Probabilistic Consolidation of Virtual Machines in Self-Organizing Cloud Data Centers

Carlo Mastroianni, *Member*, *IEEE*, Michela Meo, *Member*, *IEEE*, and Giuseppe Papuzzo, *Member*, *IEEE*

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, for example, 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 ecoCloud, a self-organizing and adaptive 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.

Index Terms—Cloud computing, VM consolidation, data center, energy saving

1 Introduction

All main trends in information technology, for example, 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. In 2006, the energy consumed by IT infrastructures in the USA was about 61 billion kWh, corresponding to 1.5 percent of all the produced electricity, and 2 percent 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 in terms of energy consumption reduction, 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 1.1. However, much space remains for the optimization of the computing facilities themselves. It has been estimated that most of the time servers operate at 10-50 percent of their full capacity [2], [3]. This low utilization is also caused by the intrinsic variability of VMs' workload: the data center is planned to sustain the peaks of load, while for long periods of time (for example, during nights and weekends), the load is much lower [4], [5]. Since an active but idle server consumes between 50 and 70 percent of the power consumed when it is fully utilized [6], a large amount of energy is used even at low utilization.

The *virtualization* paradigm can be exploited to alleviate the problem, as many Virtual Machine (VM) instances 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 [7]. 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. The problem is complicated by two circumstances: 1) the assignment of VMs should take into account multiple server resources at the same time, for example, CPU and RAM, therefore it becomes a "multidimensional bin packing problem," much more difficult than the single dimension problem; 2) 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.

- *C. Mastroianni is with the ICAR-CNR, Via P. Bucci 41C, Rende, CS 87036, Italy. E-mail: mastroianni@icar.cnr.it.*
- *M. Meo is with the DET - Politecnico di Torino, corso Duca degli Abruzzi 24, Torino 10129, Italy. E-mail: michela.meo@polito.it.*
- *G. Papuzzo is with the Eco4Cloud, Piazza Vermicelli, Rende, CS 87036, Italy. E-mail: papuzzo@eco4cloud.com.*

Manuscript received 1 Apr. 2013; revised 30 Sept. 2013; accepted 28 Nov. 2013;published online 10 Dec. 2013.
Recommended for acceptance by Y. Cui.
For information on obtaining reprints of this article, please send e-mail to: tcc@computer.org, and reference IEEECS Log Number TCC-2013-04-0063.
Digital Object Identifier no. 10.1109/TCC.2013.17.

In [8], we presented ecoCloud, an approach for consolidating VMs on a single computing resource, i.e., the CPU. Here, the approach is extended to the multidimension 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. All this is done by demanding the key decisions to single servers, while the data center manager is only requested to properly combine such local decisions. The approach is partly inspired by the ant algorithms used first by Deneubourg et al. [9], and subsequently by a wide research community, to model the behavior of ant colonies and solve many complex distributed problems. The characteristics inherited by such algorithms make ecoCloud novel and different from other solutions. Among such characteristics: 1) the use of the swarm intelligence paradigm, which allows a complex problem to be solved by combining simple operations performed by many autonomous actors (the single servers in our case); 2) the use of probabilistic procedures, inspired by those that model the operations of real ants; and 3) the self-organizing behavior of the system, which ensures that the assignment of VMs to servers dynamically adapts to the varying workload.

To evaluate the performance of ecoCloud we use two complementary approaches. We first propose a fluid mathematical model that derives the evolution of the system with time by assuming that the involved variables are continuous. The model allows us to test ecoCloud in a wide range of scenarios by simply changing the value of some parameters. The second approach consists of experiments performed on real data centers. The two approaches complement each other: the analytical model introduces some simplifying assumptions but allows for an easy exploration of a wide range of scenarios; conversely, the real experiments do not suffer from assumptions but are, somehow, less representative. Both the approaches show that ecoCloud achieves very good consolidation, and smoothly adapts to possible changes in the system conditions. Finally, to compare the performance of ecoCloud with those of [1], that is, a reference approach, and to perform a scalability study, we use an ad hoc simulator.

The remainder of this paper is organized as follows: after a general description of the scenario and of performance metrics, given in Section 2, Section 3 defines and illustrates the assignment and migration procedures, generalized for the multiresource consolidation problem. Section 4 analyzes the assignment procedure through a mathematical model based on differential equations and shows that ecoCloud not only consolidates the load but also efficiently balances the available resources between compute-intensive and memory-intensive applications. Section 5 reports the results of the ecoCloud adoption in a real data center of a telecommunications company, extending the assessment to the migration procedure. Section 6 compares ecoCloud to one of the best deterministic algorithms devised recently, and Section 7 focuses on the scalability properties of ecoCloud. Section 8 describes related work and Section 9 concludes the paper.

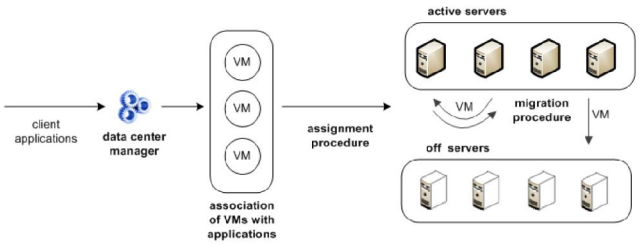

Fig. 1. Assignment and migration of VMs in a data center.

2 Scenario and Performance Metrics

The objective of ecoCloud is to dynamically map VMs to servers with the twofold objective of saving electrical costs—through the consolidation of VMs that allows some servers to enter low consuming sleep modes—and respecting the Service Level Agreements stipulated with users, especially concerning the expected quality of service. The scenario is pictured in Fig. 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, memory, storage space) and the type of operating system specified by the client. Then, the VM is assigned to one of the available servers through the *assignment procedure*.

The main idea underlying the whole approach is that it is up to the single servers to decide whether they should accept or reject a VM. These 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 is dynamic, that is, its demand for computational resources varies 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 resources utilization is too low, meaning that the server is highly underutilized, or when it is too high, possibly causing overload situations and service level agreement violations. The migration procedure consists of two steps: in the first step, a server requests the migration of a VM, on the basis of its CPU/RAM utilization. The purpose of the second step is to choose the server that will host the migrating VM, with a technique similar to the one used by the assignment procedure.

The performance of ecoCloud is assessed through the following metrics:

- *Resource utilization.* To foster consolidation and save power, a server should be either highly exploited or in a sleep mode. Analysis of CPU and RAM

utilization aims at checking if this objective is fulfilled.

- *Number of active servers.* VMs should be clustered into as few servers as possible. For example, if the overall load of the data center is equal to 30 percent of the total available capacity of servers, the number of active servers should be close to 30 percent of the overall number of servers.
- *Consumed power.* The ultimate objective is to save electrical power, so we compute the power consumed by the whole data center in different load conditions.
- *Frequency of migrations and server switches.* Any VM migration causes a slight performance degradation of the application hosted by the VM. The time needed to transfer the VM memory from the source server to the target server may vary from a few seconds up to two minutes in the worst cases [10], [11]. In this interval, the VM is active on the source server. During the actual handover of the VM, the VM experiences a downtime in the order of milliseconds. Analogously, the activation of an off server needs a startup time and additional power. Therefore, though migrations and switches are essential for VM consolidation and power reduction, it is important to limit their frequency. It is even more important to avoid massive migrations of VMs: the asynchronous and gradual migration of a number of VMs is much less detrimental than the concurrent migration of the same number of VMs; for example, concurrent migrations might overload the transmission bandwidth and, hence, increase the downtime duration.
- *SLA violations.* A violation of Service Level Agreements can happen when the workload of some VMs increases and the physical servers that host them become overloaded. Such events can be prevented by timely migrating some VMs to other less loaded servers. We measure the percentage of time in which the VMs allocated to a server demand more resources than what the server can provide. This metric, in accordance with recent studies [1], is used to assess the QoS level offered to users.

