Consiglio Nazionale delle Ricerche
Istituto di Calcolo e Reti ad Alte Prestazioni

Multi-Resource Workload Consolidation in Cloud Data Centers

Carlo Mastroianni[1,2], Michela Meo[3], Giuseppe Papuzzo[1,2]

(1) Eco4Cloud – www.eco4cloud.com
(2) ICAR-CNR
(3) Department of Electronics and Telecommunications at Politecnico di Torino, Italy

Technical Report ICAR-CS 2013/02, September 2013

National Research Council, Institute for High Performance Computing and Networking (ICAR-CNR)
Via P. Bucci 41C, 87036 Rende, Italy, URL: *www.icar.cnr.it*

Multi-Resource Workload Consolidation in Cloud Data Centers

Carlo Mastroianni
ICAR-CNR and
eco4cloud srl
via P Bucci 41C
87036 Rende (CS), Italy
Email: mastroianni@icar.cnr.it

Michela Meo
Department of Electronics and
Communications, Politecnico di Torino
Corso Duca degli Abruzzi, 24
10129 Torino, Italy
Email: michela.meo@polito.it

Giuseppe Papuzzo
ICAR-CNR and
eco4cloud srl
via P Bucci 41C
87036 Rende (CS), Italy
Email: papuzzo@eco4cloud.com

***Abstract*—Power efficiency is one of the main issues that will drive the design of data centers, especially of those devoted to provide Cloud computing services. In virtualized data centers, consolidation of Virtual Machines (VMs) on the minimum number of physical servers has been recognized as a very efficient approach, as this allows unloaded servers to be switched off or used to accommodate more load, which is clearly a cheaper alternative to buy more resources. The consolidation problem must be solved on multiple dimensions, since in modern data centers CPU is not the only critical resource: depending on the characteristics of the workload other resources, e.g. RAM and bandwidth, can become the bottleneck. The problem is so complex that centralized and deterministic solutions are practically useless in large data centers with hundreds or thousands of servers. This paper presents a self-organizing approach for the consolidation of VMs on two resources, namely CPU and RAM. Decisions on the assignment and migration of VMs are driven by probabilistic processes and are based exclusively on local information, which makes the approach very simple to implement. Both a fluid-like mathematical model and experiments on a real data center show that the approach rapidly consolidates the workload, and CPU-bound and RAM-bound VMs are balanced, so that both resources are exploited efficiently.**

I. Introduction

All main trends in information technology, e.g., Cloud Computing and Big Data, are based on large and powerful computing infrastructures. The ever increasing demand for computing resources has led companies and resource providers to build large warehouse-sized data centers, which require a significant amount of power to be operated and hence consume a lot of energy. It has been estimated by Gartner that in 2006 the energy consumed by IT infrastructures in USA was about 61 billion kWh, corresponding to 1.5% of all the produced electricity and 2% of the global carbon emissions, which is equal to the aviation industry, and these figures are expected to double every 5 years [1].

In the past few years important results have been achieved, especially by improving the efficiency of cooling and power supplying facilities in data centers. The Power Usage Effectiveness (PUE) index, defined as the ratio of the overall power entering the data center and the power devoted to computing facilities, had typical values between 2 and 3 only a few years ago, while now big Cloud companies have reached values lower than 1.2. However, much space remains for the optimization of the computing facilities themselves. It has been estimated that only 20-30% of the total capacity of servers is used on average [2][3]. Unfortunately, power consumption is not proportional to the server utilization: an active but idle server consumes between 50% and 70% of the power consumed when it is fully utilized [4] meaning that a large amount of energy is used even at low utilization.

The *virtualization* paradigm can be exploited to alleviate the problem: applications are not assigned directly to servers, but are first associated to Virtual Machine (VM) instances, many of which can be executed on the same physical server. This enables the *consolidation* of the workload, which consists in allocating the maximum number of VMs in the minimum number of physical machines [5]. Consolidation allows unneeded servers to be put into a low power state or switched off (leading to energy saving and OpEx reduction), or devoted to the execution of incremental workload (leading to CapEx savings, thanks to the reduced need for additional servers).

Unfortunately, efficient VM consolidation is hindered by the inherent complexity of the problem. The optimal assignment of VMs to the servers of a data center is analogous to the NP-hard "Bin Packing Problem", the problem of assigning a given set of items of variable size to the minimum number of bins taken from a given set. To make things even worse, the problem is complicated by two circumstances: (i) the assignment of VMs should take into account multiple server resources at the same time, for example CPU and RAM, therefore it is formally a "multi-dimensional bin packing problem", much more difficult than the single dimension problem; (ii) even when a good assignment has been achieved, the VMs continuously modify their hardware requirements, potentially baffling the previous assignment decisions in a few hours.

In [6] we presented ecoCloud, an approach for consolidating VMs according to a single computing resource, i.e., the CPU. Here the approach is extended to the multi-dimension problem, and is presented for the specific case in which VMs are consolidated with respect to two resources: CPU and RAM.

With ecoCloud, VMs are consolidated using two types of probabilistic procedures, for the *assignment* and the *migration* of VMs. Both procedures aim at increasing the utilization of servers and consolidating the workload dynamically, with the

twofold objective of saving electrical costs and respecting the Service Level Agreements stipulated with users, especially concerning the expected quality of service. All this is done demanding the key decisions to single servers

The scenario is pictured in Figure 1: an application request is transmitted from a client to the data center manager, which selects a VM that is appropriate for the application on the basis of application characteristics such as the amount of required resources (CPU, RAM memory, disk) and the type of operating system. Then, the VM is assigned to one of the available servers through the *assignment procedure*.

