Hierarchical Approach for Efficient Workload Management in Geo-Distributed Data Centers

Agostino Forestiero, Carlo Mastroianni, *Member, IEEE*, Michela Meo, *Member, IEEE*, Giuseppe Papuzzo, and Mehdi Sheikhalishahi

***Abstract*—Geographically distributed data centers (DCs) offer promising business opportunities to both big companies that own several sites and multi-owner inter-cloud infrastructures. In these scenarios, workload management is a particularly challenging task, since the autonomy of single DCs should be preserved while global objectives, such as cost reduction and load balance, should be achieved. In this paper, a hierarchical approach for workload management in geographically distributed DCs is presented. The proposed solution is composed of two algorithms devoted to workload assignment and migration. Both algorithms are based on the computation of a simple function that represents the cost of running some workload in the different sites of the distributed DC. The framework requires a very limited exchange of state information among the sites and preserves the autonomy of single DCs and, at the same time, allows for an integrated management of heterogeneous platforms. Performance is analyzed for a specific infrastructure composed of four DCs, with two goals: 1) load balance and 2) energy cost reduction. Results show that the proposed approach smoothly adapts the workload distribution to variations of energy cost and load, while achieving the desired combination of management objectives.**

***Index Terms*—Cloud computing, geographical data centers, energy saving, cost saving, load balancing, VM migrations.**

I. Introduction

THE EVER increasing demand for computing resources has led companies and resource providers to build private data centers (DCs), or to offload applications and services to the DCs owned by a Cloud company. Due to this process, the number and scale of data centers are rapidly increasing. It is estimated that data center electricity consumption is projected to increase to roughly 140 billion kilowatt-hours annually by 2020, corresponding to about 50 large power plants, with annual carbon emissions of nearly 150 million metric tons. The financial impact for the DC management is also huge, since a DC spends between 30% to 50% of its operational expenditure in electricity: the expected figure for the sector in 2020 is $13 billion per year of electricity bills.[1]

The efficient utilization of resources in the data centers is therefore essential to reduce costs, energy consumption, carbon emissions and also to ensure that the quality of service experienced by users is adequate and adherent to the stipulated Service Level Agreements. Through the allocation of multiple Virtual Machines (VMs) on the same physical server, the virtualization technology helps to increase the efficiency of DCs. A good level of efficiency must be guaranteed also in geographically distributed DCs, whose adoption is rapidly increasing. Major cloud service providers, such as Amazon, Google, and Microsoft, are deploying distributed DCs to match the increasing demand for resilient and low-latency cloud services, or to interconnect heterogeneous DCs owned by different companies, in the so-called "Inter-Cloud" scenario. In this scenario, the dynamic allocation and migration of workload among DCs has become also an opportunity to reduce costs, moving the workload where the energy is cheaper/cleaner and/or cooling costs are lower, according to what is called the "follow the moon" paradigm. Inter-site migration is enabled by the availability of a high network capacity achievable thanks to physical improvements and logical/functional enhancements (e.g., the adoption of Software Defined Networks).

While workload assignment and migration can be very effective for cost reduction, the associated decision processes are made particularly complex by the time-variability of electricity cost, and by the workload variability both within single sites and across the whole infrastructure. Workload management is typically solved as an optimization problem, often in a centralized way. This approach has three main implications: (i) poor scalability, due to the large number of parameters and servers; (ii) poor ability to adapt to changing conditions, as massive migrations of VMs may be needed to match a new decision on the workload distribution; (iii) limitation to the autonomy of the sites, which are often required to share the same strategies and algorithms. The need for autonomous management is self-explanatory in multi-owned DCs, and is crucial even within a single-owner infrastructure, for example

Manuscript received January 28, 2016; revised May 14, 2016; accepted August 4, 2016. Date of publication August 31, 2016; date of current version March 17, 2017. The associate editor coordinating the review of this paper and approving it for publication was J. Elmirghani. *(Corresponding author: Michela Meo.)*

A. Forestiero, C. Mastroianni, and G. Papuzzo are with the Institute for High Performance Computing and Networking, ICAR-CNR, 87036 Rende, Italy, and also with Eco4Cloud srl, 87036 Rende, Italy (e-mail: forestiero@icar.cnr.it; mastroianni@icar.cnr.it; papuzzo@icar.cnr.it).

M. Meo is with the Department of Electronics and Communications, Politecnico di Torino, 10129 Torino, Italy (e-mail: michela.meo@polito.it).

M. Sheikhalishahi is with CREATE-NET, 38123 Trento, Italy (e-mail: mehdi.sheikhalishahi@create-net.org).

Digital Object Identifier 10.1109/TGCN.2016.2603586

[1]Updated information can be found on the Web portal of the U.S. National Resources Defense Council, http://www.nrdc.org/energy/data-center-efficiency-assessment.asp.

in the case that one or several sites are hosted by co-located multi-tenant facilities.

To tackle these challenging issues, this paper proposes EcoMultiCloud, a hierarchical framework for the efficient distribution of the workload on a multi-site platform. The framework allows for an integrated and homogeneous management of heterogeneous platforms but at the same time preserves the autonomy of single sites. It also gives the data center administrators the opportunity of specifying the business goals that are mostly relevant for the specific scenario – minimization of energy costs, load balancing, reduction of carbon emission, etc. – and their relative importance, as well as constraints on the minimum values of the objectives. Another key feature is the self-organizing and adaptive nature of the approach: VM migrations are performed asynchronously, when and where needed, and their rate is tunable by administrators.

The framework is composed of two layers: at the *lower layer*, each site adopts its own strategy to distribute and consolidate the workload internally. At the *upper layer*, a set of algorithms – shared by all the sites – are used to evaluate the behavior of single sites and distribute the workload among them, both at the time that new applications/VMs are assigned and when some workload migration from one site to another is deemed appropriate. At each site a Data Center Manager (DCM) periodically sends to other sites' DCMs a number of parameters that summarize the *state* of the site: possible parameters include the overall utilization of resources, the efficiency of computation, the energy costs, the amount of CO_2 emissions. Upon reception of such data from the other sites, the DCM executes the upper layer algorithms to: (i) determine the target data center to which a new application or VM should be assigned, in accordance to the specified goals; (ii) check if the workload is efficiently distributed among the different sites and trigger migration of applications when needed. This strategy resembles the one used to cope with traffic routing in the Internet, where a single protocol – Border Gateway Protocol – is used to interconnect different Autonomous Systems (ASs), while every AS is free to choose its own protocol – e.g., RIP or OSPF – for internal traffic management.

The EcoMultiCloud framework was firstly presented in [1], where it was also compared to ECE (Energy and Carbon-Efficient VM Placement Algorithm) [2], the reference of non-hierarchical approaches that have full visibility about all VMs and servers. There, it was shown that the hierarchical approach does not cause performance degradation with respect to single layer algorithms, and in addition it offers notable advantages in terms of time to convergence (because the bigger problem is decomposed into several smaller ones), scalability, autonomy of sites, overall administration, information management. With respect to [1], here the work is significantly extended in many directions: (i) the algorithm for the assignment of VMs is generalized to include and balance several business goals; (ii) a new algorithm for triggering inter-DC VM migrations is defined and evaluated; (iii) a mathematical analysis is provided to confirm the validity of the approach; (iv) a thorough performance evaluation shows how energy costs can be reduced exploiting the time and space variability of energy prices.