Several studies and experiments (e.g., [6] [12]) have found that an active server with very low CPU utilization consumes between 50 and 70 percent of the power that it consumes when fully utilized. Moreover, as the CPU utilization increases, the consumed power can be assumed, with the error below 10 percent, to increase linearly from the power corresponding to the idle state to the power corresponding to full utilization [13], [14]. Though some studies have derived more accurate nonlinear relations [15], such refinements have little practical utility to our purposes. Therefore, in analytical and simulation experiments presented in this study, the power consumed by a single server is expressed as

$$P(u) = P_{idle} + (P_{max} - P_{idle})u, \tag{1}$$

where P_{max} is the power consumed at maximum CPU utilization ($u = 1$) and P_{idle} is the power consumed when the server is active but idle ($u = 0$). In experiments on real data centers, the consumed power is directly monitored and measured.

3 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, 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 overutilized or underutilized on either of the two considered resources, CPU and RAM. In the case of overutilization, the rationale is to avoid overload situations that can penalize the quality of service perceived by users, while in the case of underutilization 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 (e.g., $T = 0.8$ means that the resource utilization cannot exceed 80 percent of the server capacity), the assignment function is equal to zero when $x > T$, otherwise it is defined as

$$f(x, p, T) = \frac{1}{M_p} x^p (T - x) \quad 0 \le x \le T, \tag{2}$$

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

$$M_p = \frac{p^p}{(p+1)^{(p+1)}} T^{(p+1)}. \tag{3}$$

Fig. 2 shows the graph of the single-resource assignment function (2) 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

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.

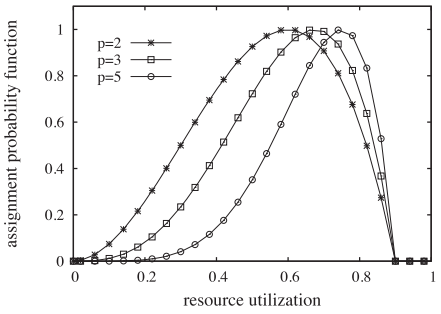

Fig. 2. Assignment probability function $f(x, p, T)$ for three different values of the parameter p, and T equal to 0.9.

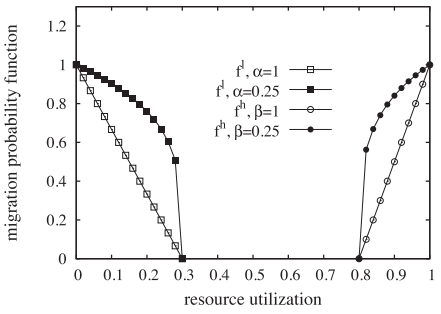

Fig. 3. Migration probability functions $f^l_{migrate}$ and $f^h_{migrate}$ (labeled as f^l and f^h) for two different values of the parameters α and β. In this example, the threshold T_l is set to 0.3, T_h is set to 0.8.

of p increases. The value of the function is zero or very low when the resource is overutilized or underutilized.

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 (2), where $x = u_s$ and $x = m_s$ are used for CPU and RAM, respectively. 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

$$f_s(u_s, m_s, p_u, p_m, T_u, T_m) = f(u_s, p_u, T_u) \cdot f(m_s, p_m, T_m). \quad (4)$$

The shape of the assignment functions, combined with the definition of function (4), 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 overutilized 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 4, but application workload changes with time. When some VMs terminate or reduce their demand for server resources, it may happen that the server becomes underutilized leading to 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, underutilization and overutilization 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 using the libraries provided by the virtualization infrastructure (e.g., VMWare or Hyper-V) 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,[3] 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, CPU or RAM, the migration probability functions are defined as follows:

$$f^l_{migrate} = (1 - x/T_l)^\alpha \quad (5)$$

$$f^h_{migrate} = \left(1 + \frac{x-1}{1-T_h}\right)^\beta. \quad (6)$$

The functions, whose graphs are shown in Fig. 3, are defined so as to trigger the migration of VMs when the utilization is below the threshold T_l or above the threshold T_h, respectively. 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.

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 overutilized resource (CPU or RAM) and considers the VMs for which 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

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

3. The overutilization of any resource is sufficient to trigger the migration procedure, because the overloaded resource becomes a bottleneck for the server. On the other hand, the under-utilization condition is only checked for the most utilized resource, which is the one that drives consolidation. For example, if RAM is the most utilized resource, some servers can have low values of CPU utilization, but this condition does not trigger migrations.

the threshold.[4] 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 helps to avoid multiple migrations of the same VM. 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 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.

Finally, the threshold values are generally given as an input by the data center administrator, possibly on the basis of a previous analysis on the variance of VMs workload. Shape parameters offer the data center administrator the chance to choose among different consolidation strategies (e.g., conservative, intermediate, aggressive): a more aggressive strategy allows more servers to be hibernated, but at the expense of more migrations. The choice of the desired strategy is made by tuning the values of the shape parameters. Since this analysis is out of the scope of the paper, in what follows the parameter values set in the experiments are those corresponding to the intermediate strategy.

4 Mathematical Analysis

This section is devoted to a mathematical 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, N_c the number of cores in each server and N_v the number of VMs that can be executed in each core. 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 $s = 0, \ldots, N_s - 1$. The utilization of both resources is a real number that changes by infinitesimal increments/decrements over the interval $[0, 1]$. A straightforward extension allows to model the evolution of a larger number of resources.

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,

$$u_s(t) = u_s^{(C)}(t) + u_s^{(M)}(t),$$
$$m_s(t) = m_s^{(C)}(t) + m_s^{(M)}(t).$$

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 (4). In the following, the acceptance probability is simply denoted as $f_s(t)$.

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

$$\frac{\partial u_s^{(C)}(t)}{\partial t} = -N_c\, N_v\, \mu\, u_s^{(C)}(t) + K\, \gamma_C\, \lambda^{(C)}(t)\, f_s(t), \tag{7}$$

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

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

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

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:

$$u_s^{(C)}(0), u_s^{(M)}(0), m_s^{(C)}(0), m_s^{(M)}(0) \quad s = 0, \ldots, N_s - 1. \tag{8}$$

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. The power consumed at maximum utilization P_{max} is set to 250 W, a typical value for the servers of a data center, while P_{idle} is set to 70 percent of P_{max}, i.e., 175 W. In the experiment, the VMs have nominal CPU frequency of 500 MHz. The average time the VM spends 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 $\lambda^{(C)}/\mu_T$ ($\lambda^{(M)}/\mu_T$).

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

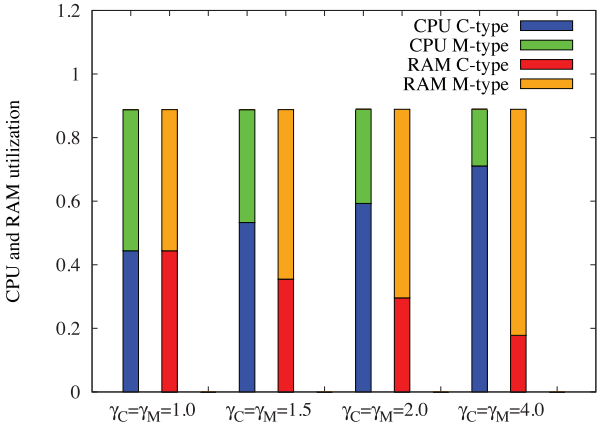

Fig. 4. CPU and RAM utilization of active servers, with $\rho_C = 0.4$, $\rho_M = 0.4$, and different values of γ_C and γ_M.