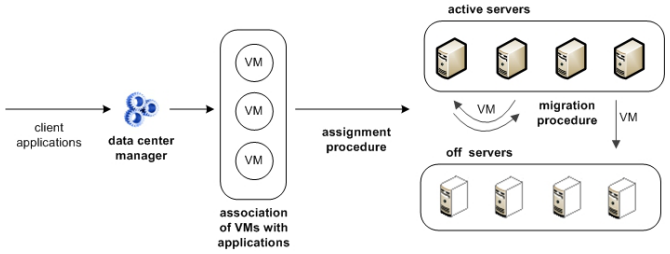

Fig. 1. Assignment and migration of VMs in a data center.

Upon an invitation from the central manager, a single server autonomously decides whether to give or deny its availability to accept a VM. Decisions are based on information available locally – for example, information on the local CPU and RAM utilization – and are founded on Bernoulli trials. The data center manager has only a coordinating role, and it does not need to execute any complex centralized algorithm to optimize the mapping of VMs.

The workload of each application typically changes with time: for example, the CPU demand of a Web server depends on the workload generated by Web users. Therefore, the assignment of VMs is monitored continuously and is tuned through the *migration procedure*. Migrating a VM can be advantageous either when the utilization of a hardware resource is too low, meaning that that resource is under-utilized, or when it is too high, possibly causing overload situations and quality of service violations.

The rest of the paper is organized as follows: Section II defines and illustrates the assignment and migration procedures, generalized for the multi-resource consolidation problem. Section III analyzes the assignment procedure through a mathematical model based on differential equations and shows that ecoCloud is able not only to consolidate the load but also to efficiently balance the available resources between compute-intensive and memory-intensive applications. Section IV reports the results of the ecoCloud adoption in a real data center of a telecommunications company, extending the assessment to the migration procedure. Section V illustrates related work and Section VI concludes the paper.

II. Assignment and Migration Procedures

In this section we describe the two main probabilistic procedures that are at the basis of ecoCloud: the assignment and migration procedures. The allocation of VMs is driven by the availability of CPU and RAM on the different servers.

The *assignment procedure* is performed when a client asks the data center to execute a new application. Once the application is associated to a compatible VM, the data center manager must assign the VM to one of the servers for execution. Instead of taking the decision on its own, which would require the execution of a complex optimization algorithm for a inherently intractable problem, the manager delegates a main part of the procedure to single servers. Specifically, it sends an invitation to all the active servers, or to a subset of them, depending on the data center size and architecture[1], to check if they are available to accept the new VM. Each server takes its decision whether or not to accept the invitation, trying to contribute to the consolidation of the workload on as few servers as possible. The invitation should be rejected if the server is over-utilized or under-utilized on either of the two considered resources, CPU and RAM. In the case of over-utilization, the rationale is to avoid overload situations that can penalize the quality of service perceived by users, while in the case of under-utilization the objective is to put the server in a sleep mode and save energy, so the server should refuse new VMs and try to get rid of those that are currently running. Conversely, a server with intermediate utilization should accept new VMs to foster consolidation.

The server decision is taken performing a Bernoulli trial. The success probability for this trial is equal to the value of the *overall assignment function* that, in turn, is defined by evaluating the *assignment function* on each resource of interest. If x (valued between 0 and 1) is the relative utilization of a resource, CPU or RAM, and T is the maximum allowed utilization (T=0.8 means that the resource utilization cannot exceed 80% of the server capacity), the assignment function is equal to zero when $x > T$, otherwise it is defined as:

$$f(x, p, T) = \frac{1}{M_p} x^p (T - x) \quad 0 \leq x \leq T \tag{1}$$

where p is a shape parameter, and the factor M_p is used to normalize the maximum value to 1 and is defined as:

$$M_p = \frac{p^p}{(p+1)^{(p+1)}} T^{(p+1)} \tag{2}$$

Figure 2 shows the graph of the single-resource assignment function (1) for some values of the parameter p, and $T = 0.9$. The value of p can be used to modulate the shape of the function. Indeed, the value of x at which the function reaches its maximum - that is, the value at which assignment attempts succeed with the highest probability - is $p/(p+1)T$, which increases and approaches T as the value of p increases. The value of the function is zero or very low when the resource is over-utilized or under-utilized.

[1]Data centers are equipped with high-bandwidth networks that naturally support broadcast messaging. In very large data centers, the servers may be distributed among several groups of servers: in this case, the invitation message may be broadcast to one of such groups only.

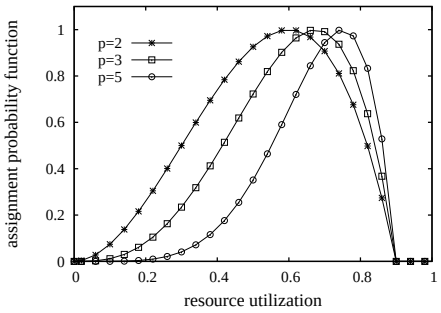

Fig. 2. Assignment probability function $f(x, p, T)$ for three different values of the parameter p and T equal to 0.9.

If u_s and m_s are, respectively, the current CPU and RAM utilization at server s, the overall assignment function is obtained by the product of two assignment functions as in (1), where $x = u_s$ and $x = m_s$ are used, respectively, for CPU and RAM. Let p_u and p_m be the shape parameters defined for the two resources, and T_u and T_m the respective maximum utilizations. The overall assignment function for the server s is denoted as f_s and defined as:

$$f_s(u_s, m_s, p_u, p_m, T_u, T_m) = f(u_s, p_u, T_u) \cdot f(m_s, p_m, T_m) \tag{3}$$

The shape of the assignment functions, combined with the definition of function (3), ensures that servers tend to respond positively when they have intermediate utilization values for both CPU and RAM: if one of the resources is under- or over-utilized the probability of the Bernoulli trial is low.

