2013 IEEE 27th International Symposium on Parallel & Distributed Processing Workshops and PhD Forum

Analysis of a Self-Organizing Algorithm for Energy Saving in 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*—The Cloud computing paradigm allows users to satisfy their increasing need for on-demand and remote computational services. These services are provided by data centers that often consume a huge amount of electrical power. Recently the** ecoCloud **algorithm has been proposed as a solution for saving energy by consolidating Virtual Machines on as few servers as possible, so as to hibernate the remaining servers and save energy. The** ecoCloud **approach founds on probabilistic processes: mapping and migration of VMs are driven by Bernoulli trials whose success probability depends on the utilization of single servers. These processes are self-organizing and decentralized, which makes them particularly efficient in large data centers. While in previous work the performance evaluation of** ecoCloud **was based on artificial traces, in this paper, a mathematical analysis is presented along with simulations fed with logs of real VMs. Results show that efficiency is very close to the theoretical minimum and comparable to that of one of the best centralized algorithms devised so far; in addition,** ecoCloud **notably reduces the frequency of events, such as VM migrations and server switches, that can deteriorate the quality of service.**

I. Introduction

The availability of powerful data centers and high bandwidth connections have expedited the success of the Cloud computing paradigm, which is making on-demand computing a common practice for companies and scientific communities. The main advantage of the Cloud paradigm is that a company does not need to operate its own data center, with all the related costs and administration burdens, but can access to CPU power, storage facilities, software packages on the basis of its needs. The choice of using external Cloud facilities frees the company from the burden of exactly estimating the required amount of resources, with the risk of under- or over-provisioning of resources due to inevitable load variations, and lead to notable money savings [1][2].

One of the main issues related to the success of Cloud computing is that the ever growing number of large data centers is causing a notable increase of electrical power consumed by hardware facilities and cooling systems. This increases the cost of computation itself and affects the carbon footprint of data centers, thus aggravating, on the global scale, the problem of global warming. 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 were expected to double by 2011 [3].

A major reason for this huge amount of consumed power is the inefficiency of data centers, which are often under-utilized: it has been estimated that only 20-30% of the total server capacity is used on average [4][5]. Unfortunately, power consumption is not proportional to the server utilization: an active but idle server consumes approximately 65-70% of the power consumed when it is fully utilized [6]. The *virtualization* paradigm can be exploited to alleviate the problem: user processes are not assigned directly to servers, but are first associated to Virtual Machine (VM) instances, which in turn are run by servers. The use of virtualization facilitates the *consolidation* of applications: VMs may be clustered on a reduced number of servers [7], and other servers can be unloaded and put in some low consuming sleep modes.

In [8] we presented ecoCloud, an approach partly inspired by the basic ant algorithms used by Deneubourg et al. [9] to model the phenomenon of larval clustering in ant colonies. In the ecoCloud case, the approach aims at clustering VMs in as few servers as possible, 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, which makes the management of the data center easier and more scalable.

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 specified by the client. Then, the VM is assigned to one of the available servers through the *assignment procedure*.

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

978-0-7695-4979-8/13 $26.00 © 2013 IEEE
DOI 10.1109/IPDPSW.2013.184

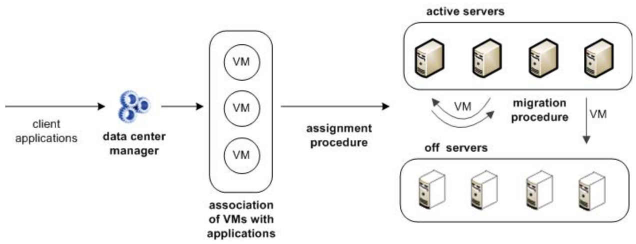

Fig. 1. Assignment and migration of VMs in a data center.

locally – in particular, information on the local CPU 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 CPU utilization is too low, meaning that the server is under-utilized, or when it is too high, possibly causing overload situations and quality of service 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 utilization. The purpose of the second step is to choose the server that will host the migrating VM.

In [8], the ecoCloud algorithm was evaluated through simulation experiments driven by artificial traces. In this paper, we analyze the behavior of ecoCloud during 48 hours of normal operation. This is done through simulation experiments driven by the traces of real Virtual Machines for a data center of 400 servers. The behavior of ecoCloud, specifically that of the assignment procedure, is then analyzed through a mathematical model based on differential equations, and the results are compared to those of tailored simulation experiments. Results prove that VMs are consolidated efficiently, while the frequency and duration of overload events is kept low.