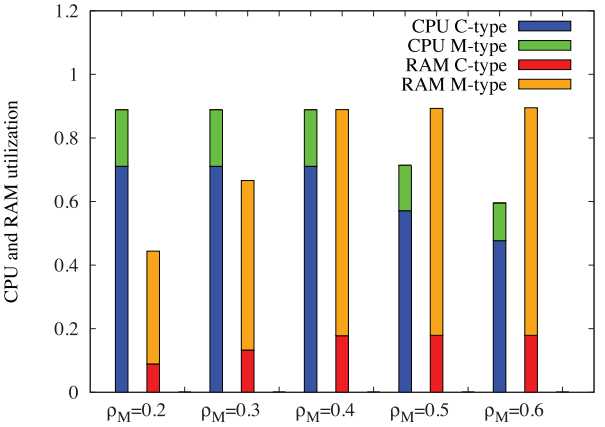

Fig. 5. CPU and RAM utilization of active servers, with different values of ρ_M, and $\rho_C = 0.4$.

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 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: $\rho_C = \rho_M = 0.4$.

The experiment started from a nonconsolidated 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 $T = 0.9$, and $p = 3$. 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 percent more CPU than M-type ones, and M-type applications need 50 percent 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 Fig. 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 nor the RAM becomes a bottleneck. For example, in the most imbalanced scenario (γ_C and γ_M equal to 4.0), about 71 percent of the CPU is assigned to C-type VMs while about 18 percent is given to M-type VMs, and the opposite occurs for memory. Both CPU and RAM are utilized up to the permitted threshold (90 percent) 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 percent 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 percent of the total CPU capacity of the servers, while the overall RAM load, ρ_M, is varied between 20 percent and 60 percent. 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 Fig. 5. Correspondingly, Figs. 6 and 7 report the number of active servers and the average value of consumed power.

When the overall memory load is lower than 0.4 (cases $\rho_M = 0.2$ and $\rho_M = 0.3$), 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

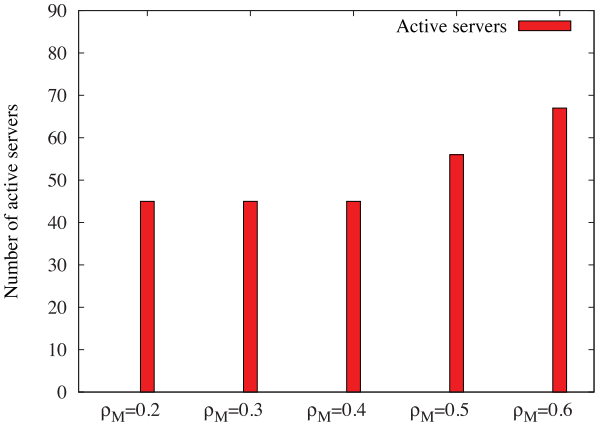

Fig. 6. Number of active servers with different values of ρ_M and $\rho_C = 0.4$.

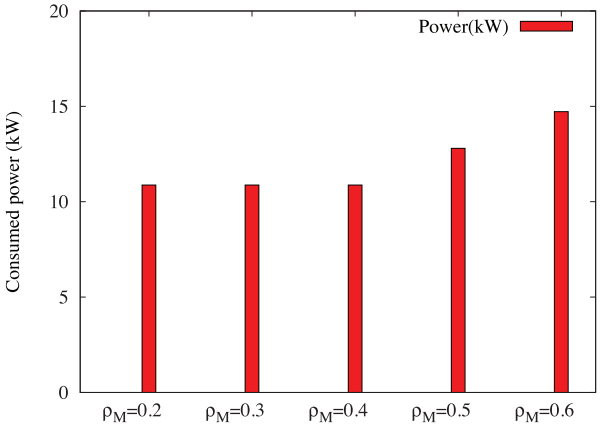

Fig. 7. Consumed power with different values of ρ_M and $\rho_C = 0.4$.

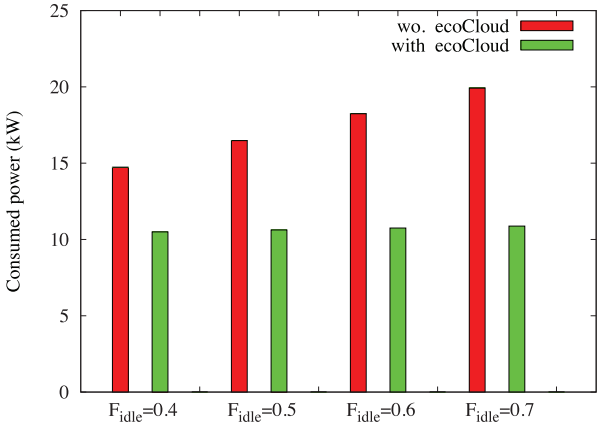

Fig. 8. Consumed power with different values of F_{idle}, before and after applying ecoCloud.

11 kW) are the same as in the case where CPU and RAM overall loads are comparable. On the other hand, when the most critical resource is the memory, as happens in the cases $\rho_M = 0.5$ and $\rho_M = 0.6$, 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 percent 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.

The benefit of consolidation depends on how much power is wasted due to server underutilization. In that respect, much research effort is devoted in trying to decrease the power consumed by idle servers, that is, the quantity P_{idle} in (1). This value can be expressed as a fraction F_{idle} of the power consumed at maximum utilization, $P_{idle} = F_{idle}P_{max}$. To analyze this aspect, we performed tests with different values of F_{idle} in a scenario with $\rho_C = 0.4$, $\rho_M = 0.4$, and a 80-20 imbalance between CPU-bound and RAM-bound VMs ($\gamma_C = \gamma_M = 4.0$). Fig. 8 reports the overall amount of power consumed in the data center before and after applying the ecoCloud algorithm. The advantage of consolidation decreases as servers become more power efficient. Nevertheless, the consumed power is reduced by about 30 percent even with servers that consume only 40 percent of the power when idle.

5 Experiments on a Real Data Center

In the previous section, we have shown through an analytical model, the effectiveness of ecoCloud in consolidating the load under various scenarios. However, the model relies on some necessary assumptions. To validate the model and prove that ecoCloud is effective in real scenarios, we report in this section the results of the experiments performed in May 2013 on a live data center owned by a major 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. All the servers have network adapters with bandwidth of 10 Gbps. The servers hosted 447 VMs which were assigned a number of virtual cores varying between 1 and 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 1,171 GHz and 2,334 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, 80 percent of the VMs, 358, were memory-bound, with an average usage of CPU and RAM of 0.345 GHz and 3.571 GB, respectively. The remaining 88 CPU-bound VMs had average values of CPU and RAM of 1.971 GHz and 1.633 GB, respectively. The M-type VMs contributed for the 49.44 percent of the overall CPU load and for the 92.15 percent of the overall memory load.

While the analytical study presented in Section 4 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 when the most utilized resource - the RAM in this case—goes below the low threshold T_l, set to 0.5. Values of α and β, in (5) and (6), were set to 0.25. The parameters of the assignment function were set as follows: $T = 0.8$ (this value was imposed by the data center administrator), $p = 3$.

Fig. 9 shows the number of active servers starting from the time at which ecoCloud is activated and for the following 7 days. Within the first day 11 servers, out of 28, are hibernated thanks to the workload consolidation. In the following days, the number of active servers is stabilized, but daily workload variations allow one or two servers to be hibernated during the night. Fig. 10 shows that the consumed power reduces thanks to consolidation, following the trend of the previous figure. Fig. 11 reports the number of high and low migrations performed during

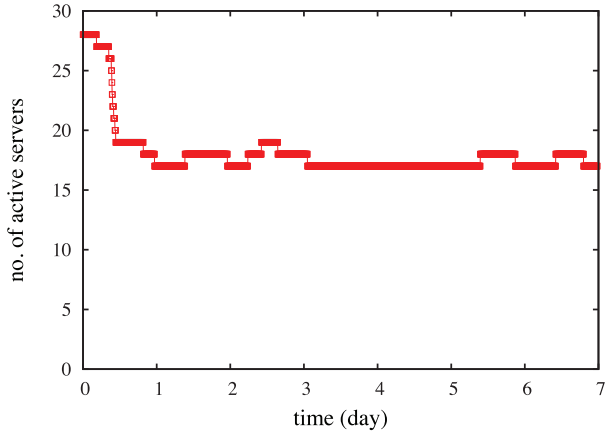

Fig. 9. Number of active servers after activation of ecoCloud.