If the Bernoulli trial is successful, the server communicates its availability to the data center manager. Then, the manager selects one of the available servers, and assigns the new VM to it. If none of the contacted servers is available – i.e., all the Bernoulli trials are unsuccessful – it is very likely that in all the servers one of the two resources (CPU or RAM) is close to the utilization threshold[2]. This usually happens when the overall workload is increasing, so that the current number of active servers is not sufficient to sustain the load. In such a case, the manager wakes up an inactive server and requests it to run the new VM. The case in which there is no server to wake up, because all the servers are already active, is a sign that altogether the servers are unable to sustain the load even when consolidating the workload: when this situation occurs, the company should consider the acquisition of new servers.

The assignment process efficiently consolidates the VMs, as shown later in Section III, but application workload changes with time. When some VMs terminate or reduce their demand for server resources, it may happen that the server becomes under-utilized leading to a lower energy efficiency. On the other hand, when the VMs increase their requirements, a server may be overloaded, possibly causing SLA violation events and affecting the dependability of the data center. In both these situations, under-utilization and over-utilization of servers, some VMs can be profitably migrated to other servers, either to switch off a server, or to alleviate its load.

The *migration procedure* is defined as follows. Each server monitors its CPU and RAM utilization (a very simple operation that can be executed every few seconds) and checks if it is between two specified thresholds, the lower threshold T_l and the upper threshold T_h. When this condition is violated, the server evaluates the corresponding probability function, $f^l_{migrate}$ or $f^h_{migrate}$, and performs a Bernoulli trial whose success probability is set to the value of the function. If the trial is successful the server requests the migration of one of the local VMs. Denoting by x the utilization of a given resource, the migration probability functions are defined as follows:

$$f^l_{migrate} = (1 - x/T_l)^{\alpha} \tag{4}$$

$$f^h_{migrate} = \left(1 + \frac{x-1}{1-T_h}\right)^{\beta} \tag{5}$$

The functions, whose graphs are shown in Figure 3, are defined so as to trigger the migration of VMs when the utilization is, respectively, below the threshold T_l or above the threshold T_h. These two kinds of migrations are also referred to as "low migrations" and "high migrations" in the following. The shape of the functions can be modulated by tuning the parameters α and β, which can therefore be used to foster or hinder migrations. The same function is applied to CPU and RAM, but the parameters, T_l, T_h, α and β can have different values for the two resources. It may be useful to remark that the over- or under-utilization of a single resource is sufficient to trigger the migration procedure. In the case of over-utilization the obvious rationale is that the overloaded resource becomes a bottleneck for the server. On the other hand the under-utilization of a single resource is a hint that the consolidation is not optimal on that resource, and the migration of one or more VMs may be profitable.

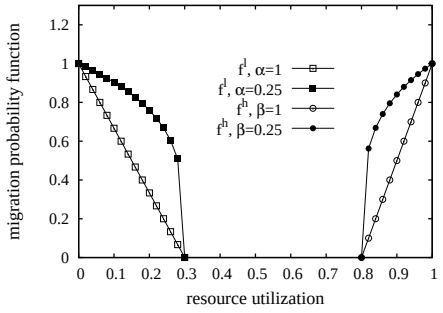

Fig. 3. Migration probability functions $f^l_{migrate}$ and $f^h_{migrate}$ (labeled as f^l and f^h) for two different values of the parameters α and β. In this example, the threshold T_l is set to 0.3, T_h is set to 0.8.

Whenever a Bernoulli trial is performed with success, the server must choose the VM to consider for migration. In the case of high migration, the server focuses on the over-utilized resource (CPU or RAM) and considers the VMs for which

[2]The case that all or many servers are not available because under-utilized on both resources is very unlikely because the process tends to consolidate the workload on highly utilized servers.

the utilization of that resource is larger than the difference between the current server utilization and the threshold T_h. Then one of such VMs is randomly selected for migration, as this will allow the utilization to go below the threshold[3]. In the case of low migration the choice of the VM to migrate is made randomly.

The choice of the new server that will accommodate the migrating VM is made using a variant of the assignment procedure described previously, with two main differences. The first one concerns the migration from an overloaded server: the threshold T of the assignment function is set to 0.9 times the resource utilization of the server that initiated the procedure, and this value is sent to servers along with the invitation. This ensures that the VM will migrate to a less loaded server, and prevents ping-pong situations in which a VM is continuously migrated from an overloaded server to another. The second difference concerns the migration from a lightly loaded server. When no server is available to run a migrating VM, it would not be acceptable to switch on a new server in order to accommodate the VM: one server would be activated to let another one be hibernated. Therefore, when no server is available, the VM is not migrated at all.

It is worth noting that our approach ensures a gradual and continuous migration process, while most other techniques recently proposed for VM migration (some are discussed in the related work section) require the simultaneous migration of many VMs.

III. Mathematical Analysis

This section is devoted to the analysis of the ecoCloud assignment procedure. The mathematical model is based on a set of differential equations inspired by fluid dynamics problems. Let N_s be the number of servers in a data center, and N_c the number of cores in each server. The equations model the evolution with time of the CPU and RAM utilization of the servers, respectively denoted by $u_s(t)$ and $m_s(t)$ for server s, with $s = 0, \cdots, N_s - 1$. The utilization of both resources is a real number that changes by infinitesimal increments/decrements over the interval $[0, 1]$.