The rest of the paper is organized as follows: section II summarizes the assignment and migrations procedures, which have been refined with respect to the version published in [8]. Section III shows the results of simulation experiments for a real scenario, and Section IV describes the analytical model and compares results to those of ad hoc simulation experiments. 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 *assignment procedure* is performed when a client asks the data center to execute a new application, and is defined as follows. 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 (as discussed in the related work section it was proven to be NP-Hard), 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 decides whether to declare its availability to accept the VM considering that: (i) a server with low CPU utilization should reject any new VM and should try to get rid of the currently running VMs in order to be put in a sleep mode and save energy; (ii) a server with very high CPU utilization should reject new VMs, to avoid overload situations that can penalize the quality of service perceived by users; (iii) a server with intermediate CPU utilization should accept new VMs to favor the consolidation.

The server decision is taken performing a Bernoulli trial, whose success probability depends on the *local* CPU utilization, u, (valued between 0 and 1), and on the maximum allowed value of utilization, T_a. The *assignment probabilistic function*, f_a, is equal to zero when $u > T_a$, otherwise it is defined as:

$$f_a(u) = \frac{1}{M_p} u^p (T_a - u) \quad 0 \leq u \leq T_a \qquad (1)$$

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

$$M_p = \frac{p^p}{(p+1)^{(p+1)}} T_a^{(p+1)} \qquad (2)$$

Figure 2 shows the function for some values of the parameter p, and $T_a = 0.9$. The function definition ensures that the CPU utilization does not exceed the threshold T_a (no further VMs can be assigned when u reaches this threshold) and that VMs are preferably assigned to servers having intermediate or moderately high workload, thus respecting the three guidelines mentioned above. The value of u at which the function reaches its maximum - that is, the value at which assignment attempts succeed with the highest probability - is $p/(p+1)T_a$, which increases and approaches T_a as the value of p increases. Therefore, the value of p can be used to modulate the shape of the function and tune the consolidation effort.

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 all the servers have CPU utilization very close to the threshold[2].

[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.

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

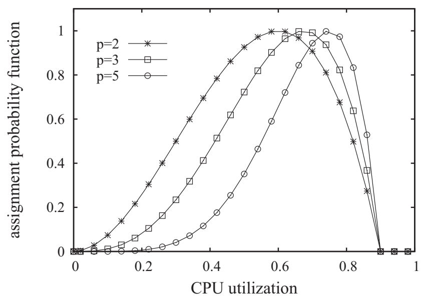

Fig. 2. Assignment probability function $f_a(u)$ for three different values of the parameter p. The value of the threshold T_a is set to 0.9.

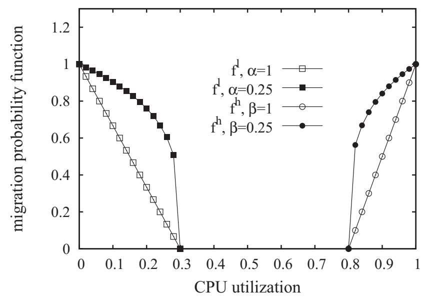

Fig. 3. Migration probability functions $f^l_{migrate}$ and $f^h_{migrate}$ (labeled as f^l and f^h) for two different values of the parameters α and β. In this example, the threshold T_l is set to 0.3, T_h is set to 0.8.

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: if this situation occurs frequently, the company should consider the acquisition of new servers.

Even after an efficient mapping of applications, the VMs running in a server may terminate or may reduce their demand for CPU, causing the under-utilization of server resources and the waste of energy. Moreover, when the VMs increase their CPU 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 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 $f^h_{migrate}$. In either case, it performs a Bernoulli trial and decides whether or not to request the migration of one of the local VMs. The migration probability functions are defined as follows:

$$f^l_{migrate} = (1 - u/T_l)^\alpha \tag{3}$$

$$f^h_{migrate} = \left(1 + \frac{u-1}{1-T_h}\right)^\beta \tag{4}$$

The functions, whose graphs are shown in Figure 3, are defined so as to trigger the migration of VMs when the CPU 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. When the utilization is in between the thresholds, migrations are inhibited. 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.