The contribution of the paper is the following: Section II summarizes related work in the fields of data center optimization and geographical workload distribution; Section III presents the EcoMultiCloud architecture and specifies the roles assigned to the upper and lower layers, as well as their interaction; Section IV illustrates the algorithms adopted for the assignment and migration of applications, and offers a mathematical analysis that can be used both to predict the performance and tune the algorithms depending on the desired objectives; Section V illustrates the performance results obtained with a simulation study for a specific scenario including four data centers located in North America and Europe; finally, Section VI concludes the paper.

II. Related Work

Many successful efforts have been done to increase the physical efficiency of data centers; for example, for its components devoted to cooling and power distribution, and this is confirmed by the general decrease of the PUE (Power Usage Effectiveness Index), the ratio between the overall power entering the data center and the power needed for the IT infrastructure. However, much remains to be done in terms of the computational efficiency: for example, on average only a fraction of CPU capacity of servers – between 15% and 30% – is actually exploited, and this leads to huge inefficiencies due to the lack of proportionality between resources usage and energy consumption [3]. Improvements in this field are related to a more efficient management of the workload and a better use of the opportunities offered by virtualization. The efforts may be categorized in two big fields: workload consolidation within a single data center, and efficient workload management in geographical infrastructures that include several remote data centers.

Workload consolidation is a powerful means to improve IT efficiency and reduce power consumption within a data center [4]–[7]. Sheikhalishahi *et al.* [8] presented a multi-resource scheduling technique to provide a higher degree of consolidation in multi-dimensional computing systems. Some approaches - e.g., [9] and [10] - 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.

Self-organizing and decentralized algorithms have been proposed to improve scalability, since the problem of consolidation is known to be NP-hard when addressed with a centralized approach. In [11], the data center is modeled as a P2P network, and ant-like agents explore the network and collect information needed to migrate VMs and reduce power consumption. The approach presented in [12] decentralizes

part of the intelligence to single servers that take decisions based on local information, using probabilistic functions, while a central manager coordinates servers' decisions to efficiently consolidate the workload.

The problem is even more complex in geographically distributed data centers. Research efforts are focused on two related but different aspects [13]: the routing of service requests to the most efficient data center, in the so called *assignment* phase, and the live *migration* of portions of the workload when conditions change and some data centers become preferable in terms of electricity costs, emission factors, or more renewable power generation.

Several studies explore the opportunity of energy cost-saving by routing jobs when/where the electricity prices are lower [14], [15]. Some prior studies assume that the electricity price variations and/or job arrivals follow certain stationary (although possibly unknown) distributions [16]–[18]. Rao *et al.* [19] tackle the problem taking into account the spatial and time diversity in dynamic electricity markets. They attempt to minimize overall costs for multiple data centers located in different energy marketing regions. Shao *et al.* [20] study the effect of transmission delay introduced by the routing of service requests and related data across DCs. Yao *et al.* [17] propose a solution in which the power cost can be reduced under delay tolerant workloads. By exploiting temporal and spatial variations of both workload and electricity prices, they provide a power cost-delay trade off which is exploited to minimize power expenses at the cost of service delay. The considered target applications that can generate delay tolerant workloads are based on MapReduce programming, including searching, social networking, data analytics. Lučanin and Brandic [21] focus on the significant impact of "geotemporal inputs", i.e., the time- and location-dependent factors that may impact energy consumption in geographically distributed data centers. Among such factors, they consider real-time electricity pricing enabled by the deregulated electricity market, the cooling efforts needed at different sites and different times, and the availability of renewable energy. The scheduling of the VMs in a geographical context is tackled through a two-stage approach, which combines best-effort global optimization, driven by genetic algorithms, with deterministic local optimization for constraint satisfaction.

Liu *et al.* [14] propose a geographical load balancing (GLB) approach to route general Internet service-requests to data centers located in various geographical regions, by computing the optimal number of active servers at each data center. Yu *et al.* [22] propose a GLB algorithm to minimize energy cost and control the risks at the same time, as they model the uncertainties of price and workload as risk constraints. Luo *et al.* [23] exploit temporal and spatial diversities of energy price to trade service delay for energy cost. The authors proposed a novel spatio-temporal load balancing approach to minimize energy cost for distributed DCs. The algorithms presented in [24] and [25] tackle the problem considering the user's point of view, and aim to choose the most convenient data center to which the user should consign a service or VM.

Inter-DC VM migration is a more novel research topic, as virtualization infrastructures have not offered such features so far. However they will do in the near future: for example, the vSphere 6.0 release of VMware includes new long-distance live migration capabilities, which will enable VM migrations across remote virtual switches and data centers. While opportunities opened by long distance migrations are big, involved issues are also extremely complex: among them, determine whether the benefits of workload migrations overcome the drawbacks, from which site and to which site to migrate, what specific portion of the workload should be migrated, how to reassign the migrating workload in the target site, etc.

Some significant efforts have been done in this area. The electricity price variation, both across time and location, is exploited to reduce overall costs using different strategies. The Stratus approach [26] exploits Voronoi partitions to determine to which data center requests should be routed or migrated. In [27], an optimization problem is formulated aiming at minimizing operational costs. Ren *et al.* [28] use an online scheduling algorithm based on Lyapunov optimization techniques. Kayaaslan *et al.* [29] propose an optimization framework based on the observation that energy prices and query workloads show high spatio-temporal variation for throughput-intensive applications like Web search engines. The optimization framework is based on a workload shifting algorithm considering both electricity prices, to reduce the energy cost, and workload of data centers at the time of shifting, to reduce response time. Le *et al.* [30] consider VM placement in cloud for high performance applications. The authors propose VM migration policies across multiple data centers in reaction to variable power pricing. In order to adapt to the dynamic availability of renewable energy, Akoush *et al.* [31] argue for either pausing VM executions or migrating VMs between sites based on local and remote energy availability.

Most proposed approaches aim to solve the problem as a whole, in a centralized fashion, undergoing the risk of originating three main issues, as discussed in the introductory section: poor scalability due to the size of the problem and the heterogeneity of involved business objectives, poor ability to adapt to changing conditions (e.g., changes in amount of workload, electricity price or carbon taxes) and lack of autonomy of single data centers. To efficiently cope with these issues, we believe that it is necessary to decentralize part of the intelligence and distribute the decisions points, while still exploiting the centralized architecture and functionalities offered by virtualization infrastructures in single data centers. This naturally leads to a hierarchical infrastructure, in which single data centers manage the local workload autonomously but communicate with each other to route and migrate VMs among them. A self-organizing hierarchical architecture is proposed in [32], but so far it is limited to the management of a single data center. A recent study [33] proposes a hierarchical approach that combines inter-DC and intra-DC request routing. The VM scheduling problem is decomposed and solved at single data centers, and is able to combine different objectives, e.g., minimize electricity cost, carbon taxes and bandwidth cost. While the work certainly deserves attention, it only solves the routing problem and does not exploit the opportunity of

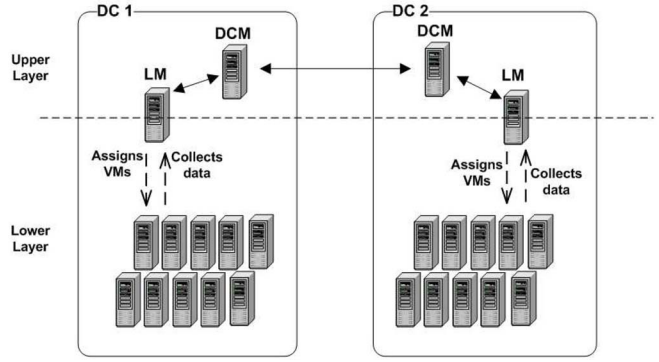

Fig. 1. EcoMultiCloud scenario: upper and lower layer of two interconnected data centers.