It is assumed that two types of VMs are executed on the data center: CPU-bound and RAM-bound VMs, respectively indicated as C-type and M-type. C-type VMs need an amount of CPU that is larger than the amount needed by M-type VMs of a factor $\gamma_C > 1$; conversely, the amount of RAM required by M-type VMs is larger than the one needed by C-type VMs by a factor $\gamma_M > 1$. Given the fluid model assumption described above, the VM arrival process is a continuous process that makes it arrive, in a time period Δt, an amount of VMs that is $\lambda^{(C)}(t)\Delta t$ for C-type VMs and $\lambda^{(M)}(t)\Delta t$ for M-type VMs. The rate at which services are completed is denoted by μ.

To analyze the two classes of VMs separately, we also define the following state variables: $u_s^{(C)}(t)$ and $u_s^{(M)}(t)$ are the amount of CPU that in a server s is occupied by C-type and M-type VMs, respectively; while $m_s^{(C)}(t)$ and $m_s^{(M)}(t)$ are the amounts of RAM occupied by the two types of VMs. The total utilization of CPU and RAM in server s is given by the sum of the utilization of the two classes of VMs,

$$\begin{aligned} u_s(t) &= u_s^{(C)}(t) + u_s^{(M)}(t) \\ m_s(t) &= m_s^{(C)}(t) + m_s^{(M)}(t) \end{aligned}$$

Since the probability of assigning a VM to a server increases with the value of the assignment function, in the model the fraction of workload assigned to a server s is proportional to the acceptance probability $f_s(u_s(t), m_s(t), p_u, p_m, T_u, T_m)$, as defined in expression (3). In the following, the acceptance probability is simply denoted as $f_s(t)$.

The set of differential equations (with server index $s = 0, \cdots, N_s - 1$) is the following:

$$\frac{\partial u_s^{(C)}(t)}{\partial t} = -N_c \cdot \mu \cdot u_s^{(C)}(t) + K \cdot \gamma_C \cdot \lambda^{(C)}(t) \cdot f_s(t) \quad (6)$$

$$\frac{\partial u_s^{(M)}(t)}{\partial t} = -N_c \cdot \mu \cdot u_s^{(M)}(t) + K \cdot \lambda^{(M)}(t) \cdot f_s(t)$$

$$\frac{\partial m_s^{(C)}(t)}{\partial t} = -N_c \cdot \mu \cdot m_s^{(C)}(t) + K \cdot \lambda^{(C)}(t) \cdot f_s(t)$$

$$\frac{\partial m_s^{(M)}(t)}{\partial t} = -N_c \cdot \mu \cdot m_s^{(M)}(t) + K \cdot \gamma_M \cdot \lambda^{(M)}(t) \cdot f_s(t)$$

K is a normalization factor K, defined as:

$$K = \frac{1}{\sum_{i=0}^{N_s-1} f_s(t)}$$

The equations can be solved with the initial conditions that define the state of the system at the time that ecoCloud is executed:

$$u_s^{(C)}(0), u_s^{(M)}(0), m_s^{(C)}(0), m_s^{(M)}(0) \quad s = 0, \cdots, N_s - 1 \quad (7)$$

To analyze the behavior of the system, we performed an experiment for a data center with N_s=100 servers, each having $N_c = 6$ cores with CPU frequency of 2 GHz and 4 GB RAM. In the experiment, the VMs have nominal CPU frequency of 500 MHz. The average time the VM spent in service, $1/\mu$, is set to 100 minutes. The average CPU (memory) load of the data center is defined as the ratio between the total amount of CPU (RAM) required by VMs and the corresponding CPU (RAM) capacity of the data center, is denoted as ρ_C (ρ_M) and is computed as $\lambda^{(C)}/\mu_T$ ($\lambda^{(M)}/\mu_T$). Here, μ_T is the overall service rate of the data center, obtained as $\mu_T = \mu N_s N_c N_v$, where N_v is the number of VMs that can be executed on a single 2 GHz core, in this case 4. To analyze the system with a specified overall CPU or memory load, the arrival rates $\lambda^{(C)}$ and $\lambda^{(M)}$ must be set accordingly. In the first set of

[3]If no VM matches the condition, the largest VM will be chosen and a new Bernoulli trial will be executed to trigger another migration.

experiments, values of $\lambda^{(C)}$ and $\lambda^{(M)}$ are set to 9.6. With these values the overall load of the data center, is equal to 0.40 for both CPU and RAM: ρ_C=ρ_M=0.4.

The experiment started from a non consolidated scenario: for each server, initial CPU and RAM utilizations are set using a Gamma probabilistic function having average equal to 40 percent of the server capacity. The parameters of the assignment function were set as follows: maximum utilization threshold T=0.9, p=3. Under normal operation, without using ecoCloud, the data center would tend to a steady condition in which all the servers remain active with CPU and RAM utilization around 40 percent. With ecoCloud, the workload consolidates to only 45 servers, while 55 are switched off. This allows the data center to nearly halve the consumed power, from more than 20 kW to about 11 kW.

It was assumed that VMs are equally shared between compute-intensive (C-type) and memory-intensive applications (M-type). We considered the values of γ_C and γ_M, i.e., the ratios between the CPU and RAM demanded by the two types of VMs. The values of the two parameters were kept equal to one another, and in different tests were set to: 1.0 (the two kinds of applications coincide), 1.5 (C-type applications need 50% more CPU than M-type ones, and M-type applications need 50% more RAM than C-type ones), 2.0, and 4.0 as the most extreme case. At the end of the consolidation process, i.e., after about two hours of the modeled time, the 45 active servers show nearly the same distribution of their hardware resources between the two types of applications. This distribution is shown in Figure 4 for one of the active servers and for the above-mentioned values of γ_C and γ_M. The most interesting outcome of this experiment is that the probabilistic assignment process balances the two kinds of VMs so that neither the CPU or the RAM becomes a bottleneck. For example, in the most imbalanced scenario (γ_C and γ_M equal to 4.0), about 71% of the CPU is assigned to C-type VMs while about 18% is given to M-type VMs, and the opposite occurs for memory. Both CPU and RAM are utilized up to the permitted threshold (90%) and the workload is consolidated efficiently, which allows 55 servers to be hibernated and the consumed power to be almost halved.