If the Bernoulli trial is positive, the server considers the local VMs whose CPU utilization 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 CPU utilization to go below the threshold[3]. 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_a of the assignment function is set to 0.9 times the CPU 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. Trace-driven Simulations

To understand how ecoCloud works and to evaluate its performance, in this section we present the results obtained by trace-driven simulations. A home-made Java simulator was fed with the logs of real VMs collected in an operative data center composed of 400 servers. We used workload traces retrieved by the data of the CoMon project, a monitoring infrastructure for PlanetLab [10]. The traces represent the CPU utilization of 6,000 VMs, monitored in March-April 2012 and updated every 5 minutes. A graphical characterization of the traces is provided in the following. Figure 4 reports the distribution of the average CPU utilization of the VMs, measured as a

[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.

percentage of the total CPU capacity of the hosting physical machine. The graph shows that the average CPU utilization is under 20% 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.

To consider the variability of the VM load in time, we then collected, for all the VMs, the difference – or deviation – between the punctual value of utilization (in percentage with respect to the total CPU capacity) and the average value of the VM utilization. The distribution of the deviations obtained in this way is reported in Figure 5. Most values are close to zero, meaning that for most VMs CPU deviations are very small. Specifically, about 94% 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 6% of the times the VM will exceed the allocated CPU by more than 10% of the CPU capacity. Nevertheless, such deviations can still cause QoS violations, especially when multiple VMs increase their CPU demand at the same time.

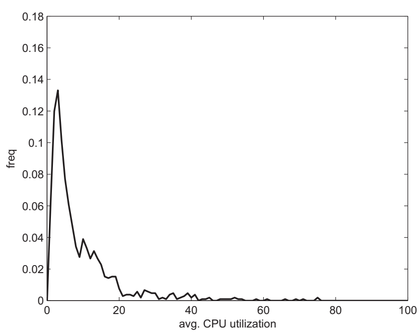

Fig. 4. Distribution of the average CPU utilization of the VMs.

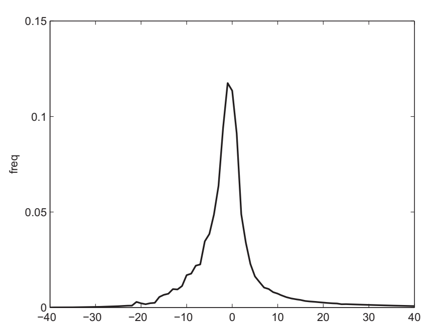

Fig. 5. Distribution of the deviation between the punctual CPU utilization and the average CPU utilization of the same VM.

To assess the behavior of ecoCloud in daily conditions, we report performance metrics for two consecutive days, starting from the midnight of the first day. This allows us to see what happens to the system when the overall load follows the normal daily pattern, with increasing load in the morning and decreasing load in the evening. All the metrics are computed every 30 minutes. The VMs are distributed to 400 physical servers, using the ecoCloud algorithms for assignment and migration of VMs. These servers are all equipped with 2 GHz cores. One third of the servers have 4 cores, one third have 6 cores and the remaining third have 8 cores. The parameters of the assignment and migration functions are set as follows: T_a=0.90, p=3, T_l=0.50, T_h=0.95, α=0.25 and β=0.25. Figure 6 shows the CPU utilization of the servers, along with the overall load as a reference (blach dots). When the load increases, some idle servers are activated to allocate the new VMs. When the load of active servers decreases, low migrations allow some of the servers to be hibernated, and the load consolidates to a lower number of servers.

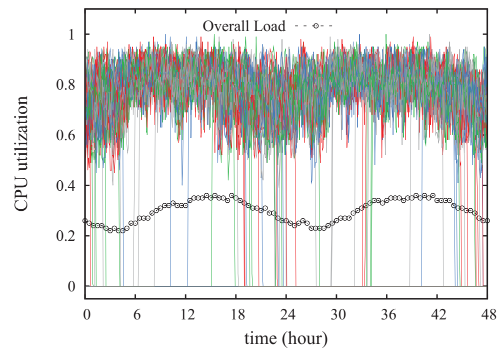

Fig. 6. CPU utilization of 400 servers during two consecutive days. The trend of the overall load is shown as a reference.