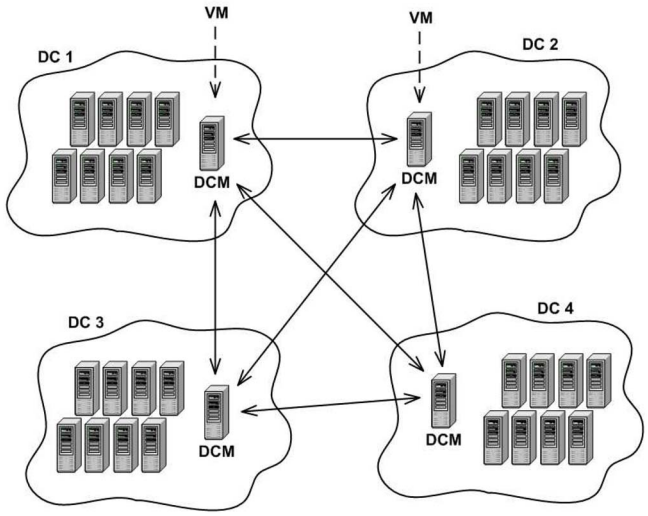

Fig. 2. EcoMultiCloud scenario: the DCMs of four data centers exchange high level information about the state of local data centers. Such information is used, for example, to decide which site should accommodate a new VM.

dynamic workload migration, nor the approach seems to be easily extensible in that direction.

To the best of our knowledge, our approach is among the first to offer a solution for the multi-DC scenario that exploits the benefits of a hierarchical architecture, balances multiple business objectives and constraints, and integrates algorithms for the assignment/routing problem and algorithms that trigger inter-DC migrations to adapt the workload distribution to varying conditions.

III. Architecture for Inter-DC Workload Distribution

This section describes the hierarchical architecture of EcoMultiCloud for the efficient management of the workload in a multi-site scenario. The architecture is composed of two layers: (i) the *upper layer* is used to exchange information among the different sites and drive the distribution of VMs among the DCs and (ii) the *lower layer* is used to allocate the workload within single DCs.

EcoMultiCloud extends the decentralized/self-organizing approach, recently presented in [12] and referred to as EcoCloud, for the consolidation of the workload in a single data center. With EcoCloud key decisions regarding the local data center are delegated to single servers, which autonomously decide whether or not to accommodate a VM or trigger a VM migration. The data center manager has only a coordination role. In a similar fashion, the EcoMultiCloud architecture leaves most of the intelligence to single DCs. At the lower layer, each DC is fully autonomous, and can manage the internal workload using either EcoCloud or any other consolidation algorithm. At the upper layer, coordinating decisions, for example about the necessity of migrating an amount of workload from one site to another, are taken combining the information related to single DCs. The upper layer algorithms may be tuned or modified without causing any impact on the operation of single sites.

The reference scenario is depicted in Figure 1, which shows the upper and lower layer for two interconnected DCs, as well as the main involved components. At each DC, a data center manager (DCM) runs the algorithms of the upper layer, while the local manager (LM) performs the functionalities of the lower layer. In the most typical case, both the DCM and LM may be deployed on the same host as the manager of the local virtualization infrastructure, e.g., the vCenter in the case of VMware. The DCM integrates the information coming from the lower layer and uses it to implement the functionalities of the upper layer. The DCM is required to: (i) communicate with the local LM in order to acquire detailed knowledge about the current state of the local DC, for example regarding the usage of host resources and the state of running VMs; (ii) extract relevant high level information about the state of the DC; (iii) transmit/receive such high level information to/from all the other DCMs; (iv) execute the algorithms of the upper layer to combine the collected information and take decisions about the distribution of the workload among the DCs. For example, the assignment algorithm is used to decide to which DC a new VM should be assigned. Once the VM is delivered to the target site, the local LM runs the lower layer algorithms to assign the VM to a specific host.

As depicted in Figure 2, the framework is designed so that all the DCMs execute the upper layer algorithms and, for example, choose the target DC for a VM originated locally. This requires an all-to-all data transmission among the DCMs. While the approach requires information exchanges that scale quadratically with the number of DCs, the number of interconnected sites is expected to be small. Moreover, the amount of information to be distributed is tiny, i.e., of the order of a few packets, since it contains only a general description of the status of the DC: pieces of information such as the load, the PUE, the electricity price need to be exchanged. The periodicity of the information exchange is of the order of an update per one or a few minutes, meaning that less than 1 Kbps of information exchange is needed. Thus, the choice of a distributed approach, based on all-to-all data exchanges, is not critical in terms of scalability and avoids the choice of a single coordination point that in a multi-site scenario may be inappropriate for administrative reasons. However, should the number of DCs grow to large values (say, above a few tens of sites) other kinds of interconnections are possible. As

an example, a hierarchical architecture can be adopted for the DCMs, in which DCs are organized in clusters; within clusters, the DCs coordinate in an all-to-all fashion like the one discussed in this paper. Periodically, each cluster leader communicates with other clusters' headers some information such as the total workload in the cluster, the average cost and energy consumption. Inter-cluster VM migrations can then be decided. In what follows, we will describe the case in which a few sites communicate with all the others in a full-mesh kind of architecture.

Since the single DCs are autonomous regarding the choice of the internal algorithms for workload management, the focus here is on the algorithms of the upper layer. Two basic algorithms are executed at each DCM: (i) the assignment algorithm that determines the appropriate target DC for each new VM; (ii) the migration algorithm that periodically evaluates whether the current load distribution is appropriate, decides whether an amount of workload should be migrated and, if so, determines from which source site to which target site.

IV. EcoMultiCloud Algorithms for Workload Assignment and Migration

As mentioned in the previous section, a key responsibility of the DCM is to analyze detailed data about the local data center and summarize relevant information that is then transmitted to remote DCMs and used for the assignment and redistribution of workload. The nature of the high level information depends on the objectives that must be achieved. Some important goals are:

1) Reduction of consumed energy. Moderns DCs are equipped with instrumentation to monitor the energy consumed in computational resources. The total energy, including that needed for cooling and power distribution, is obtained by multiplying the power used for computation by the PUE (Power Usage Efficiency) index;
2) Reduction of energy costs. The cost of electricity is generally different from site to site and also varies with time, even on a hour-to-hour basis, therefore the overall cost may be reduced by shifting portions of the workload to more convenient sites;
3) Reduction of carbon emissions. Companies are today strongly encouraged to reduce the amount of carbon emissions, not only to compel to laws and rules, but also to advertise their green effort and attract customers that are increasingly careful about sustainability issues;
4) Quality of service. The workload must be distributed without overloading any single site, as this may affect the quality of the service perceived by users. The quality of service may also be improved by properly combining applications having different characteristics, for example, CPU-bound and RAM-bound applications;
5) Load balancing among different sites. Among the rationales are: a better balance may help improve the responsiveness of the sites, decrease the impact on physical infrastructure – e.g., in terms of cooling and power distribution – and help prevent overload situations;
6) Inter-DC data transmission. The assignment/migration of VMs to remote sites should take into account many factors, among which the type of application hosted by the VM, the amount of involved data and the available inter-DC bandwidth. For example, migrating a VM may not be convenient in the case that the VM hosts a database server, while it may be appropriate if it runs a Web application, especially in the frequent case that Web services are replicated on several DCs.