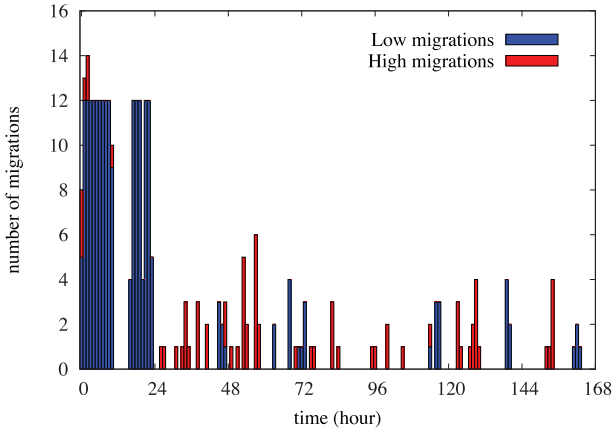

Fig. 11. Number of VM migrations after activation of ecoCloud.

each hour of the analyzed period on the whole data center. In the first day, migrations are mostly from low utilized servers, which are first unloaded and then hibernated. As the consolidation process proceeds, active servers tend to be well utilized and some high migrations are needed to prevent overload events, while low migrations allow to improve consolidation during the night. The number of migrations is definitely acceptable: after the first day, only a few migrations per day are performed.

The overhead induced by migrations was very low and did not cause a significant impact on the performance of running applications. None of the physical resources (CPU, memory, bandwidth) underwent an overload event, and the responsiveness of virtual machines was never deteriorated. More in detail, the typical effects on the source and target hosts, measured during the time needed to migrate the VM memory (from 30 to 90 seconds in our test) were the following: 1) an increase of CPU utilization up to 2 percent, never sufficient to saturate the CPU; 2) an extra bandwidth utilization equal to no more than 500 Mbps, i.e., only a fraction of the network adapter capacity (equal to 10 Gbps in our case), so that the available bandwidth was never saturated. As for the impact on the migrating VM, the downtime experienced in the final phase of the migration was always between 100 and 300 milliseconds, adding only a small delay to the normal response time of the application. These values are fully compatible with those recently published in a VMWare technical report [16]. It is useful to recall here that ecoCloud does not impact directly on the migration overhead, as migrations are executed by the virtualization platform, VMWare in this case. However, ecoCloud limits the number of migrations and, even more importantly, migrates the VMs gradually and asynchronously, in this way preventing the occurrence of bandwidth saturation and reducing the migration duration.

Figs. 12 and 13 offer a snapshot of the data center at the end of the seventh day of ecoCloud operation, when only 17 of 28 servers are 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 all active servers the RAM utilization is about 70 percent. 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 Fig. 13, which reports the numbers of VMs of the two types that run on each server. With the exceptions of servers 2 and 3, in which no C-type VM is running, 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 server capacities are not homogeneous, as detailed at the beginning of this section.

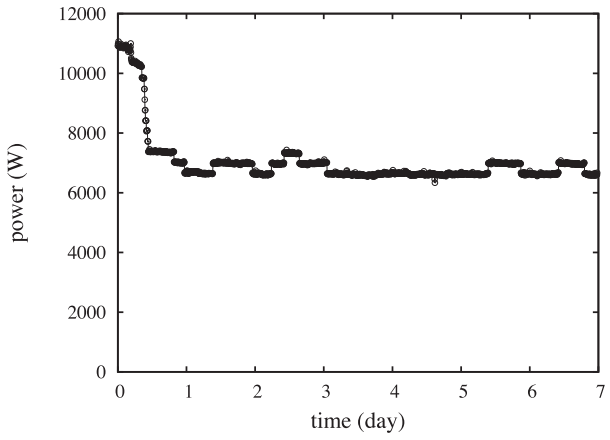

Fig. 10. Consumed power after activation of ecoCloud.

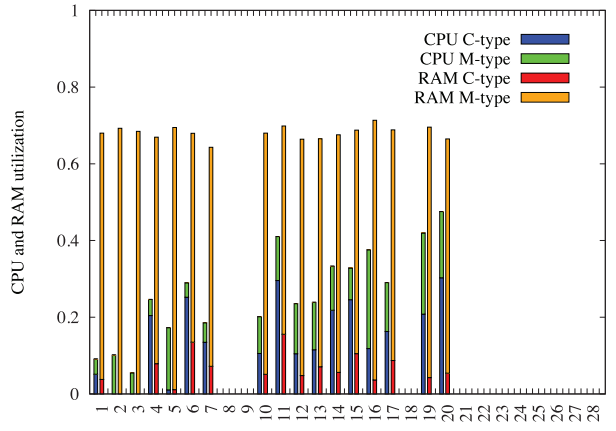

Fig. 12. RAM and CPU utilization on the 28 servers, separated for the C-type and M-type VMs. Values are taken at the end of the seventh day of operation.

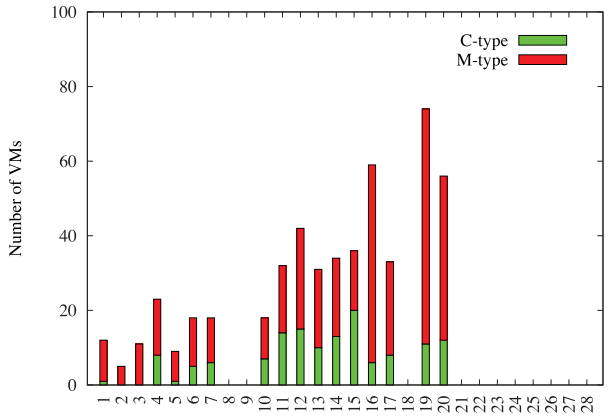

Fig. 13. Number of C-type and M-type VMs running on the 28 servers. Values are taken at the end of the seventh day of operation.

6 Comparison between ecoCloud and BFD

The problem of optimally mapping VMs to servers can be reduced to the *bin packing problem*. The analogy is indeed exploited in recent research [1], [17]. The problem is very complex - it was proved to be NP-hard—and currently adopted optimization algorithms are time consuming when applied to large data centers. A significant drawback of deterministic algorithms is that any efficient mapping may be valid only for a short period of time, due to the arrival/termination of VMs and to the dynamic nature of the workload. As a consequence, many simultaneous migrations of VMs may be needed to adapt the mapping to these variations.

A set of experiments were performed to compare ecoCloud to one of these algorithms. In particular, we implemented a variant of the classical Best Fit Decreasing algorithm described and analyzed in [1], referred to as BFD in the following. This choice was made because it was proved in [18] that the Best Fit Decreasing algorithm is the polynomial algorithm that gives the best results in terms of effectiveness. Its consolidation ratio is 11/9, which means that at most (11/9)MIN+1 servers are used, where MIN is the minimum theoretical number of servers. At each execution of BFD, VMs of overutilized and underutilized servers are collected, and then they are sorted in decreasing order of CPU utilization. Respecting this order, each VM is allocated to the server that provides the smallest increase of the power consumption caused by the allocation. A key parameter of BFD is the interval of time between two successive executions of the algorithm; therefore, we performed experiments with four different values of the interval: 1, 5, 15, and 60 minutes.

So far, we could not install ecoCloud in real data centers having more than 100 servers; thus, we used a home-made Java simulator fed with the logs of real VMs to compare ecoCloud and BFD in a data center with 400 servers. We used workload traces retrieved by the data of the CoMon project, a monitoring infrastructure for PlanetLab [19]. The traces represent the CPU utilization of 6,000 VMs, monitored in March/April 2012 and updated every 5 minutes. Since the CPU is the only resource considered in [1], we also consider this resource only for the experiments reported below.

