group) within a network. A number of common methods of measuring "centrality" exist,

Degree centrality [14] is based on the idea that the central nodes in a network are those with the most connections to other nodes. It is determined by adding up the number of edges an actor has to all other nodes or their weights. In the case of a directed network (where edges have direction), we usually define two separate measures of degree centrality, namely indegree and outdegree. When edges are associated to some positive aspects, such as friendship or collaboration, indegree is often interpreted as a form of popularity, and outdegree as influential. This is a measure that can easily be used in analyzing DC traffic traces, providing a view of the most important VMs. In particular, the weighted degree centrality WCE_i of VM_i in a communication graph, is equal to:

$$WCE_i = \frac{\sum_{j \in N_i} w_{i,j}}{\sum w}, \tag{4}$$

where N_i is the set of VMs with which VM_i exchanges data in the period of observation that is the neighbors. In the definion of Eq. (4) we assume that the weighted centrality is normalized against the sum of the weights of all edges (connections). For weight $w_{i,j}$ one can use any of the definitions presented earlier, in Eq. (1), (2) and (3). Interestingly, centrality considering only the number of a VMs edges and not the total weight can also be important, indicating a high influential VM. In this case, degree centrality CE_i of a VM_i in a communication graph as the one defined in Section II, is equal to

$$CE_i = \frac{|N_i|}{|E|}, \tag{5}$$

In the definion of Eq. (5) we assume that the centrality is normalized against the sum of the total number of edges (connections).

Other centrality measures include *closeness* [14] and *betweenness centrality* [14][15], which we believe however, are not so significant in analyzing DC traces, so we focused our study on the degree centrality metric.

B. Dynamic analysis and event detection

Network dynamicity and event detection are probably the most important and interesting features for the analysis of communication traces.

Existing centrality metrics for the study of dynamically changing social networks are based on a static network model, where edges that appear (and disappear over time) are aggregated into a single static graph. Another approach is to create static snapshots at a fixed interval generating insights into how network properties change over time. In this context, temporal (betweeness and closeness) centrality metrics have been proposed to account for dynamic interactions over time [18].

Event detection is another important parameter in social network analysis. In [17], the authors analyze tweets to detect real-life events. They build a signal for individual words, by measuring the number of words appearing in tweets at fixed intervals and then applying wavelet analysis on the frequency-based raw signals of the words. The authors then build a correlation matrix, which is partitioned, with a modularity-based graph partitioning technique, so as to group of words that indicate a particular event. A similar event detection method can also be used for the analysis of the communication traces, by creating a signal per VM and counting the amount of data sent (and received) by a VM in each period / time slot:

$$S_{VM_i} = \left\{WCE_i^{[0,t_1]}, WCE_i^{[t_1,t_2]}, \cdots, WCE_i^{[t_{n-1},t_n]}\right\} \tag{7}$$

using Eq. (3) and (4). The correlation matrix that is build using these signals (actually the signals' produced after calculating their DF-IDF score and applying wavelet transformation as in [17]), provides a measure of the relation over time of all the considered VMs. The graph partitioning applied next makes possible the identification of groups of VMs that behave similarly over time, in terms of the data transmitted and/or received. The resource management system could then migrate these VMs to the same rack, e.g., so as to reduce the communication overhead. This approach can be used to identify VMs whose correlation cannot be captured by graph analysis (that is, by using centrality and clustering measures).

IV. TRACES NETWORK ANALYSIS

A. Data Traces and Tools

In our analysis we used the traces provided by [2] and presented in [1], where an analysis of the traffic characteristic of data centers was performed. In our analysis we assume that each IP in the trace corresponds to a VM; this assumption however does not in any way affects our results or conclusions. For the trace analysis, we used NetworkX, Python language software package.

B. Results

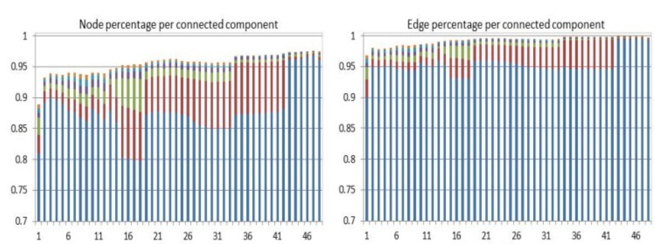
We analyzed the traces from [2], namely TRACE, obtained in an about 1 hour interval in a data center (DC). TRACE's basic characteristics are presented in the following Table. According to [1], these traces are collected from a university DC containing 500 servers.