TABLE I
Symbols and Notation

N_{DC}	Number of DCs
f^i_{assign}	Assignment function for DC i
F_i (F_{max})	Carbon emission for DC i (max c.e. among the DCs)
U_i (U_{max})	Utilization of DC i (max utilization among the DCs)
C_i (C_{max})	Energy cost for DC i (max energy cost among the DCs)
α, β, γ	Weights of the terms considered in the f_{assign}
PUE_i	Power Usage Effectiveness for DC i
P_i	Price of electricity for DC i
Λ	Total workload
U_{T_i}	Maximum allowed workload in DC i
l_i	Relative load of DC i wrt the best performing DC
X	Number of VMs to migrate after a change of some term
v_i	Migration speed (in VMs per second)
μ_i	VMs termination rate
E_{mig}	Energy consumed for VM migrations
D_{mig}	Amount of data to migrate

All the above mentioned goals are important, yet different data centers may focus on different aspects, depending on the specific operating conditions and on the priorities prescribed by the management. It is up to the company's management to specify the objectives and their relative weights. For example, let us assume that the primary objectives are the reduction of overall carbon emissions, the load balancing and the reduction of costs. These goals are representative of opposite needs, the need for optimizing the overall efficiency (in terms of costs and carbon emissions) and the need for guaranteeing the fairness among data centers. Such opposite needs are to be combined through properly defined weights, as described in the next section.

Next sections are devoted to the description of the two basics algorithms executed by the DCMs: the assignment and migration algorithms. To simplify reading, in Table I the notation used throughout the paper is reported.

A. Assignment Algorithm

The optimal distribution of the workload among the data centers is driven by a purposely defined *assignment function*, which balances and weights the chosen business goals. This function associates to each DC a value that represents the cost to run some workload in that DC, low values correspond to low overall cost of the DC. The strategy, then, is to assign a VM to the DC with the lowest value of the function. For example, if the objectives are the balance of load, the minimization of carbon emissions and the minimization of costs related to energy, the assignment function f^i_{assign}, for each DC *i*, is defined as follows:

$$f^i_{assign} = \alpha \cdot \frac{F_i}{F_{max}} + \beta \cdot \frac{U_i}{U_{max}} + \gamma \cdot \frac{C_i}{C_{max}} \quad (1)$$

function AssignmentAlgorithm(α, β, γ)
  while VM arrives
    for each remote datacenter DC_i
      Request values of F_i, U_i, C_i
    end for
    $F_{max} = Max\{F_i|\ i = 1 \cdots N_{DC}\}$
    $U_{max} = Max\{U_i|\ i = 1 \cdots N_{DC}\}$
    $C_{max} = Max\{C_i|\ i = 1 \cdots N_{DC}\}$
    for each $DC_i : DC_i$ is not full, that is, $U_i < U_{T_i}$
      $f^i_{assign} = \alpha \cdot \frac{F_i}{F_{max}} + \beta \cdot \frac{U_i}{U_{max}} + \gamma \cdot \frac{C_i}{C_{max}}$
    end for
    $DC_{target} = DC_j$ such that $f^j_{assign} = min\{f^i_{assign}|\ i = 1 \cdots N_{DC}\}$
    Assign VM to DC_{target}
  end while
end function

Fig. 3. The EcoMultiCloud assignment algorithm, executed by the DCM of each data center.

where the coefficients α, β and γ are positive and $\alpha + \beta + \gamma = 1$.

The function represents the decided balance among the various targets, and it corresponds to the strategic decision taken by the system administrator on how the system should work. It is applied to the system as a whole. The three terms F_i, U_i and C_i, are related, respectively, to carbon emissions, overall utilization and energy costs. The terms are normalized with respect to the maximum values communicated by DCs. The three mentioned goals – reduction of costs, reduction of carbon emissions and load balancing – are weighted through the values of the coefficients. After computing the values of f_{assign} for each DC, the VM is assigned to the data center having the lowest value. Once consigned to the target DC, the VM is allocated to a physical host using the local assignment algorithm, for example EcoCloud [12]. The assignment function and its effects will be discussed with more details in Section IV-C, with the help of a mathematical model.

To compute the assignment function, it is required that the DCM of each data center transmits to the others some very simple pieces of data, which are then used to compute the three mentioned terms. In the examined case, relevant information is: (i) the best available carbon footprint rate of a local server, f_s, (ii) the utilization of the bottleneck resource, U_i, and (iii) the energy cost, C_i. In this paper, for simplicity, we assume that the PUE value of each DC is known and constant. However, no modification is required to the algorithms if the PUE changes, as long as it is known. The carbon term F_i of a DC i, measured in Tons/MWh, defines the *best available carbon rate*, i.e., the carbon footprint rate (carbon emitted per consumed energy) of the most efficient available server [2], and is computed as:

$$F_i = PUE_i \cdot min\{f_s|\ server\ s\ is\ available\} \tag{2}$$

The rationale is that, when assigning a VM, the target DC should be chosen so as to minimize the incremental increase of the carbon footprint. To this aim, a DCM does not need to know the carbon footprint rate of all the servers of remote sites: it only needs to know, per each site, the minimum rate among the servers that are available to host the VM.

The utilization of the bottleneck resource is determined by computing the overall utilization of each hardware resource: CPU, RAM, storage, etc. For example, the utilization of CPU is defined as the total amount of CPU utilized by servers divided by the CPU capacity of the entire DC, and the utilization of other resources is computed in a similar way. The bottleneck resource for a DC i is the one with the highest value of utilization, and this value is denoted as U_i.

Finally, the energy cost term, C_i, is defined as:

$$C_i = PUE_i \cdot P_i \tag{3}$$

where P_i is the price of electricity (\$/kWh) and is assumed to be the same on all the servers of a data center. Indeed, the overall cost of energy is obtained by multiplying the energy consumed by the IT component of the data center first by the PUE – which gives the total amount of consumed energy, including power distribution and cooling – and then by the price of energy.

In conclusion, each DCM transmits to the other DCMs, the following vector of values, which corresponds to the *state* of the DC:

$$s_i = \{F_i, U_i, C_i\} \tag{4}$$

Figure 3 reports the pseudo-code used by a data center DCM to choose the target data center, among the N_{DC} data centers of the system, for a VM originated locally. First, the DCM requests the values of F_i, U_i and C_i to all the remote data centers.[2] Then, it computes the maximum values of the terms, for the normalization, and computes expression (1) for any data center that has some spare capacity, i.e., for which the utilization of the bottleneck resource has not exceeded a given threshold U_{T_i}. Finally, the VM is assigned to the DC that has the lowest value of (1). Once consigned to the target DC, the

[2]As an alternative, values can be transmitted periodically in a push fashion. In both cases the amount of transmitted information is tiny.