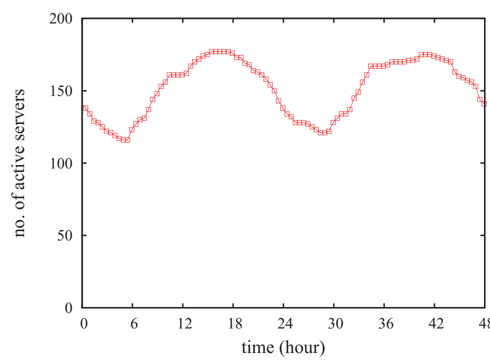

Fig. 7. Number of active servers during two consecutive days.

The capacity of consolidating the load is confirmed by the trend of the number of active servers, shown in Figure 7. This number is nearly proportional to the overall load reported in Figure 6, which means that servers are switched on and consume power when needed, and are hibernated whenever the decreasing load makes it possible. Figure 8 shows the power consumed by the data center. Again, the consumed energy follows the trend of the overall load and does not present any peak or sudden variation, which confirms that the asynchronous and self-organizing nature of ecoCloud allows the power to be continuously and smoothly adapted to the varying load conditions. Figure 9 focuses on the frequency of high and low migrations. Not surprisingly, high migrations are more frequent when the load increases, because the CPU utilization of active servers tends to approach the threshold, while low migrations are triggered when the load decreases and some servers can be emptied and hibernated. The total frequency of migrations is always lower than 200 migrations per hour for the entire data center, which corresponds to a migration every 2 hour for a single server and a migration every 3 days for a single VM (since each server runs about 40 VMs on average). These rates are easily sustainable, also considering that migrations are performed asynchronously. Figure 10 reports the frequency of server switches in the data center. Switches are performed only when needed: in ascending phases there are only activations and no hibernations and the opposite happens in descending phases.

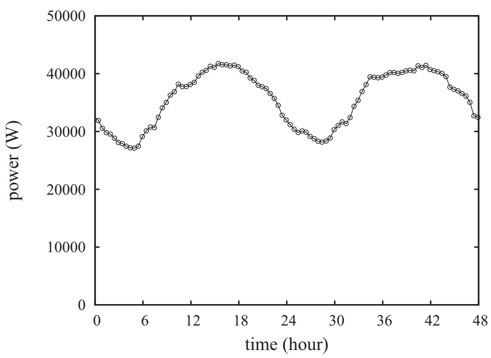

Fig. 8. Power consumed by the data center.

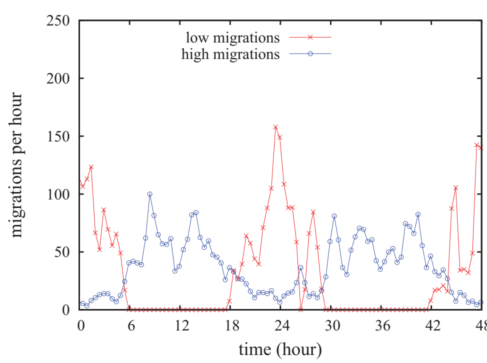

Fig. 9. Number of low and high migrations per hour in the data center.

Finally, Figure 11 reports the fraction of time, in percentage, in which the CPU demanded by a VM cannot be completely granted by the hosting server because of an overload event. This index is very small, never higher than 0.02%, thanks to the possibility of requesting the migration of a VM as soon as a server approaches the full utilization of its CPU. In the presence of overload events, the response of the server may be to forcedly decrease the CPU usage of all the VMs or only of those that have low priority. Due to the very limited frequency

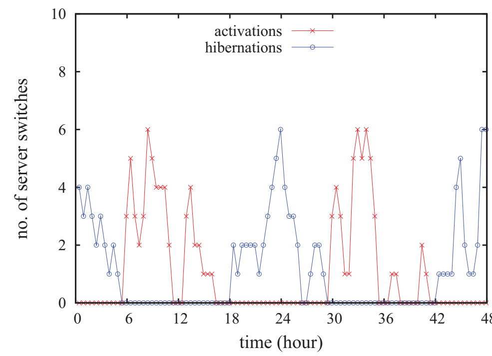

Fig. 10. Number of server switches per hour in the data center.

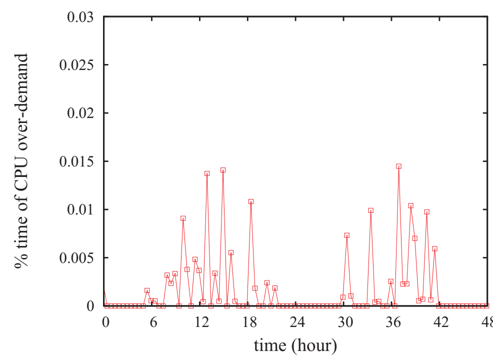

Fig. 11. Fraction of time of CPU over-demand in percentage.

of these events, normal operations can be rapidly restored. In these experiments it was found that, thanks to migration procedures, more than 98% of violations are shorter than 30 seconds, and even in those time intervals the VMs are granted no less than 98% of the demanded CPU.