Of course, such an efficient consolidation is possible when the relative overall loads of CPU and RAM are comparable (both equal to 40% in this case). If one of the two resources undergoes a heavier demand, that resource inevitably limits the consolidation degree. For such a case, it is still interesting to assess the behavior of the assignment algorithm. To this purpose, we run experiments in which the overall CPU load, ρ_C, is set to 40% of the total CPU capacity of the servers, while the overall RAM load, ρ_M, is varied between 20% and 60%. This is accomplished by appropriately varying the value of $\lambda^{(M)}$, the arrival frequency of M-type VMs. For this set of experiments, the values of γ_C and γ_M are set to 4.0. The CPU and RAM utilizations observed for each server after the consolidation phase are shown in Figure 5. Correspondingly, Figures 6 and 7 report the number of active servers and the average value of consumed power. When the overall memory

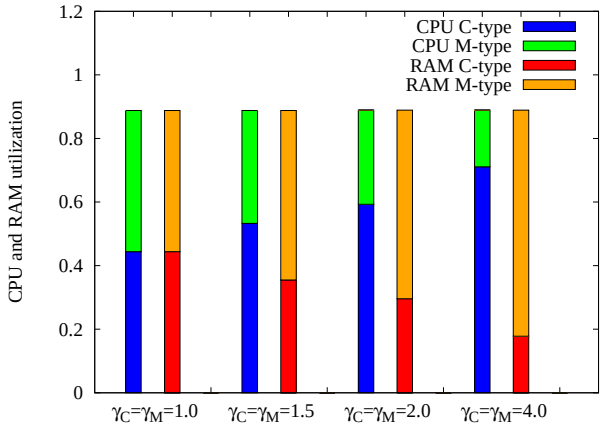

Fig. 4. CPU and RAM utilization of active servers, with different values of γ_C and γ_M.

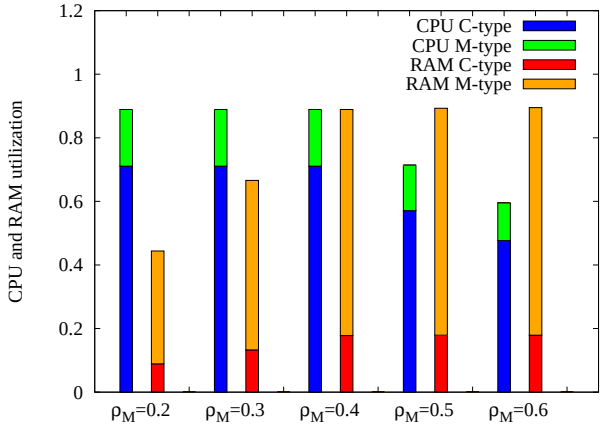

Fig. 5. CPU and RAM utilization of active servers, with different values of ρ_M, and ρ_C=0.4.

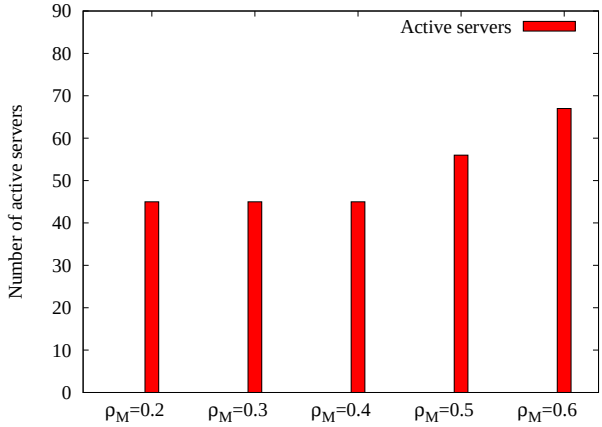

Fig. 6. Number of active servers with different values of ρ_M, and ρ_C=0.4.

load is lower than 0.4 (cases ρ_M=0.2 and ρ_M=0.3), the CPU is the critical resource and is the one that drives the consolidation process. The number of active servers (45), and the consumed power (about 11 kW) are the same as in the case where CPU and RAM overall loads are comparable. On the other hand,

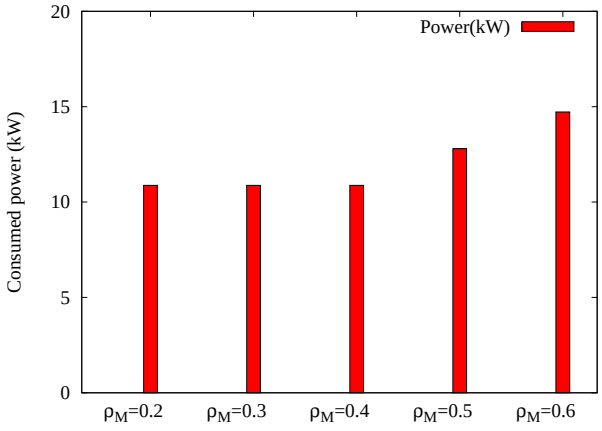

Fig. 7. Consumed power with different values of ρ_M, and ρ_C=0.4.

when the most critical resource is the memory, as happens in the cases ρ_M=0.5 and ρ_M=0.6, the consolidation process is driven by the allocation of RAM to the VMs. More active servers and more power are needed to satisfy the increased demand for memory: in the cases that the memory load is equal to 50% and 60% of the data center capacity, 56 and 67 servers are kept active, respectively, and corresponding values of consumed power are equal to about 13 kW and about 15 kW. Overall, it may be concluded that the approach is always able to consolidate the load as much as is allowed by the most critical hardware resource.