VM is allocated to a physical host using the local assignment algorithm.

B. Migration Algorithm

The assignment algorithm optimizes the distribution of the VMs on the basis of the chosen objectives and their respective weights. The values of the f_{assign} function tend to be equal in the different data centers, as discussed in detail in Section IV-C. However, the distribution may become inefficient when the conditions change, e.g., the load or the price of energy vary in one or more data centers. In such cases, inter-DC VM migrations are performed to redistribute the workload.

The migration algorithm is triggered when the values of the f_{assign} functions of two DCs differ by more than a predetermined threshold, for example in what follows we will use 3%.[3] The frequency at which this condition is evaluated should depend on the dynamism of the specific scenario, for example on the frequency at which the price of energy varies. When such an imbalance is detected, VMs are migrated from the data center having the highest value of f_{assign} to the data center with the minimum value, until the values reenter within the tolerance range. The frequency of migrations is limited by the bandwidth between the source and target data centers. This bandwidth may correspond to the physical bandwidth of inter-DC connections or may be a portion of the physical bandwidth reserved by data center administrators for this purpose. In some cases, a few migrations might be needed between two DCs simply to balance some load fluctuations that make the f_{assign} functions of the DCs differ more than desired. These events typically require only a few migrations. In other cases, a batch of migrations are instead necessary to compensate abrupt changes of the f_{assign} function in a DC, for example, due to some electricity price variations. When this happens, the process as described above translates into a sequence of VM migrations between pairs of DCs until a new balance among all the f_{assign} functions is reached. When the abrupt change makes a DC become the best performing DCs, multiple VM migration requests will be made by the other DCs; to avoid congestion on the access links of the receiving DC, the VM migration process can be easily coordinated by the DCM of the receiving DC.

Here, we would like to highlight the adaptive and self-organizing nature of the algorithm, as migrations are performed only when needed, asynchronously, and at predetermined and controlled rates. This is in contrast with most migration algorithms which require that the assignment of VMs is recomputed at fixed time intervals and generally need lots of concurrent migrations to achieve the new assignment pattern, possibly deteriorating the quality of service.

C. Analysis of the Assignment and Migration Algorithms

We now analyze the effect of the assignment function (1).

The function represents a metric of cost, that is the cost to run a VM in a given DC. As mentioned before, in the case of (1), three objectives are considered: reduction of cost, reduction of carbon emissions and load balance. Other objectives can also be considered and, in the general case, the expression to define the assignment function for DC i becomes:

$$f^i_{assign} = \sum_{k=1}^{M} \beta_k \frac{Y^k_i}{Y^k_{max}} \quad \text{with} \sum_{k=1}^{M} \beta_k = 1 \tag{5}$$

where M objectives are defined, based on the costs Y^k_i, normalized with respect to the maximum cost Y^k_{max}. The weights β_k sum to 1 to represent the relative importance of the various components of cost.

In what follows, the focus is shifted to the minimization of costs in the case that the price of energy varies both among different data centers, located in different countries, and with time. The general assignment function, given in (5), is thus instantiated to take into account two objectives, *load balancing* and *monetary cost minimization*, thus obtaining:

$$f^i_{assign} = \beta \cdot \frac{U_i}{U_{max}} + (1-\beta) \cdot \frac{C_i}{C_{max}} \tag{6}$$

The utilization U, defined as the overall utilization of the bottleneck hardware resource, and the energy cost C, are used to balance two opposite needs: the optimization of the overall efficiency and the fairness among data centers.

We order the DCs based on the value of C_i, so that, $C_i < C_j$ if $i < j$; in other words, we order the DCs from the best performing in terms of energy cost to the least performing one. We now study the effect of (6) on the steady-state working point of the multi-site system. The steady-state corresponds to the values of workload that are reached by the DCs, once the system has adapted to the values of the assignment functions. Given a total load Λ, at the steady-state, the load distributes among the DCs in such a way that all the DCs exhibit the same value of f^i_{assign}; in addition, no other load distribution would allow such a low value of f^i_{assign}. Indeed, the DCM allocates a VM to the DC i with the smallest value of f^i_{assign}; but, the allocation of the VM to the DC makes f^i_{assign} increase and get closer to the other functions f^j_{assign}. Thus, differences among the values of f^i_{assign} reduce and at the steady-state vanish. The reached steady-state minimizes the maximum value of f^i_{assign}.

The steady-state distribution of the load, denoted by the terms U^*_i, can be derived by the solution of the system of linear equations,

$$\begin{cases} \beta \frac{U^*_i}{U^*_{max}} + (1-\beta)\frac{C_i}{C_{max}} = \beta \frac{U^*_j}{U^*_{max}} + (1-\beta)\frac{C_j}{C_{max}} \\ \sum_{i=1}^{N_{DC}} U^*_i = \Lambda \quad \text{with} \quad 0 \le U^*_i \le U_{T_i} \end{cases} \tag{7}$$

where U_{T_i} is the maximum allowed workload in DC i.

Given the chosen DC ordering, $C_1 \le C_i$ and $U_1 \ge U_i$ for all i. The DC 1 is, thus, the most loaded DC and $U_{max} = U_1$. We call DC 1 the *reference* DC. The system (7) leads to the

[3]In our scenario, the threshold was chosen after some trials not reported here for the sake of brevity.

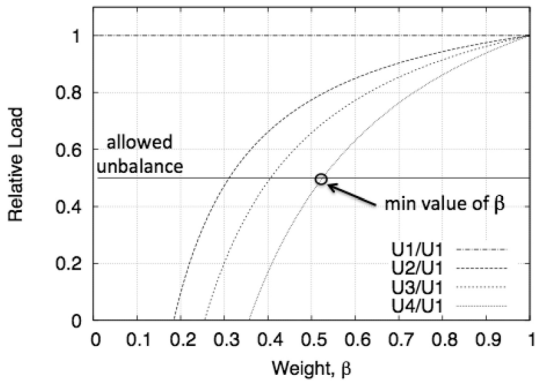

Fig. 4. Relative load versus β for a case with 4 DCs.

following solution,

$$\begin{cases} U_i^* = U_1^* \left[1 - \frac{1-\beta}{\beta C_{max}}(C_i - C_1)\right] = U_1^* l_i \\ \sum_{i=1}^{N_{DC}} U_i^* = \Lambda \end{cases} \tag{8}$$

The load distributes in such a way that the larger the terms $(C_i - C_1)$ is (that means the larger the cost to run a VM in DC i is), the smaller the load allocated to DC i is.

The terms $l_i = 1 - (1-\beta)/(\beta C_{max})(C_i - C_1)$ define the *relative load* of the DCs with respect to the best performing DC (the reference DC). The term β corresponds to the weight of load balance with respect to monetary costs; as $\beta \to 1$, the policy tends to a pure load balance in which the load is the same for all the DCs; i.e., $l_i \to 1$.