A sensitivity study on the parameters of the migration probability function was also performed, results are not reported here for the sake of brevity. The analysis shows that the threshold T_h must be higher than the assignment threshold T_a, otherwise VM migrations would not allow the CPU to be exploited to the desired extent; the value of T_l should be chosen so as to ensure that servers are never utilized under 40% of their capacity; the values of α and β can be tuned depending on the willingness to accept that a server is under- or over-utilized for a short interval of time.

IV. 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, s the index of a generic server, $s = 0, \cdots, N_s - 1$, and let N_c be the number of cores in each server. The equations model the evolution of the server utilization with time, $u_s(t)$. Two assumptions are made about $u_s(t)$. First, the utilization is proportional to the number of Virtual Machines (VMs) that at time t are assigned to server s; this means that the load of a

VM is assumed to be constant. Second, the utilization is a real number that changes by infinitesimal increments/decrements over $[0, 1]$. This assumption implicitly means that the number of VMs is represented by a real number and changes by infinitesimal intervals.

The rate at which VMs arrive at the data center is denoted by $\lambda(t)$, while $\mu(t)$ is the service rate of each server core. Given the fluid model assumption described above, the VM arrival process is a continuous process that makes it arrive, in a time period Δt, a number of VMs equal to $\lambda(t)\Delta t$ (a real number). Similarly, for the departure process according to which VMs leave the system, in a time period Δt, $\mu(t)\Delta t$ VMs leave the system.

The set of differential equations is the following:

$$\frac{\partial u_s(t)}{\partial t} = -N_c\mu(t)u_s(t) + \lambda(t)A_s(t)f_a(u_s(t)) \qquad (5)$$
$$s = 0, \cdots, N_s - 1$$

where $A_s(t)$ is the fraction of VMs that, at time t, is assigned to server s. In particular, $A_s(t)$ depends on the assignment function f_a that, in turn, depends on the server utilization. For server s, $A_s(t)$ can be computed by considering that a VM is assigned to s with probability $1/k$ if $k-1$ other servers beyond s gave their availability to serve the VM. Letting $P_s^{(k)}(t)$ be the probability that k servers are available to accept the VM, we have:

$$A_s(t) = \frac{1}{1 - \prod_{i=0}^{N_s-1}(1 - f_a(u_i(t)))} \sum_{k=0}^{N_s-2} \frac{1}{k+1} P_s^{(k)}(t) \qquad (6)$$

where the first term is a normalization factor. $P_k(t)$ is derived by the combinatorial computation that any subset of k servers accept the VM, while the remaining servers do not accept,

$$P_s^{(0)}(t) = \prod_{i=0, i\neq s}^{N_s-1} (1 - f_a(u_i(t))) \qquad (7)$$

$$P_s^{(1)}(t) = \sum_{j=0, j\neq s}^{N_s-1} f_a(u_j(t)) \prod_{\substack{i=0 \\ i\neq s \\ i\neq j}}^{N_s} (1 - f_a(u_i(t))) \qquad (8)$$

$$\vdots$$

$$P_s^{(N_s-2)}(t) = \prod_{i=0, i\neq s}^{N_s-1} f_a(u_i(t)) \qquad (9)$$

As can be noticed from (5), the utilization of a server s increases according to the rate at which VMs arrive and are assigned to s. The utilization decreases in proportion to the VM leave rate $\mu(t)$ and to the number of VMs, i.e., to the utilization.