TABLE I. TRACE BASIC CHARACTERISTICS

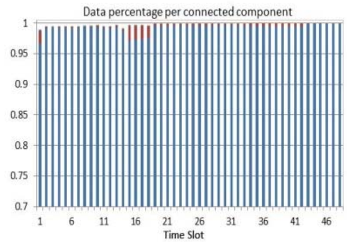
	TRACE
Duration	3914.6 sec
Lines of trace file	19,855,388
Number of unique IP	2,646,311
Total data transferred	118 GB

Initially, we attempted to identify the main characteristics of the network graph, mainly in terms of the connected components and their size. The graph was created as described in Section II, where the nodes are the VM-IPs and the links represent the existence of communication between two VMs.

In Figure 1 we try to identify the main characteristics of the seven (7) largest, in terms of their total edge weight, connected graphs. We measure the number of nodes, edges and the total weight, or else data per connected component as percentage of the corresponding totals (nodes, edges, weights). We observe that in three of these connected graphs, depending on which measure one observes, belong the overwhelming percentage of nodes-VMs, edges-connections and weight-data transferred. This of course has to do with the specific traces analyzed and the particular data center, revealing its single-dimensionality, meaning that it is used mainly for a particular kind of application.



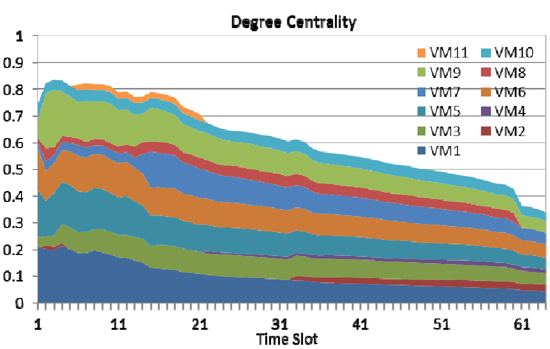
(a) **(b)**



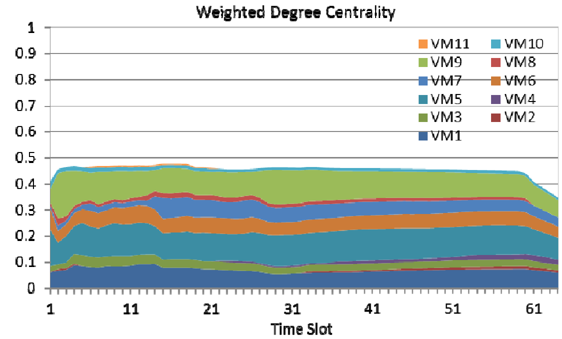
(c)

Figure 1. The number of (a) nodes, (b) edges, (c) weights of the seven largest, in terms of their total edge weight, connected graphs, as percentage of the corresponding totals (nodes, edges, weights). These metrics are calculated by aggregating over time the new nodes-VMs and edges-connections. Each time slot is equal to 60 seconds.

Next, we apply a number of social metrics in the largest of these connected graphs, attempting to identify, important VMs or groups of VM that somehow relate to each other. Figure 2 shows the degree centrality and the weighted degree centrality of the eleven most influential VMs (in terms of their transmitted data), assuming that the corresponding metrics are calculated by aggregating over time the new edges and weights, according to Eq. (2), while each time slot is equal to 60 seconds. In the figure is easy to identify the most influential VMs along with how their influence changes over time. Generally, the most important VMs (in term of their centrality), such as VM1 and VM9, remain important for all the duration of observation, while some VMs with smaller centrality values, such as VM2 and VM11, become more important (or less important) over time. The degree and weighted degree centralities provide the same information, regarding which VMs are more important than others. However, the former decrease over time while the latter remains stable. This is due to the fact that over time new connections are established and the denominator of Eq. (5) increases faster than its numerator. However, it seems that the weight (that is data transferred) of these new connections is small in comparison to the data transferred from/to the VM1-VM11, keeping the ratio of Eq. (4) more or less the same.