As an example, Figure 4 shows the relative load for the case with N_{DC}=4 DCs, with energy prices and PUE values taken from the scenario that will be discussed in Section V. Clearly, as the weight β decreases, the importance of load balance decreases, and the gap among values of the load in the various DCs increases. Figure 4 can be used to define the setting of β. Assume, for example, that a load balance target imposes that the relative load between DCs cannot be smaller than 0.5, i.e., one DC cannot have more than twice the load of another DC. Then, from the figure, we can find that the minimum possible value for β that guarantees this load balance target is 0.52. This value, or a slightly higher one, should then be used if the objective is to minimize the monetary cost while respecting the constraint on the load balance.

In the solution of (7), some values of U_i^* might turn to be negative. These are the cases in which the corresponding DC i has such a high cost that it is more convenient to allocate the VMs to the other better performing DCs, i.e., DC j, with $j < i$. In Figure 4, for example, when $\beta = 0.3$, the VMs are assigned to DC 1, 2, 3 while the fourth DC is not used. Moreover, when Λ is large, the solution of (7) leads to some $U_i > 1$. Clearly, these solutions are not acceptable. In these cases, the corresponding DCs are fully loaded and the additional load is distributed among the less performing DCs.

One of the main characteristics of the proposed solution is the possibility to adapt the load allocation to changes of the considered terms, that is, in the case considered above, to changes of the electricity cost. When the cost C_i varies, for example due to electricity tariffs that have daily variations, the system adapts to it by changing the allocation of the load to the DCs, i.e., the values of U_i. In particular, the variations of load must follow those of C_i according to the derivative of (8),

$$\frac{dU_i}{dC_i} = -\frac{U_1^*}{C_{max}} \frac{1-\beta}{\beta} \tag{9}$$

An increase of C_i causes a decrease of the load associated to DC i; the decrease depends on the parameter β and, similarly, a decrease of C_i causes an increase of the load of DC i. These load changes are performed through migrations: in the first case, migrations out of DC i are needed, in the second case, some VMs have to be migrated to DC i. Migration flows will occur in proportion to the values of the relative loads under the new conditions, i.e., after the change of C_i. In practice, the migrations can be coordinated by the DCM of DC i.

The effectiveness of the adaptation of load to variations of cost depends on the relative timescale of tariff variations with respect to VM arrivals and departures. For example, when electricity tariffs change a few times per day, as is usually the case, systems with highly dynamic VM arrivals and departures easily and quickly adapt to tariff changes. Conversely, when VMs lifetime is of the order of days, the system is too slow to adapt to tariff variations; in this case, VM migrations are needed to make the system adaptive.

Assume that, at some time of the day, the cost C_i increases of a quantity ΔC_i and that the relative load of a VM in DC i is given by u_i. The variation of the number of VMs in DC i that is needed to reach the new optimal load allocation, is given by

$$X = -\frac{\Delta C_i}{u_i} \frac{U_1^*}{C_{max}} \frac{1-\beta}{\beta} \tag{10}$$

where the negative terms corresponds to the fact that if the energy price increases, the number of VMs should decrease.

To reach the new desired load allocation, DC i has to get rid of X VMs. This can be achieved by the natural termination of the lifetime of a VM, and by migrations. Assume that the rate at which VMs terminate is given by μ_i VMs/s; in general, μ_i depends on the average VM lifetime and on the typical number of VMs in the DC. If we want to guarantee that the optimal allocation after the tariff change is reached in a time T_m (that stands for *time for migrating*), we need to guarantee that, in addition to VM terminations, some VMs are migrated out of DC i at speed v_i that can be computed as,

$$(\mu_i + v_i) T_m \geq -X \tag{11}$$

That is, the number of VMs that leave the DC (due to either VMs termination, with rate μ_i, or migration, with rate v_i) in the time T_m must be larger than X. Hence, the migration speed must be,

$$v_i \geq \frac{-X}{T_m} - \mu_i = \frac{\Delta C_i}{T_m u_i} \frac{U_1^*}{C_{max}} \frac{1-\beta}{\beta} - \mu_i \tag{12}$$

As discussed previously, the VMs migrate from DC i to the other DCs based on the differences between the functions f_{assign}^i.

Assume that the migration speed v_i is set according to (12) and such that the duration of the migration period is $T < T_m$;

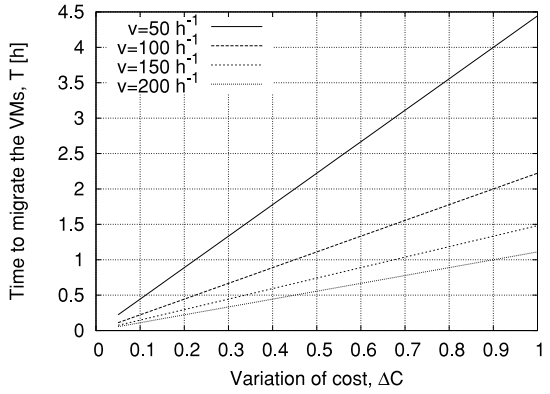

Fig. 5. Time to migrate the VMs versus variations of the energy price ΔC, for several values of the migration speed; $\beta = 0.5$.

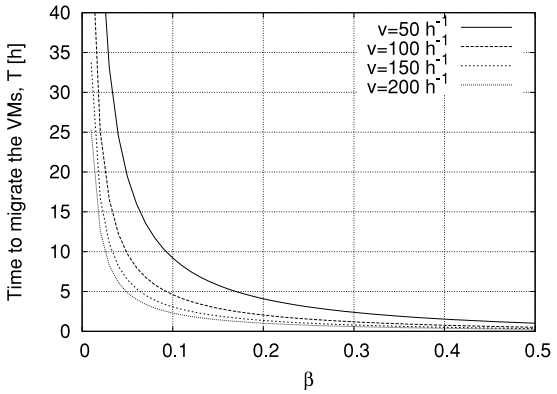

Fig. 6. Time to migrate the VMs versus β, for several values of the migration speed, $\Delta C = 0.2$ \$/kWh.

with $T = -X/(v_i + \mu_i)$. As can be seen in Fig. 5, in scenarios like the one considered above, the time to migrate the VMs is of the order of a few hours. Increases of the migration speed can have very beneficial effect when large price variations occur. For example, for $\Delta C = 0.5$ \$/kWh, doubling the migration speed from 50 to 100 VMs per hour reduces the time T to about an 1 h. However, when β is small, variations of electricity price translate into very large numbers of needed migrations and long times to converge to the optimimal load distribution. This behavior is shown in Fig. 6. For $\beta = 0.1$, up to 10 h are needed when the migration speed is low. Similar considerations can be done for the case in which the energy cost at DC i decreases and some VMs have to be migrated to DC i.

V. Performance Evaluation

This section is devoted to the performance evaluation of EcoMultiCloud. The evaluation is organized in three main parts. In the first one, we focus on the potential cost into which the system incurs if it does not adapt to time-varying energy price, and to isolate this effect we assume a stable load and observe the system working conditions after the assignment phase. In the second part, we examine how migrations can help to make the workload distribution adaptive to changes of the electricity price. In the third part, we estimate the energy consumption, including the amount of energy needed for VM migrations.