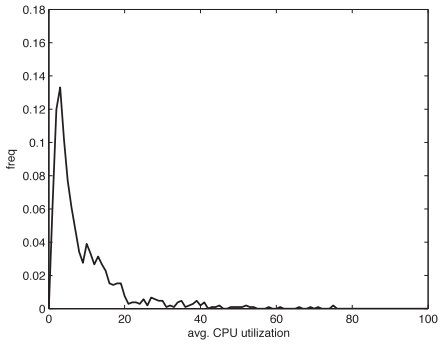

Fig. 14. Distribution of the average CPU utilization of the VMs.

A graphical characterization of the traces is provided in the following. Fig. 14 reports the distribution of the average CPU utilization of the VMs, measured as a percentage of the total CPU capacity of the hosting physical machine. The graph shows that the average CPU utilization is under 20 percent for most VMs, even though there are a few VMs with very high CPU requirements. It is clear that this kind of distribution leaves much room for clever consolidation algorithms, since in many cases tens of VMs can be executed on the same physical machine. We then collected, for all the VMs and for all the values of the CPU utilization, the difference—or deviation—between the punctual value and the average value of the same VM. The distribution of the deviations obtained in this way is reported in Fig. 15. Most values are close to zero, meaning that for most VMs CPU deviations are very small. Specifically, about 94 percent of the deviations are lower than 10, which means that if the average CPU utilization of a VM can be estimated—in most cases this is possible using historical data—and each VM is allocated as much CPU as this average value, only for 6 percent of the times the VM will exceed the allocated CPU by more than one tenth of the CPU capacity. Nevertheless, such deviations can still cause QoS violations, especially when multiple VMs increase their CPU demand at the same time.

The VM traces are picked randomly during the tests, in a number that depends on the desired overall load. We assigned the VMs to 400 servers, using the ecoCloud and BFD algorithms for assignment and migration of VMs. These servers are all equipped with 2-GHz cores. One third of the servers have four cores, one third have six cores and the remaining third have eight cores. The parameters of the

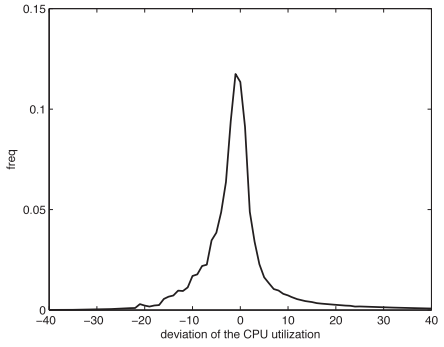

Fig. 15. Distribution of the deviation between the punctual CPU utilization and the average CPU utilization of the same VM.

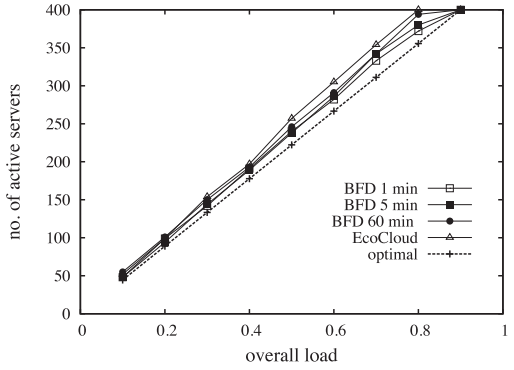

Fig. 16. Number of active servers versus load: comparison between ecoCloud and BFD. For BFD, the legend reports the time interval between two successive executions.

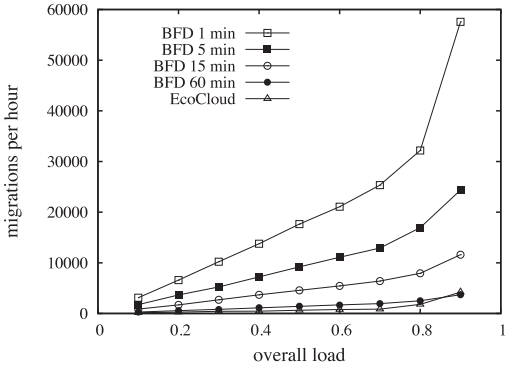

Fig. 18. Number of migrations per hour in the data center versus load: comparison between ecoCloud and BFD.

assignment and migration functions were set as follows: $T_a = 0.90$, $T_l = 0.50$, $T_h = 0.95$, $\alpha = 0.25$, and $\beta = 0.25$.

Fig. 16 reports the average number of active servers versus the overall load in ecoCloud and BFD. The curves are close to each other, and also close to the optimal value of the associated bin packing problem. ecoCloud requires a slightly larger number of active servers, mostly because of its behavior in descending load phases, during which the CPU utilization of servers is allowed to decrease by a certain amount before low migrations are triggered, to avoid migrations that are not strictly necessary. Similar observations can be done by analyzing the average consumed power of the two algorithms, shown in Fig. 17.

The slightly better consolidation degree of BFD, however, comes at a considerable cost in terms of the number of migrations and the probability of overload events. Fig. 18 shows that the number of migrations is much higher in BFD than in ecoCloud. For example, with load equal to 0.3, less than 400 migrations per hour are needed by ecoCloud, while about 10,000 migrations per hour are needed by BFD in the case that the time interval between two successive executions is set to 1 minute, as in [1]. This corresponds to more than 150 simultaneous migrations to be performed at each algorithm execution. If the BFD time interval is enlarged the frequency of migrations can be reduced, but the number of required simultaneous migrations increases: for example, about 750 simultaneous migrations are needed when the time interval is set to 60 minutes. Conversely, migrations are executed asynchronously with ecoCloud. Fig. 19 reports the percentage of time of CPU overload. The value of this index is remarkably lower in ecoCloud, due to its capacity of immediately reacting with high migrations each time the CPU utilization exceeds the upper threshold. The probability of overload in BFD comes as the combination of two contrasting phenomena: if the algorithm is executed frequently, the consolidation effort is stressed (cfr. Fig. 16), which brings the servers closer to their CPU limits and increases the overload probability. This is particularly evident when the overall load is high. When the time interval is larger the consolidation effort is lower, but VM workload variations are not controlled for a longer time, which can also be a cause of overload events. Thus, overload events are present at any load condition. With ecoCloud the index is hardly affected by the value of the overall load.

A comparison in terms of complexity in also interesting. The complexity of BFD [20] is $n \cdot m$, where n is the number of hosts and m is the number of VMs that need to be assigned or migrated. The complexity of ecoCloud is equal to the number of servers invited during the assignment/migration of a VM. This number is at most n, but in general it is much lower, because in large data centers it is sufficient to invite only a subset of servers, as discussed in the next section.

7 Results with Different Data Center Sizes

One of the most interesting and peculiar features of ecoCloud is its scalability, inherited from the probabilistic, self-organizing and partially distributed nature of the

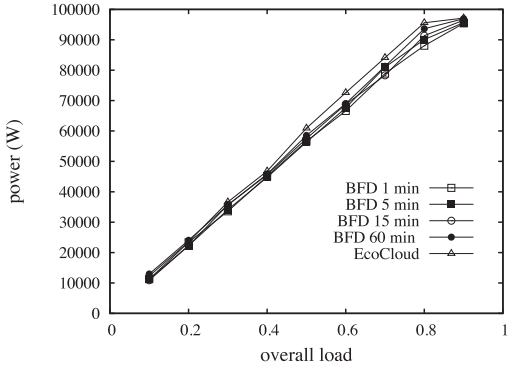

Fig. 17. Power consumed by the data center versus load: comparison between ecoCloud and BFD.

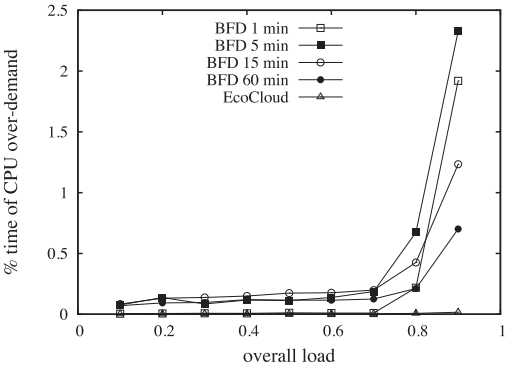

Fig. 19. Percentage of time of CPU overdemand versus load: comparison between ecoCloud and BFD.