IV. Experiments on a real data center

This section reports the results of the experiments performed in May 2013 on a data center owned by a major Italian telecommunications operator. The experiment was run on 28 servers virtualized with the platform VMWare vSphere 4.0. Among the servers, 2 are equipped with processor Xeon 32 cores and 256 GB RAM, 8 with processor Xeon 24 cores and 100 GB RAM, 11 with processor Xeon 16 cores and 64 GB RAM and 7 with processor Xeon 8 cores and 32 GB RAM. The servers hosted 447 VMs which were assigned a number of virtual cores varying between 1 to 4 and an amount of RAM varying between 1 GB and 16 GB.

The VMs were categorized into CPU-bound (C-type) and memory-bound (M-type) depending on their usage of the two resources. We took as a reference the overall CPU and memory capacity of the data center that were equal, respectively, to 1171 GHz and 2334 GBytes. A VM was classified as CPU-bound if, at the end of the analyzed period, the average ratio between its CPU and memory utilization was higher than the ratio between the CPU and memory capacity of the data center. In the opposite case, it was classified as memory-bound. In this data center, 75 percent of the VMs, 335, were memory-bound, with an average usage of CPU and RAM of, respectively, 0.382 GHz and 3.25 GB. The remaining 112 CPU-bound VMs had average values of CPU and RAM of 1.76 GHz and 1.58 GB, respectively. The M-type VMs contributed for about 40% of the overall CPU load and for about 90% of the overall memory load.

While the analytical study presented in Section III focuses on the assignment procedure, during the real experiments both the assignment and the migration were activated. VMs are migrated either when the CPU or memory load exceeds the high threshold T_h, set to 0.95, or goes below the low threshold T_l, set to 0.5. Values of α and β, in expressions (4) and (5), were set to 0.25. The parameters of the assignment function were set as in the mathematical analysis: T=0.9, p=3.

Figure 8 shows the number of active servers starting from the time at which ecoCloud is activated and for the following 12 days. Within the first three days 11 servers, our of 28, are switched off thanks to the workload consolidation. In the following days, the number of active servers is stabilized. Figure 9 shows that the consumed power reduces thanks to consolidation. Figure 10 reports the number of high and low migrations performed during each day of the analyzed period on the whole data center. In the first days after the activation of ecoCloud, migrations are mostly from low utilized servers, which are first unloaded and then switched off. As the consolidation process proceeds, active servers tend to be well utilized and some high migrations are needed to prevent overload events. The number of migrations stabilizes to definitely acceptable values: for example, in the last two days no more than four migrations per day are performed.

Figures 11 and 12 offer a snapshot of the data center at the end of the twelfth day of ecoCloud operation, when only 17 of 28 are still active. The first figure reports, for each of the 28 servers, the amount of CPU and RAM utilized by C-type and M-type VMs. Since in this scenario most VMs are memory-bound, the consolidation is driven by RAM: in the majority of active servers the RAM utilization is over 70%, three servers have a RAM utilization between 60% and 70%, and a single server – the one labeled as server *6* – has a RAM utilization lower than 50%. The consolidation is made possible by the fact that VMs of the two types are distributed among the servers in a proportion that never diverts too much from the overall proportion observed in the whole data center. This is clear from Figure 12, which reports the numbers of

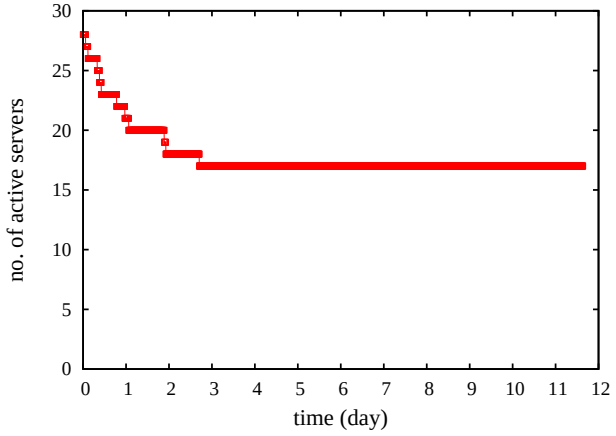

Fig. 8. Number of active servers after activation of ecoCloud.

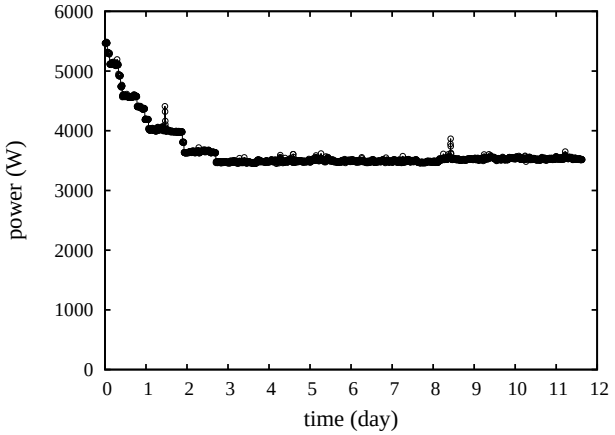

Fig. 9. Consumed power after activation of ecoCloud.