TABLE II
PUE and Local Time of the Four DCs in the Examined Scenario

Data center	PUE	Local time
DC 1	1.56	UTC-8
DC 2	1.7	UTC-5
DC 3	1.9	UTC
DC 4	2.1	UTC+1

TABLE III
Energy Price, Expressed as \$/*kWh*, for the 4 DCs. The Table Shows Only the Time, Expressed in UTC, Corresponding to the Energy Price Change in at Least One DC

Time (UTC)	DC_1	DC_2	DC_3	DC_4
0:00 am	0.15	0.07	0.09	0.11
2:00 am	0.11	0.07	0.09	0.11
6:00 am	0.08	0.07	0.09	0.11
7:00 am	0.08	0.07	0.14	0.21
11:00 am	0.08	0.07	0.19	0.21
12:00 am	0.08	0.12	0.19	0.21
2:00 pm	0.08	0.12	0.14	0.21
4:00 pm	0.08	0.10	0.19	0.21
5:00 pm	0.11	0.10	0.19	0.21
7:00 pm	0.11	0.10	0.14	0.11
8:00 pm	0.15	0.10	0.14	0.11
10:00 pm	0.15	0.12	0.14	0.11
11:00 pm	0.15	0.12	0.09	0.11

As mentioned in the introductory section, a careful analysis of the hierarchical approach was already performed in a previous work [1] by comparing the results of EcoMultiCloud with the reference of non-hierarchical approaches, namely ECE (Energy and Carbon-Efficient VM Placement Algorithm) [2]. Thus, our purpose here is not to validate the hierarchical approach, but, rather, to focus on the minimization of costs in the case that the energy price varies both among DCs located in different countries, and with time.

We consider a system with two objectives, load balancing and cost minimization, that are reflected by the assignment function in (6), reported here for the reader's convenience,

$$f^i_{assign} = \beta \cdot \frac{U_i}{U_{max}} + (1 - \beta) \cdot \frac{C_i}{C_{max}}$$

The function works with two terms per DC: the utilization U_i and the energy cost C_i. The scenario under analysis is the same of [1] and [2], with four interconnected DCs and values of the PUE as reported in Table II; time zones are also indicated with respect to UTC, assuming that the DC locations are, respectively, California, Ontario (Canada), U.K. and Germany. Table III reports energy prices in a 24 hours interval, again time is expressed in UTC.[4] To simplify the analysis, it is assumed that the prices repeat periodically for a few days. The term C_i is obtained by multiplying the PUE of DC i as in Table II by the electricity price reported in Table III.

Data about VMs and physical hosts are taken from the logs of a Proof of Concept performed by the company

[4]Energy prices are taken or extrapolated from the following Web sites:
- California: www.pge.com/tariffs/IndustrialCurrent.xls.
- Ontario: www.hydroone.com/RegulatoryAffairs/RatesPrices/Pages.
- U.K.: en.wikipedia.org/wiki/Electricity_billing_in_the_UK.
- Germany: www.iwr-institut.de/en/press/background-informations.

Eco4Cloud srl (www.eco4cloud.com), spin-off from the National Research Council of Italy, on the DC of a telecommunication operator. The DC contains 112 servers virtualized with the platform VMware vSphere 5.0. Among the servers, 76 are equipped with processor Xeon 24 cores and 100-GB RAM, and 36 with processor Xeon 16 cores and 64-GB RAM. All the servers have network adapters with bandwidth of 10 Gbps. The servers host 2000 VMs which are assigned a number of virtual cores varying between 1 and 8 and an amount of RAM varying between 1 GB and 16 GB. The most utilized resource in this scenario is the RAM, therefore the RAM utilization of DC i is considered when computing the utilization U_i in (6). A constraint imposed by the DC administrators was that the utilization of server resources must not exceed 80%, i.e., $U_{T_i} = 0.8$. Servers and VMs are replicated for all the DCs, while the values of PUE and energy price are differentiated as described above.

The performance is analyzed with an event-based Java simulator that was previously validated with respect to real data for the case of a single DC [12]. At a time UTC=0, corresponding to midnight for DC 3 that is located in U.K., all the VMs are assigned one by one by executing the assignment algorithm described in Section IV-A: (i) each VM is delivered by the local DCM to the DC having the lowest value of the assignment function f_{assign}; (ii) within the target DC, the VM is assigned to a specific host using, as local assignment algorithm, the EcoCloud algorithm presented in [12], which proved to achieve a nearly optimal degree of workload consolidation.[5]

Results are obtained, unless otherwise stated, for β=0.5 and total load Λ=50%. Since the RAM is the bottleneck resource, the *overall load* Λ of the system is defined as the ratio between the total amount of RAM utilized by the VMs and the RAM capacity of the entire system. Thus, the overall number of VMs is chosen so as to load the whole system to the desired extent.

A. Constant Load, No Migration

In the first scenario, the number of running VMs is assumed to be stable: no VM terminates or is generated, and inter-DC migrations are not allowed.[6]

Table IV reports the values of RAM utilization of the DCs, at the end of the assignment phase, for different values of the overall load Λ (50% and 75%) and β (0, 0.5 and 1). The table shows that, for any given load, the parameter β can be used to tune the two objectives, cost minimization and load balance. With β=1, all the DCs are equally loaded since load balancing is the only objective. With β=0, the load is preferably assigned to the DCs with lowest energy price, which are loaded up to their maximum capacity. With β=0.5, there is a tradeoff between the two objectives: for example, with overall load equal to 50%, the RAM utilization ranges between 36.2% (at the most expensive DC at time of assignment, namely DC 1) to 69.4% (at the most convenient DC, DC 2).

[5]Any other efficient consolidation algorithm can be adopted as local assignment algorithm, with no remarkable effect on the overall performance of multi-DC assignment.

[6]The hardware requirements of single VMs, as extracted by the real traces used for the experiment, are dynamic.

TABLE IV
RAM UTILIZATION OF THE DCS WITH DIFFERENT VALUES OF β AND OVERALL LOAD Λ, AT THE END OF THE ASSIGNMENT PHASE

Λ	β	U_1	U_2	U_3	U_4
50%	0	0.0%	79.9%	79.9%	36.4%
50%	0.5	36.2%	69.4%	51.3%	39.2%
50%	1	49.9%	49.9%	49.9%	49.9%
75%	0	55.8%	79.9%	79.9%	79.9%
75%	0.5	66.1%	79.9%	79.9%	69.6%
75%	1	75.0%	75.0%	75.0%	75.0%

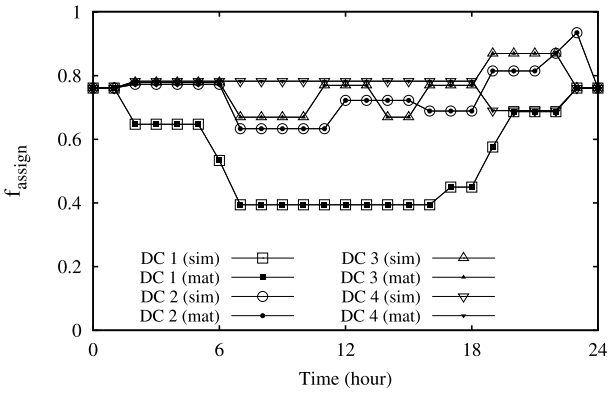

Fig. 7. Values of f_{assign} vs. time. Results obtained with simulation and mathematical analysis are reported for comparison.