The equations can be solved with the initial conditions:

$$u_s(0) \quad s = 0, \cdots, N_s - 1 \qquad (10)$$

Since the computation of the terms $A_s(t)$ becomes costly as the number of servers increases, we also propose a simplified model. The results of this model proved to be very close to those of the exact model described so far. Specifically, we approximate the expression of $A_s(t)$ by assuming that the probability that a VM is assigned to s is proportional to the acceptance probability $f_a(u_s(t))$. The equations become,

$$\frac{\partial u_s(t)}{\partial t} = -N_c\mu(t)u_s(t) + \lambda(t)\frac{f_a(u_s(t))}{\sum_{i=0}^{N_s-1}(f_a(u_i(t)))} \qquad (11)$$
$$s = 0, \cdots, N_s - 1$$

A simple experiment was performed to compare analytical and simulation results. Since the equations cannot model migration events, we performed a simulation experiment in which migrations are inhibited, and analyze the behavior of ecoCloud when it is only driven by the assignment procedure. In this experiment we set the number of servers N_s to 100, each having $N_c = 6$ cores with CPU frequency of 2 GHz. These servers were loaded with 1,500 VMs randomly chosen among the 6,000 described in the previous section. The simulation experiment started from a non consolidated scenario, in which most servers have CPU load between 10% and 30%. The parameters of the assignment function f_a were set as follows: maximum utilization threshold T_a=0.9, p=3.

Figure 12 shows that at the activation of the assignment procedure, at time 0, some servers experience a decrease of their CPU utilization, until they are hibernated, while other servers keep on accommodating VMs until their CPU utilization approaches the threshold T_a. The probabilistic nature of the procedure is the reason why servers react in a different fashion and with different speed. After about 6 hours the system has reached a steady condition, with all the servers either hibernated or working nearly at their maximum allowed utilization. The simulation was initiated at midnight, in a phase of low load. The figure shows that, starting at about 8:30 am, more servers are activated to accommodate the increasing load. At the end of the experiment, 45 servers take all the load of the data centers and 55 are hibernated.

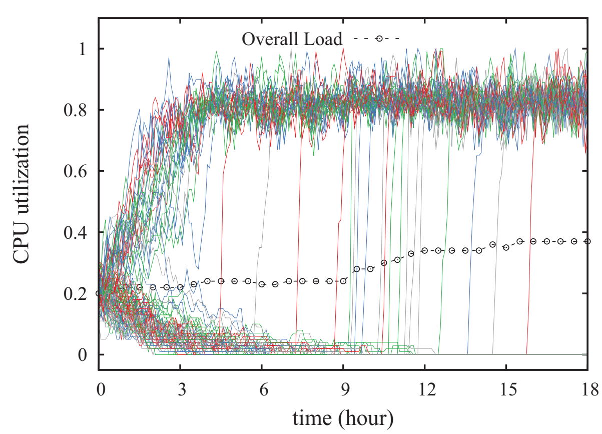

Fig. 12. CPU utilization of 100 servers, obtained with simulation.

From the traces we computed the values of $\lambda(t)$ and $\mu(t)$ and put the same values in the approximate differential

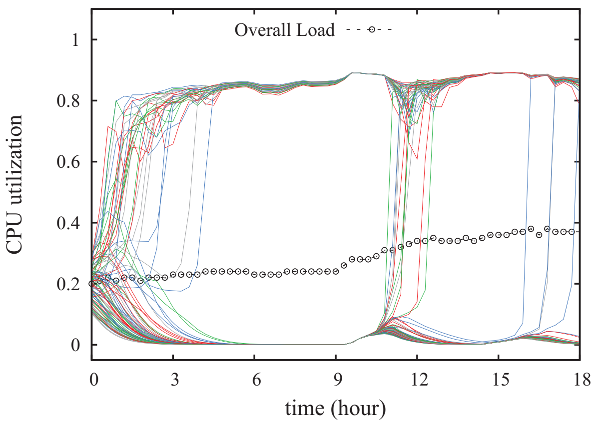

Fig. 13. CPU utilization of 100 servers, obtained with the analytical model.

equations (11), and also the initial conditions (10) were set as in the simulation experiment. Figure 13 reports the CPU utilization of the 100 servers.

The behavior is very similar to the one observed in Figure 12, yet some interesting differences can be appreciated. First, the load consolidates on 43 servers instead of 45. The variations of CPU utilization, caused by the arrivals and departures of VMs, are continuous when modeled with the differential equations and discrete when simulated. Finally, it may be observed that, when solving the equations, several servers are activated simultaneously to accommodate the increasing load. Then some servers are again switched off because their CPU values does not reach a critical mass, while others continue to accept VMs until they approach the threshold. This effect is avoided in the simulation experiment by imposing that a newly activated server always responds positively to new assignment requests for a limited interval of time, set to 30 minutes.