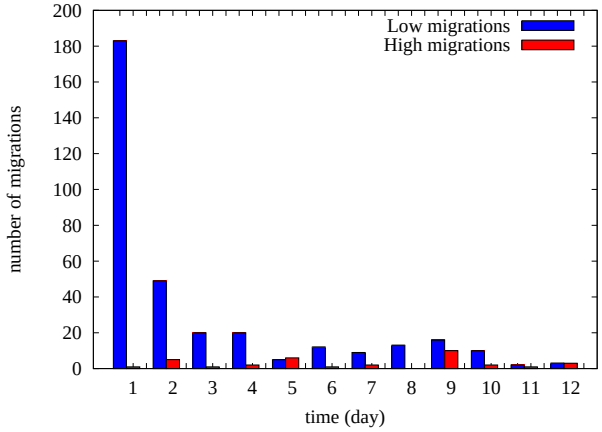

Fig. 10. Number of VM migrations after activation of ecoCloud.

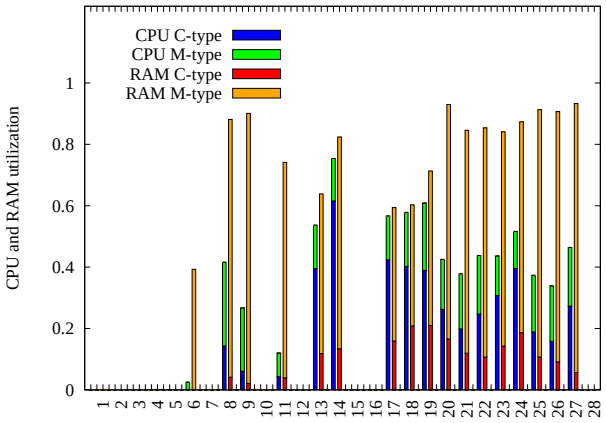

Fig. 11. RAM and CPU utilization on the 28 servers, separated for the C-type and M-type VMs. Values are taken at the end of the 12th day of operation.

VMs of the two types that run on each server. With the only exception of server *6*, in which no C-type VM is running, in all the servers the proportion between the two types of VMs is comparable to the 80-20 proportion observed in the data center. The absolute numbers are different because servers are not homogeneous.

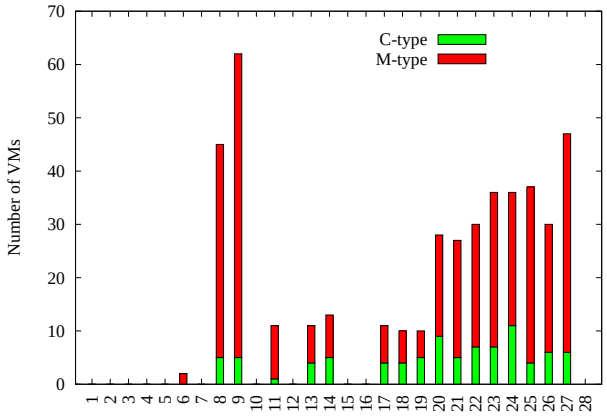

Fig. 12. Number of C-type and M-type VMs running on the 28 servers. Values are taken at the end of the 12th day of operation.

V. Related Work

Recently, a notable amount of studies has focused on algorithms and procedures that aim at improving the "green" and energy-efficient characteristics of data centers. A survey and a taxonomy are given in [7], while in [8] the focus is on the categorization of green computing performance indices: power metrics, thermal metrics, combined metrics, etc.

Virtualization is a common means to consolidate applications and in this way reduce power consumption [5][9][10]. The problem of optimally mapping VMs to servers can be reduced to the *bin packing problem* [1][11][12]: VMs can be treated as items of different size that must be assigned to the minimum number of servers (representing the bins) taken from a given set. This problem is known to be NP-hard, therefore heuristic approaches can only lead to sub-optimal solutions. Live migration of VMs among servers is adopted by the VMWare Distributed Power Management system, using lower and upper utilization thresholds to enact migration procedures. The heuristic approaches presented in [1] and in [12] use techniques derived, respectively, from the classical Best Fit Decreasing and the First Fit Decreasing algorithms. In both cases, the goal is to place each migrating VM on the server that minimizes the overall power consumption of the data center. An interesting study is presented in [13]. The paper proposes the Delayed Off strategy (the name derives from the fact that a server is turned off after been idle for some time), which is proved to be asymptotically optimal but only under some assumptions, for example stationary Poisson arrival process and homogeneous servers.

These approaches represent important steps ahead for the deployment of energy-efficient data centers, but still they share a couple of notable drawbacks. First, they use deterministic and centralized algorithms whose efficiency deteriorates as the size of the data center grows. Secondly, they may require the concurrent migration of many VMs, which causes considerable performance degradation during the reassignment process.

A novel approach for the consolidation of VMs, based on probabilistic trials, was presented in [6], and its mathematical

foundation was given in [14]. The solution has proved to be scalable thanks to its self-organizing nature, and ensures that VMs are relocated gradually using an asynchronous and smooth migration process. In most studies, including the last two, energy-efficiency strategies focus on CPU to obtain a consistent reduction of consumed power. The reason is that only CPU supports active low-power modes, whereas other hardware components can only be completely or partially switched off [2]. Nevertheless, important fractions of power are consumed by memory, disk, and power supplies [15]. Moreover, applications hosted by VMs often present complementary resource usage, so it may be profitably to let a server execute a mix of memory-bound and CPU-bound applications. When the assignment problem needs to consider multiple hardware resources, it can be formally modeled as a multidimensional bin packing problem, in which servers are represented by bins, and each resource (CPU, disk, memory) is a dimension of the bin [16]. This problem is clearly more difficult than the single-dimension bin packing problem, and centralized/deterministic solutions are hardly applicable even in small data centers.