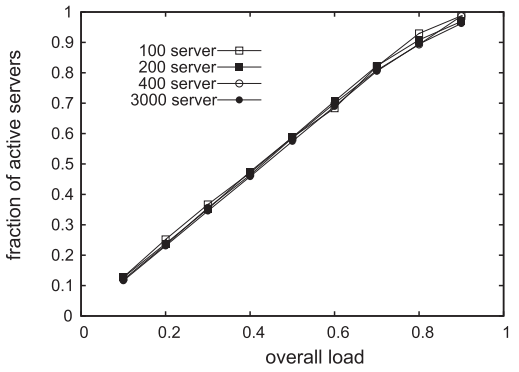

Fig. 20. Scalability test. Fraction of active servers in data centers with different size.

algorithm. Centralized and deterministic algorithms may be appropriate in data centers with a limited number of servers, but may become inefficient in large and very large data centers, due to the complexity of the problem and the need for the simultaneous migrations of a large number of VMs, as discussed in the previous section. Conversely, ecoCloud is particularly suited for large data centers. To understand why consolidation improves with the number of servers, it is useful to remember that a hibernated server is switched on when a new or migrating VM is rejected by all the active servers. In small systems, it can happen that all the servers—after the execution of Bernoulli trials—reject the VM even when some of them have enough spare CPU to accommodate the VM. The probability of this event becomes negligible in large data centers, where the invitation to accommodate a VM is forwarded to many servers: as a consequence, a server is activated only when strictly needed. This argument also motivates the fact that in large data centers it is not necessary to send invitations to all the servers, but it is sufficient to invite a subset of them. This has two beneficial consequences: 1) the traffic overhead can be limited; and 2) ecoCloud fits well with distributed and multisite data centers, since each invitation can be forwarded to the servers of a specific site, chosen randomly or on the basis of environmental parameters (the cost of energy in different sites, the external temperature, etc.).

To assess the ecoCloud scalability, we performed simulations with data centers of different size (100, 200, 400, and 3,000 servers), using the VM traces described in the previous section, and keeping the same proportion between the number of VMs and the number of physical servers. Fig. 20 reports the fraction of active servers versus the overall load and shows that this fraction is nearly independent on the system size. We also performed tests for a data center with 3,000 servers in which invitations are forwarded to varying numbers of servers. These tests confirm that there is no advantage to send invitations to more than about 100 servers. The good scalability is confirmed by the other performance metrics. For example, the frequency of migrations experienced by a single server is nearly independent from the system size.

8 Related Work

As the Cloud computing paradigm rapidly emerges, a notable amount of studies focus on algorithms and procedures that aim at improving the "green" characteristics of Cloud data centers. An interesting survey is given in [21], along with a useful taxonomy of examined methods. Another recent survey [22] focuses on the categorization of green computing performance metrics in data centers, such as power metrics, thermal metrics and extended performance metrics, i.e., multiple data center indicators. We are experiencing a turning point in this area. So far, most efforts have been devoted to the optimization of the physical infrastructure, commonly evaluated through the PUE index, and results have been notable, as this index is now as low as 1.08 in some modern data centers. Today, focus is switching to the efficiency of the IT infrastructure itself, and is testified by the definition of appropriate indices. Two examples are: 1) eBay has recently defined DSE, the Digital Service Efficiency index [23], which computes the useful work (in terms of transactions) performed per kWh; 2) Intel has proposed two new metrics [24]: IT-power usage effectiveness (ITUE), similar to PUE but "inside" the IT and total-power usage effectiveness (TUE), which combines the two for a total efficiency picture.

Consolidation is a powerful means to improve IT efficiency and in this way reduce power consumption [7], [25], [26]. Some approaches—for example, [27] and [13]—try to forecast the processing load and aim at determining the minimum number of servers that should be switched on to satisfy the demand, so as to reduce energy consumption and maximize data center revenues. However, even a correct setting of this number is only a part of the problem: algorithms are needed to decide how the VMs should be mapped to servers in a dynamic environment, and how live migration of VMs can be exploited to unload servers and switch them off when possible, or to avoid SLA violations.

The problem of optimally mapping VMs to servers can be reduced to the *bin packing problem* [17], [1], [28]. Unfortunately, this problem is known to be NP-hard, therefore heuristic approaches can only lead to suboptimal solutions. Live migration of VMs between servers is adopted by the VMWare Distributed Power Management system, using lower and upper utilization thresholds to enact migration procedures [29]. The heuristic approaches presented in [1] and in [28] use techniques derived, respectively, from the 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. The framework presented in [30] tackles the consolidation problem by exploiting the Constraint Programming paradigm. Rule-based constraints, for example concerning SLA negotiation, are managed by an optimizer that adopts a branching approach: the variables are considered in a priority descending order, and at each step one of the variables is set to the value that is supposed to guide the solver to a good solution. All these approaches represent important steps ahead for the deployment of green-aware 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. The second drawback is that mapping strategies

may require the concurrent migration of many VMs, which can cause considerable performance degradation during the reassignment process. Conversely, the approach presented here adopts a probabilistic approach, naturally scalable, and uses an asynchronous and smooth migration process, which ensures that VMs are relocated gradually.

An interesting study is presented in [31]. The paper confirms that the problem of energy saving in server farms is almost intractable and proposes the Delayed Off strategy (the name derives from the fact that a server is turned off only after a predetermined amount of time in which it has been idle), which is proved to be asymptotically optimal but only under some assumptions, for example stationary Poisson arrival process and homogeneous servers.

Bioinspired algorithms and protocols are emerging as a useful means to manage distributed systems, and Clouds are not an exception. Assignment and migration procedures presented here are partly inspired by the *pick* and *drop* operations performed by some species of ants that cluster items in their environment [9]. The pick and drop paradigm, though very simple and easy to implement, has already proved surprisingly powerful: for example, it is used to cluster and order resources in P2P networks, to facilitate their discovery [32]. Another ant-inspired mechanism is proposed in [33]: in this study, the data center is modeled as a P2P network, and ant-like agents explore the network to collect information that can later be used to migrate VMs and reduce power consumption. The V-MAN system, proposed in [34], is also based on the P2P paradigm. Here, a gossip protocol is used by servers to communicate their state to each other, and migrate VMs from servers with low load to servers with higher load, with the aim of switching off the former and save energy. The approach is promising but needs more assessment, as it makes the unrealistic assumption that all VMs are identical. In our opinion, the main problem of pure P2P approaches is that the complete absence of centralized control can be seen as an obstacle by the data center administrator. With ecoCloud, despite the fact that servers can autonomously decide whether or not to migrate or accept a VM, final decisions are still granted to the central manager of the data center, which ensures a better control of the operations.

Since the mapping of VMs to servers is essentially an optimization problem, evolutionary and genetic algorithms can also represent a valid solution. In [35], a genetic algorithm is used to optimize the assignment of VMs, and minimize the number of active servers. The main limitations of this kind of approach are the need of a strong centralized control and the difficulties in the setting of key parameters, such as the population size and the crossover and mutation rates.