V. Related Work

A notable amount of studies focus on algorithms and procedures that aim at improving the "green" characteristics of Cloud data centers. A survey and a useful taxonomy are given in [11], while in [12] focus is on the categorization of green computing performance metrics: power metrics, thermal metrics, combined metrics, etc. Virtualization is a common means to consolidate applications on as few servers as possible and in this way reduce power consumption [7]. Some approaches - e.g., [13] and [14] - 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* [15][3][16]. Unfortunately, this problem is known to be NP-hard, therefore heuristic approaches can only lead to sub-optimal 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. The heuristic approaches presented in [3] and in [16] 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. 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. The ecoCloud approach presented here is naturally scalable, thanks to its probabilistic nature, and uses an asynchronous and smooth migration process, which ensures that VMs are relocated gradually.

Bio-inspired 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, in order to facilitate their discovery [17]. Another ant-inspired mechanism is proposed in [18]: 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. 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.

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% of their peak power in low-activity modes, leading to dynamic power range of more than 70% of peak power [4]. Dynamic power ranges of other components are much narrower, or even negligible. Nevertheless, recent experiments have shown that important fractions of power are consumed by memory, disk, and power supplies [19]. Moreover, 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. The probabilistic approach presented here can be extended to consider multiple hardware resources, following at least two possible avenues. The first one is to define assignment and migration functions for each resource type. A server executes a Bernoulli trial for each resource, and declares its availability to execute an application only when all trials are successful. The second possibility 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. These strategies are currently under investigation.

VI. Conclusion and Future Work

This paper analyzes the performances of ecoCloud, a novel approach that tackles the issue of energy-related costs in data centers and Cloud infrastructures. The aim is to consolidate Virtual Machines on the minimum number of physical servers, so as to minimize power consumption and carbon emissions. The decentralized and probabilistic nature of the approach makes ecoCloud particularly efficient in large data centers. The scalable behavior derives from the fact that important decisions on VM assignments and migrations are taken by single servers exclusively on the basis of local information. Analysis is performed through a mathematical model based on fluid-like differential equations and on simulation experiments driven by real log data of servers and VMs. Experiments show that the approach succeeds in the combined objective of reducing power consumption and avoiding overload events that could deteriorate the quality of service. Moreover, only a very limited number of VM migrations and server switches are needed. An important avenue for future work is the extension of the approach to take assignment decisions on the basis not only of the CPU utilization but also of other important parameters, such as memory, disk and bandwidth consumption.

References

[1] Y. Chen and R. Sion, “To cloud or not to cloud?: musings on costs and viability,” in *Proc. of the 2nd ACM Symposium on Cloud Computing, SOCC ’11*, Cascais, Portugal, October 2011, pp. 29:1–29:7.

[2] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, “Above the clouds: A berkeley view of cloud computing,” EECS Department, University of California, Berkeley, Tech. Rep., February 2009.

[3] 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.

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

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[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,” in *First International Conference on Simulation of Adaptive Behavior on From Animals to Animats*. Cambridge, MA, USA: MIT Press, 1990, pp. 356–363.

[10] K. Park and V. S. Pai, “Comon: a mostly-scalable monitoring system for planetlab,” *SIGOPS Oper. Syst. Rev.*, vol. 40, January 2006.

[11] 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.

[12] 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.

[13] Y. Chen, A. Das, W. Qin, A. Sivasubramaniam, Q. Wang, and N. Gautam, “Managing server energy and operational costs in hosting centers,” *SIGMETRICS Perform. Eval. Rev.*, vol. 33, no. 1, pp. 303–314, June 2005.

[14] M. Mazzucco, D. Dyachuk, and R. Deters, “Maximizing cloud providers’ revenues via energy aware allocation policies,” in *10th IEEE/ACM Int. Symp. on Cluster Computing and the Grid, CCGrid 2010*, Melbourne, Australia, May 2010, pp. 131–138.

[15] 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.

[16] 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.

[17] A. Forestiero, C. Mastroianni, and G. Spezzano, “So-grid: A self-organizing grid featuring bio-inspired algorithms,” *ACM Transactions on Autonomous and Adaptive Systems*, vol. 3, no. 2, May 2008.

[18] D. Barbagallo, E. Di Nitto, D. J. Dubois, and R. Mirandola, “A bio-inspired algorithm for energy optimization in a self-organizing data center,” in *Proc. of the 1st international conference on Self-organizing architectures, SOAR ’09*, Cambridge, UK, September 2010, pp. 127–151.

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