The probabilistic approach presented here extends the one published in [6] to the case of multiple hardware resources. The avenue is to define assignment and migration functions for each resource type and let a server declare its availability to the accommodation of a VM only when Bernoulli trials are successful for every resource type. The second possibility, simpler and not analyzed here, is to execute a single Bernoulli trial for the most critical resource and use the other resources as constraints to be satisfied to enable the accommodation of the new or migrating applications.

To consolidate VMs, it is often necessary to migrate them between two servers of the same cluster, or between different clusters. This opportunity is favored by the recent trend in the development and implementation of Software-Defined Networks (SDN) [17], which extend the virtualization effort from applications to network facilities, also thanks to the definition of open standards like OpenFlow [18]. The SDN paradigm will allow the data center to be viewed and managed as a single large pool of computing and network resources, and VMs to be migrated between any two physical servers.

VI. Conclusion and Future Work

The paper focuses on the problem of making data centers and Cloud infrastructures more energy efficient. One of the most promising approaches consists in avoiding low utilization of servers, which implies not optimal use of energy, by consolidating the load on as few servers as possible. In particular, the paper deals with a recently proposed solution, namely ecoCloud, that, by being decentralized and probabilistic in nature, is highly scalable and allows smooth adaptation of the infrastructure to the actual traffic load. In this paper, ecoCloud is analyzed by extending the concept of server utilization from a scalar individual value to a set of values that represent the utilization of different kinds of resources inside the server: for example, utilization is applied separately to CPU and memory. An analytical model as well as experiments on a real data center, show that ecoCloud achieves high consolidation whatever combination of resource availability and resource demand is considered.

References

[1] A. Beloglazov and R. Buyya, "Energy efficient allocation of virtual machines in cloud data centers," in *10th IEEE/ACM Int. Symp. on Cluster Computing and the Grid, CCGrid 2010*, Melbourne, Australia, May 2010, pp. 577–578.

[2] L. A. Barroso and U. Hölzle, "The case for energy-proportional computing," *IEEE Computer*, vol. 40, no. 12, pp. 33–37, December 2007.

[3] G. Dasgupta, A. Sharma, A. Verma, A. Neogi, and R. Kothari, "Workload management for power efficiency in virtualized data centers," *Commun. ACM*, vol. 54, pp. 131–141, July 2011.

[4] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 1, pp. 68–73, 2009.

[5] M. Cardosa, M. R. Korupolu, and A. Singh, "Shares and utilities based power consolidation in virtualized server environments," in *Proceedings of the 11th IFIP/IEEE Integrated Network Management (IM 2009)*, Long Island, NY, USA, June 2009.

[6] C. Mastroianni, M. Meo, and G. Papuzzo, "Self-economy in cloud data centers: Statistical assignment and migration of virtual machines," in *17th International European Conference on Parallel and Distributed Computing, Euro-Par 2011*, vol. 6852. Bordeaux, France: Springer LNCS, September 2011, pp. 407–418.

[7] A. Beloglazov, R. Buyya, Y. C. Lee, and A. Y. Zomaya, "A taxonomy and survey of energy-efficient data centers and cloud computing systems," in *Advances in Computers*, M. Zelkowitz, Ed. Elsevier, 2011, pp. 47–111.

[8] L. Wang and S. U. Khan, "Review of performance metrics for green data centers: a taxonomy study," *The Journal of Supercomputing*, pp. 1–18, October 2011.

[9] E. Feller, C. Morin, and A. Esnault, "A case for fully decentralized dynamic vm consolidation in clouds," in *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, 2012, pp. 26–33.

[10] D. Huang, D. Yang, H. Zhang, and L. Wu, "Energy-aware virtual machine placement in data centers," in *Global Communications Conference (GLOBECOM), 2012 IEEE*, 2012, pp. 3243–3249.

[11] A. Verma, P. Ahuja, and A. Neogi, "pMapper: Power and migration cost aware application placement in virtualized systems," in *Middleware 2008, ACM/IFIP/USENIX 9th International Middleware Conference, Leuven, Belgium, December 1-5, 2008, Proceedings*, ser. Lecture Notes in Computer Science, V. Issarny and R. E. Schantz, Eds., vol. 5346. Springer, 2008, pp. 243–264.

[12] D. M. Quan, R. Basmadjian, H. de Meer, R. Lent, T. Mahmoodi, D. Sannelli, F. Mezza, L. Telesca, and C. Dupont, "Energy efficient resource allocation strategy for cloud data centres," in *26th Int. Symp. on Computer and Information Sciences, ISCIS 2011*, London, UK, September 2011, pp. 133–141.

[13] A. Gandhi, V. Gupta, M. Harchol-Balter, and M. A. Kozuch, "Optimality analysis of energy-performance trade-off for server farm management," *Perform. Eval.*, vol. 67, no. 11, November 2010.

[14] C. Mastroianni, M. Meo, and G. Papuzzo, "Analysis of a self-organizing algorithm for energy saving in data centers," in *Proc. of the 9th Workshop on High-Performance, Power-Aware Computing*, Boston (MA) USA, May 2013.

[15] L. Minas and B. Ellison, *Energy Efficiency for Information Technology: How to Reduce Power Consumption in Servers and Data Centers*. USA: Intel Press, 2009.

[16] S. Srikantaiah, A. Kansal, and F. Zhao, "Energy aware consolidation for cloud computing," in *USENIX Workshop on Power Aware Computing and Systems*, San Diego, CA, USA, December 2008.

[17] D. Drutskoy, E. Keller, and J. Rexford, "Scalable network virtualization in software-defined networks," *Internet Computing, IEEE*, vol. 17, no. 2, pp. 20–27, 2013.

[18] T. A. Limoncelli, "Openflow: A radical new idea in networking," *Queue*, vol. 10, no. 6, pp. 40:40–40:46, June 2012.