In most studies, CPU is the main component on which energy-efficiency strategies focus to obtain a consistent reduction of consumed power. The reason is that, among hardware components, only CPU supports active low-power modes, whereas other components can only be completely or partially switched off. Server CPUs can consume less than 30 percent of their peak power in low-activity modes, leading to dynamic power range of more than 70 percent of peak power [2]. Dynamic power ranges of other components are much narrower, or even negligible. Nevertheless, important fractions of power are consumed by memory, disk, and power supplies [36]. Applications hosted by VMs often present complementary resource usage, so it may be profitably to let a server execute, for example, a mix of memory-bound and CPU-bound applications. In [37], the mapping of VMs to servers was modeled as a multidimensional bin packing problem, in which servers are represented by bins, and each resource (CPU, disk, memory, and network) was considered as a dimension of the bin. While formally interesting, this problem is even more difficult than the classical bin packing problem, therefore it is hardly applicable in large data centers. The algorithm presented in [38] is based on the first-fit approximation for the bin packing problem. The algorithm was devised for the single resource problem, but tips are given about the extension to multiple resources. In [39] the multiresource problem is tackled by using an LP formulation that gives higher priority to virtual machines with more stable workload. ReCon [40] is a tool that analyzes the resource consumption data of various applications, discovers applications which can be consolidated, and subsequently generates static or dynamic consolidation recommendations. Only CPU utilization is considered, the complete extension to the multiresources problem is left to future research. The Entropy resource manager presented in [41] performs dynamic consolidation based on constraint programming, where constraints are defined both on CPU and on RAM utilization. When compared to these interesting approaches, our algorithm differentiates for its ability to adaptively consolidate the workload without using any complex centralized algorithm and balance the assignment of CPU- and RAM-intensive applications on each server, which helps to optimize the use of resources.

Owing to the increased size of data centers, several big companies are adopting a multi-data center infrastructure. This allows companies to balance the load, improve the quality of service and, when sites are distributed over multiple regions or States, save energy by exploiting the different energy costs at different locations and time zones. Distributed solutions for data centers are analyzed in [42] and [43]. The solution presented here can be easily tailored to these environments, by splitting the assignment and migration processes into two phases. In the first phase, the system decides on which specific data center an application should be assigned or migrated, on the basis of management, load balancing and energy-efficiency criteria, not differently from other distributed environments. In the second phase, the probabilistic approach is used to decide on which specific server of the selected data center the application should be executed.

9 Conclusion

This paper tackles the issue of energy-related costs in data centers and Cloud infrastructures, which are the largest contributor to the overall cost of operating such environments. The aim is to consolidate the Virtual Machines on as few physical servers as possible and switch the other servers off, so as to minimize power consumption and carbon emissions while ensuring a good level of the QoS

experienced by users. With ecoCloud, the approach proposed in the paper, the mapping of Virtual Machines is based on Bernoulli trials through which single servers decide, on the basis of the local information, whether or not they are available to execute an application. The self-organizing and probabilistic nature of the approach makes ecoCloud particularly efficient in large data centers. This is a notable advantage with respect to other fully deterministic algorithms, which inevitably encounter significant difficulties when the size of the data center grows, since the problem of the optimal assignment of Virtual Machines to servers is known to be very complex.

Mathematical analysis and experiments performed in a real data center in operation show that the adopted techniques succeed in the objectives of reducing power consumption, avoiding overload events that could cause SLA violations, limiting the number of VM migrations and server switches, and balancing CPU-bound and memory-bound applications. Simulation experiments prove that these achievements can be obtained for any system load and system size and that ecoCloud performance is competitive with other approaches based on more traditional algorithms.

Acknowledgments

The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement no. 257740 (Network of Excellence TREND).

References

[1] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing," *Future Generation Computer Systems,* vol. 28, no. 5, pp. 755-768, 2012.

[2] L.A. Barroso and U. Hölzle, "The Case for Energy-Proportional Computing," *IEEE Computer,* vol. 40, no. 12, pp. 33-37, Dec. 2007.

[3] G. Dasgupta, A. Sharma, A. Verma, A. Neogi, and R. Kothari, "Workload Management for Power Efficiency in Virtualized Data Centers," *Comm. ACM,* vol. 54, pp. 131-141, July 2011.

[4] L. Hosman and B. Baikie, "Solar-Powered Cloud Computing Datacenters," *IT Professional,* vol. 15, no. 2, pp. 15-21, 2013.

[5] M. Aggar, "Developers, Developers, Developers: Engaging the Missing Link in It Resource Efficiency," technical report, The Green Grid, Mar. 2013.

[6] A. Greenberg, J. Hamilton, D.A. Maltz, and P. Patel, "The Cost of a Cloud: Research Problems in Data Center Networks," *Proc. ACM SIGCOMM Computer Comm. Rev.,* vol. 39, no. 1, pp. 68-73, 2009.

[7] M. Cardosa, M.R. Korupolu, and A. Singh, "Shares and Utilities Based Power Consolidation in Virtualized Server Environments," *Proc. 11th IFIP/IEEE Integrated Network Management (IM '09),* June 2009.

[8] C. Mastroianni, M. Meo, and G. Papuzzo, "Self-Economy in Cloud Data Centers: Statistical Assignment and Migration of Virtual Machines," *Proc. 17th Int'l European Conf. Parallel Processing (Euro-Par '11),* pp. 407-418, Sept. 2011.

[9] J.L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain, and L. Chrétien, "The Dynamics of Collective Sorting: Robot-Like Ants and Ant-Like Robots," *Proc. First Int'l Conf. Simulation of Adaptive Behavior on from Animals to Animats,* pp. 356-363, 1990.

[10] T. Hirofuchi, H. Ogawa, H. Nakada, S. Itoh, and S. Sekiguchi, "A Live Storage Migration Mechanism over Wan for Relocatable Virtual Machine Services on Clouds," *Proc. Ninth IEEE/ACM Int'l Symp. Cluster Computing and the Grid (CCGrid '09),* pp. 460-465, May 2009.

[11] H. Liu, C.-Z. Xu, H. Jin, J. Gong, and X. Liao, "Performance and Energy Modeling for Live Migration of Virtual Machines," *Proc. 20th Int'l Symp. High Performance Distributed Computing (HPDC '11),* pp. 171-182, June 2011.

[12] A. Khosravi, S. Garg, and R. Buyya, "Energy and Carbon-Efficient Placement of Virtual Machines in Distributed Cloud Data Centers," *Proc. 19th Int'l Conf. Parallel Processing (Euro-Par '13),* 2013.

[13] M. Mazzucco, D. Dyachuk, and R. Deters, "Maximizing Cloud Providers" Revenues via Energy Aware Allocation Policies," *Proc. 10th IEEE/ACM Int'l Symp. Cluster Computing and the Grid (CCGrid '10),* pp. 131-138, May 2010.

[14] S. Rivoire, P. Ranganathan, and C. Kozyrakis, "A Comparison of High-Level Full-System Power Models," *Proc. Conf. Power Aware Computing and Systems (HotPower '08),* Dec. 2008.

[15] X. Fan, W.-D. Weber, and L.A. Barroso, "Power Provisioning for a Warehouse-Sized Computer," *Proc. 34th Ann. Int'l Symp. Computer Architecture (ISCA '07),* pp. 13-23, June 2007.

[16] VMWare, "VMware vSphere 5.1 vMotion Architecture, Performance and Best Practices," technical report, VMWare tech. papers, http://www.vmware.com/resources/techresources/10305, Aug. 2012.

[17] A. Verma, P. Ahuja, and A. Neogi, "pMapper: Power and Migration Cost Aware Application Placement in Virtualized Systems," *Proc. ACM/IFIP/USENIX Ninth Int'l Middleware Conf. (Middleware '08),* pp. 243-264, 2008.

[18] M. Yue, "A Simple Proof of the Inequality FFD (L) $\leq$ 11/9 OPT (L) + 1, for All L for the FFD Bin-Packing Algorithm," *Acta Mathematicae Applicatae Sinica,* vol. 7, no. 4, pp. 321-331, 1991.

[19] K. Park and V.S. Pai, "CoMon: A Mostly-Scalable Monitoring System for Planetlab," *ACM SIGOPS Operating Systems Rev.,* vol. 40, pp. 65-74, Jan. 2006.

[20] A. Beloglazov and R. Buyya, "Energy Efficient Allocation of Virtual Machines in Cloud Data Centers," *Proc. 10th IEEE/ACM Int'l Symp. Cluster Computing and the Grid (CCGrid '10),* pp. 577-578, May 2010.