(a)



(b)

Figure 2. (a) The degree centrality and (b) weighted degree centrality of the eleven most influential VMs (in terms of their transmitted data). These metrics are calculated by aggregating over time the new edges and weights, according to Eq. (2). Each time slot is equal to 60 seconds.

Figure 3 show the weighted degree centrality of the eleven most influential VMs (the degree centrality figure is omitted for brevity), where the corresponding metric is calculated by aggregating the new edges and weights over a particular time slot, according to Eq. (3), with each time slot corresponding to a 60 seconds period. We observe that the graphs are not smooth as before. Also, though one can pin point which VMs are the most important over time, looking however in a particular time slot one can draw the wrong conclusions. For example in the time slots *t*=31 to (approximately) *t*=38, VM9 seems as important as VM8, something that evidently does not hold looking at Figure 2. In general, the main trade-off between Eq. (2) (as in Figure 2) and Eq. (3) (as in Figure 3) is that by using the latter is possible to overlook the importance of a VM in a particular time slot, while by using the former the past behavior of a VM is not recorder. Moreover, the use of Eq. (2) requires to store information about previous/old connections and VMs, something that may not be possible in case of many connections and VMs.

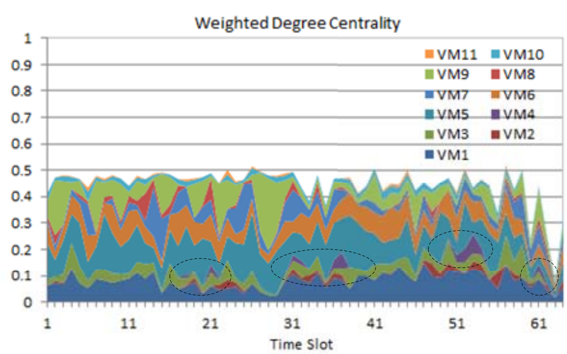


Figure 3. The weighted degree centrality of the eleven most influential VMs (in terms of their transmitted data). These metrics are calculated by aggregating the new edges and weights over a particular time slot, according to Eq. (3), with each time slot being equal to 60 seconds.

We also calculated the clustering coefficients of the graph nodes (that is, of the IPs) and discovered that it was always zero for all nodes-VMs. This has probably to do with the kind of DC monitored (university DC) and the kind of applications running on it (usually web/ftp servers) that sketch a scenario where several central machines accept connections from not correlated (at least at a first glance, as we will see next) IPs.

Figure 4 shows the weighted degree centrality, using Eq. (3), of two VMs, whose correlation has been identified using the methodology described in Subsection III.C. This is probably the most interesting result of this paper, supporting our claim that event detection methodologies can be applied in communication traces. In this figure, the event detected is that two VMs receive data at the same time slots, over a total period of an hour. These VMs do not communicate directly to each other, and their relation over time could not be identified by looking at a particular time slot, or by analyzing further the communication graph. In Figure 3 we have marked the VM5, appearing also in Figure 4, so as to illustrate that even though this is not the most important VM, its relation to VM_{other} makes it important, due to the fact that this relation can be used in subsequent resource management decisions, e.g., by moving these VMs in the same rack.

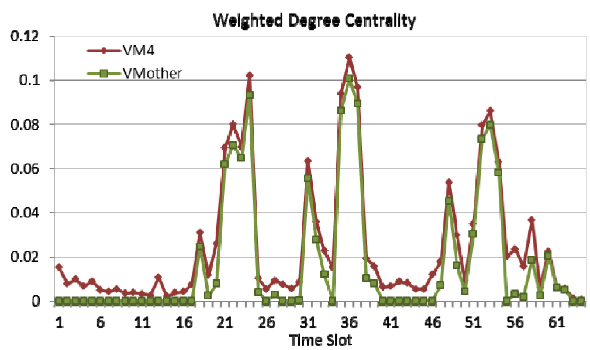


Figure 4. The weighted degree centrality of two VMs-IP, whose correlation has been identified using the methodology described in Subsection III.C.