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

[22] L. Wang and S.U. Khan, "Review of Performance Metrics for Green Data Centers: A Taxonomy Study," *The J. Supercomputing,* pp. 1-18, Oct. 2011.

[23] N. Greene et al., "White Paper on Digital Service Efficiency," technical report, eBay Inc., http://dse.ebay.com/sites/default/files/eBay-DSE-130305.pdf, Mar. 2013.

[24] M. Patterson, S. Poole, C.-H. Hsu, D. Maxwell, W. Tschudi, H. Coles, D. Martinez, and N. Bates, "TUE, a New Energy-Efficiency Metric Applied at ORNL's Jaguar," *Proc. Int'l Supercomputing Conf.,* 2013.

[25] P. Graubner, M. Schmidt, and B. Freisleben, "Energy-Efficient Virtual Machine Consolidation," *IT Professional,* vol. 15, no. 2, pp. 28-34, 2013.

[26] K. Schröder and W. Nebel, "Behavioral Model for Cloud Aware Load and Power Management," *Proc. Int'l Workshop Hot Topics in Cloud Services (HotTopiCS '13),* 2013.

[27] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam, "Managing Server Energy and Operational Costs in Hosting Centers," *ACM SIGMETRICS Performance Evaluation Rev.,* vol. 33, no. 1, pp. 303-314, June 2005.

[28] 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," *Proc. 26th Int'l Symp. Computer and Information Sciences (ISCIS '11),* pp. 133-141, Sept. 2011.

[29] A. Gulati, A. Holler, M. Ji, G. Shanmuganathan, C. Waldspurger, and X. Zhu, "Vmware Distributed Resource Management: Design, Implementation, and Lessons Learned," *VMware Technical J.,* https://labs.vmware.com/academic/publications/gulati-vmtj-spring2012, Spring 2012.

[30] K. Dhyani, S. Gualandi, and P. Cremonesi, "A Constraint Programming Approach for the Service Consolidation Problem," *Proc. Int'l Conf. Integration of AI and OR Techniques in Constraint Programming (CPAIOR '10),* pp. 97-101, June 2010.

[31] A. Gandhi, V. Gupta, M. Harchol-Balter, and M.A. Kozuch, "Optimality Analysis of Energy-Performance Trade-Off for Server Farm Management," *Performance Evaluation,* vol. 67, no. 11, pp. 1155-1171, Nov. 2010.

[32] A. Forestiero, C. Mastroianni, and G. Spezzano, "So-Grid: A Self-Organizing Grid Featuring Bio-Inspired Algorithms," *ACM Trans. Autonomous and Adaptive Systems,* vol. 3, no. 2, article 5, May 2008.
[33] D. Barbagallo, E.Di Nitto, D.J. Dubois, and R. Mirandola, "A Bio-Inspired Algorithm for Energy Optimization in a Self-Organizing Data Center," *Proc. First Int'l Conf. Self-Organizing Architectures (SOAR '09),* pp. 127-151, Sept. 2009.
[34] M. Marzolla, O. Babaoglu, and F. Panzieri, "Server Consolidation in Clouds through Gossiping," *Proc. IEEE 12th Int'l Symp. a World of Wireless, Mobile and Multimedia Networks,* pp. 1-6, June 2011.
[35] H. Mi, H. Wang, G. Yin, Y. Zhou, D. Shi, and L. Yuan, "Online Self-Reconfiguration with Performance Guarantee for Energy-Efficient Large-Scale Cloud Computing Data Centers," *Proc. IEEE Int'l Conf. Services Computing (SCC '10),* pp. 514-521, July 2010.
[36] L. Minas and B. Ellison, *Energy Efficiency for Information Technology: How to Reduce Power Consumption in Servers and Data Centers.* Intel Press, 2009.
[37] S. Srikantaiah, A. Kansal, and F. Zhao, "Energy Aware Consolidation for Cloud Computing," *Proc. USENIX Workshop Power Aware Computing and Systems,* Dec. 2008.
[38] N. Bobroff, A. Kochut, and K. Beaty, "Dynamic Placement of Virtual Machines for Managing SLA Violations," *Proc. 10th IFIP/IEEE Int'l Symp. Integrated Network Management (IM '07),* 2007.
[39] T. Ferreto, M. Netto, R. Calheiros, and C. De Rose, "Server Consolidation with Migration Control for Virtualized Data Centers," *Future Generation Computer Systems,* vol. 27, no. 8, pp. 1027-1034, 2011.
[40] S. Mehta and A. Neogi, "ReCon: A Tool to Recommend Dynamic Server Consolidation in Multi-Cluster Data Centers," *Proc. IEEE Network Operations and Management Symp. (NOMS),* 2008.
[41] F. Hermenier, X. Lorca, J.-M. Menaud, G. Muller, and J. Lawall, "Entropy: A Consolidation Manager for Clusters," *Proc. ACM SIGPLAN/SIGOPS Int'l Conf. Virtual Execution Environments (VEE),* 2009.
[42] K. Le, O. Bilgir, R. Bianchini, M. Martonosi, and T.D. Nguyen, "Managing the Cost, Energy Consumption, and Carbon Footprint of Internet Services," *ACM SIGMETRICS Performance Evaluation Rev.,* vol. 38, pp. 357-358, June 2010.
[43] S.K. Garg, C.S. Yeo, A. Anandasivam, and R. Buyya, "Environment-Conscious Scheduling of HPC Applications on Distributed Cloud-Oriented Data Centers," *J. Parallel and Distributed Computing,* vol. 71, pp. 732-749, June 2011.

Carlo Mastroianni received the Laurea degree and the PhD degree in computer engineering from the University of Calabria, Italy, in 1995 and 1999, respectively. He has been a researcher at the Institute of High Performance Computing and Networking of the Italian National Research Council (ICAR-CNR) in Cosenza, Italy, since 2002. Previously, he worked at the Computer Department of the Prime Minister Office in Rome. He coauthored more than 100 papers published in international journals, including *IEEE/ACM Transactions on Networking*, *IEEE Transactions on Evolutionary Computation* and *ACM Transactions on Autonomous and Adaptive Systems*, and conference proceedings. He edited special issues for the journals *Future Generation Computer Systems*, *Journal of Network and Computer Applications*, *Multiagent* and *Grid Systems*. His research interest include Cloud and Grid computing, P2P, bioinspired algorithms, and multiagent systems. He is a cofounder of the Eco4Cloud company (www.eco4cloud.com). He is a member of the IEEE.

Michela Meo received the Laurea degree in electronics engineering in 1993, and the PhD degree in electronic and telecommunications engineering in 1997, both from the Politecnico di Torino, Italy. Since November 1999, she is an assistant professor at Politecnico di Torino. She coauthored more than 150 papers, about 50 of which are in international journals. She edited six special issues of international journals, including *ACM Monet*, *Performance Evaluation*, and *Journal and Computer Networks*. She was program cochair of two editions of ACM MSWiM, general chair of another edition of ACM MSWiM, program cochair of the IEEE QoS-IP, IEEE MoVeNet 2007, and IEEE ISCC 2009, and she was in the program committee of about 50 international conferences, including SIGMETRICS, INFOCOM, ICC, and GLOBECOM. Her research interests include the field of performance evaluation and modeling, traffic classification and characterization, P2P, and green networking. She is a member of the IEEE.

Giuseppe Papuzzo received the Laurea degree in computer engineering from the University of Calabria, Cosenza, Italy, in 2004. Since 2004, he collaborates with the Institute of High Performance Computing and Networks of the Italian National Research Council (ICAR-CNR) in Cosenza, Italy. He coauthored scientific papers published in international conferences and journals like *Future Generation Computing Systems* and *Transactions on Computational Systems Biology*. His research interests include workflow management, P2P networks, Grid and Cloud Computing, and data streaming. He is cofounder of the Eco4Cloud company (www.eco4cloud.com). He is a member of the IEEE.