V. CONCLUSIONS

Our work evaluates the applicability of a number of social network analysis methods for analyzing communication traces, collected from large computing infrastructures, like data centers (DCs), in order to identify VMs or groups of VMs that are important, or who employ some kind of relation that spans over time. In particular, we evaluate the applicability of (weighted) degree centrality, of clustering coefficient and of an event detection methodology in traces provided by [2] and used in [1]. Through this analysis we identify the characteristics of the formulated communication graphs, along with the VMs that are most important, in terms of the data transferred. We also discuss the pros and cons of keeping historical data regarding the communication behavior of the VMs versus of concentrating in each time period separately. In addition, the applied event detection methodology was able to identify VMs that correlated over time in way which could not be identified (possibly) through any other mean. We believe that the use of social metrics for identifying import or related VMs can be a highly interesting field of research in the DC community. In this way many of the research efforts for social analysis can find their way and analogy in data center analysis / profiling, building related tools.

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.

REFERENCES

[1] T. Benson, A. Akella, and D. A. Maltz, "Network traffic characteristics of data centers in the wild", 10th Annual Conference on Internet measurement (IMC '10), pp 267-280, 2010.

[2] Data Set for IMC 2010 Data Center Measurement, http://pages.cs.wisc.edu/~tbenson/IMC10_Data.html

[3] Amazon data center size: http://huanliu.wordpress.com/2012/03/13/amazon-data-center-size/

[4] Designs, Lessons and Advice from Building Large Distributed Systems, http://www.cs.cornell.edu/projects/ladis2009/talks/dean-keynote-ladis2009.pdf

[5] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, "Virtual Infrastructure Management in Private and Hybrid Clouds", IEEE Internet Computing, Vol. 13, No. 5, pp. 14-22, 2009.

[6] X. Meng, V. Pappas, L. Zhang, "Improving the Scalability of Data Center Networks with Traffic-aware Virtual Machine Placement", IEEE INFOCOM, 2010.

[7] V. Shrivastava, P. Zerfos, L. Kang-won, H. Jamjoom, L. Yew-Huey, S. Banerjee, "Application-aware virtual machine migration in data centers", IEEE INFOCOM, , pp.66-70, 2011.

[8] A. Beloglazov, R. Buyya, Y.-C. Lee, A. Zomaya, "A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing Systems", Advances in Computers, Vol. 82, pp. 47-111, 2011.

[9] D. Mangot, "Measuring EC2 system performance", http://tech.mangot.com/roller/dave/entry/ec2_variability_the_numbers_revealed

[10] T. Benson, A. Akella, D. Maltz, "Network traffic characteristics of data centers in the wild", Conference on Internet measurement (IMC), pp. 267-280, 2010.

[11] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron, "Towards predictable datacenter networks", ACM SIGCOMM, 2011.

[12] M. Oikonomakos, K. Christodoulopoulos, and E. (Manos) Varvarigos, "Profiling Computation Jobs in Grid Systems", Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGRID '07), pp. 197-204, 2007.

[13] S. Kamil, L. Oliker, A. Pinar, J. Shalf, "Communication Requirements and Interconnect Optimization for High-End Scientific Applications", IEEE Transactions on Parallel and Distributed Systems (TPDS), 2009.

[14] L. Freeman, "Centrality in networks: I. conceptual clarification", Social Networks, 1979.

[15] U. Brandes, "A Faster Algorithm for Betweenness Centrality", Journal of Mathematical Sociology, 2001.

[16] T. Opsahl, P. Panzarasa, "Clustering in weighted networks", Social Networks, Vol. 31, No. 2, pp. 155-163, 2009.

[17] J. Weng, Y. Yao, E. Leonardi, F. Lee, "Event Detection in Twitter", HP Laboratories, 2011.

[18] J. Tang, M. Musolesi, C. Mascolo, V. Latora, and V. Nicosia, "Analysing information flows and key mediators through temporal centrality metrics", 3rd Workshop on Social Network Systems (SNS '10), 2010.

[19] NetworkX: http://networkx.lanl.gov/