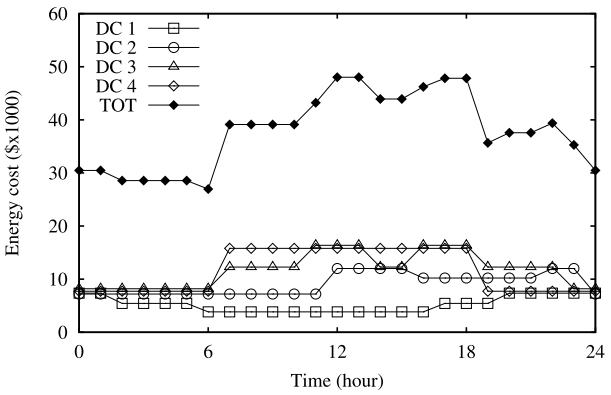

Fig. 8. Energy cost per hour vs. time for the single DCs. The total cost is the sum of the four costs.

Figure 7 shows the values of the f_{assign} function for the different DCs, as obtained from simulations and from the mathematical analysis illustrated in Section IV-C, for the scenario with β=0.5 and overall load Λ=50%. Right after the assignment phase (executed at time 0), the values of the f_{assign} function for the different DCs are the same, as discussed and anticipated in Section IV-C. Subsequently, due to energy price variations during the day, the values of f_{assign} vary and differentiate from each other, which is a sign that the initial assignment becomes inefficient (and it cannot be modified since migrations are not allowed and VMs do not start or terminate). For example, at time UTC=7 the f_{assign} value of DC 4 is higher than the value of DC 1, therefore it would be advantageous to move a portion of the workload from the former DC to the latter.

Figure 8 shows the hourly cost of the energy consumption for the single DCs as well as the total cost. Clearly, the costs

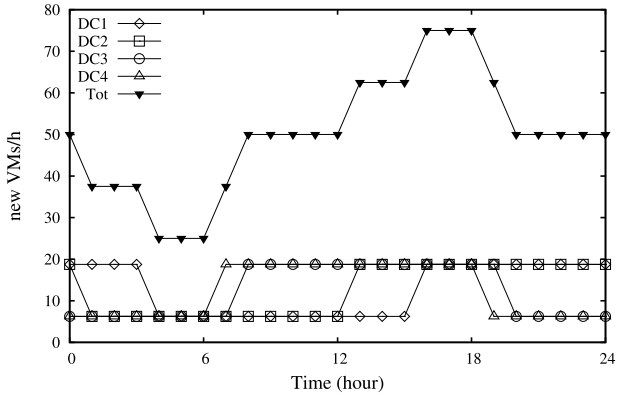

Fig. 9. Arrival rate of new VMs per hour.

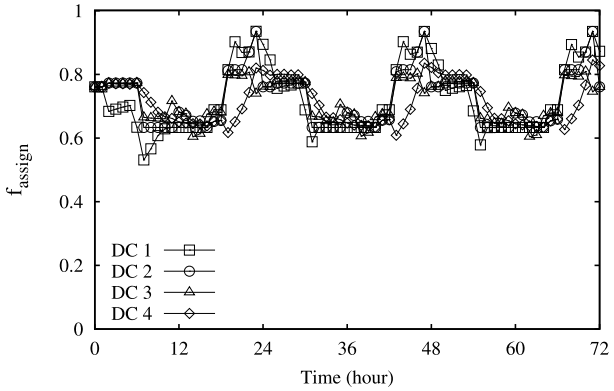

Fig. 10. Values of f_{assign} vs. time for the first three days after the initial assignment.

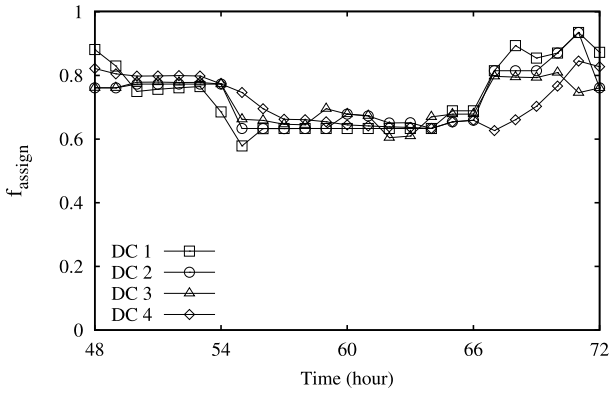

Fig. 11. Values of f_{assign} during the third day after the initial assignment.

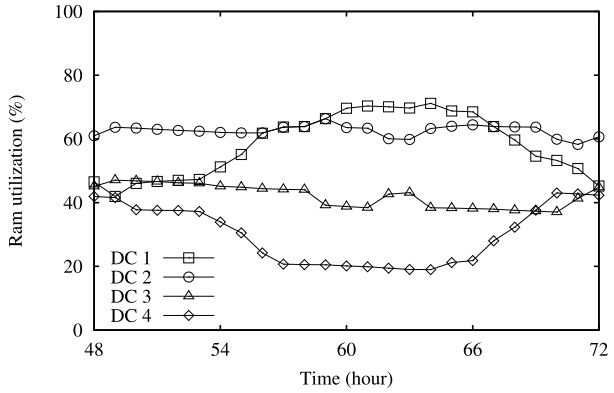

Fig. 12. Utilization of DCs vs. time during the third day after the initial assignment.

are heavily affected by the variations of energy price along the 24 hours. In the next section, it is shown that the costs can be notably reduced through inter-DC migrations.

B. Dynamic Load and Migrations

The second set of experiments is performed in a scenario in which two phenomena are enabled: the turnover of VMs and the inter-DC migration process. The first phenomenon is related to the arrival and departure of VMs. We assume that new VMs are launched at different rates during the day and the night, namely λ_{day} and λ_{night}, and that $\lambda_{day} = 2 \cdot \lambda_{night}$. Figure 9 reports the arrival rates at the four DCs in a 24 hours interval, and the overall arrival rate to the whole system. We also assume that the average lifetime of a VM, denoted as $1/\mu$, is equal to 180 hours.

The variations of energy price and the arrival/departure process contribute to break the equilibrium achieved at the assignment phase. Inter-DC migrations are then used to properly redistribute the workload, as explained in Section IV-B. Workload migration is triggered when f_{assign} values of two DCs differ by more than 3%, which is checked at intervals of 60 minutes. Experiments were performed with different values of the bandwidth that is available or reserved for DC migrations: 0.5 Gbps, 1 Gbps, 2 Gbps and 5 Gbps. In the examined scenario, such values of bandwidth enable the migration, respectively, of about 50 VMs, 100 VMs, 200 VMs and 500 VMs per hour. When it is not specified, a bandwidth of 2 Gbps is assumed. The results reported in the following are related to a 24-hour interval corresponding to the *third day* after the initial assignment of VMs. This allows the results to become independent from the conditions experienced at the time of the initial assignment, in particular from the price of energy at that time. Indeed, it was observed that the biasing caused by the initial conditions vanishes after the first day, thanks to inter-DC migrations that are performed during this time. This can be observed in Figure 10: values of f_{assign} repeat cyclically every 24 hours, starting from the second day.

Figure 11 focuses on the values of f_{assign} during the third day. While the variations of energy prices tend to stretch f_{assign} values apart, as previously seen in Figure 7, inter-DC migrations let the functions approach each other, making the workload distribution more efficient. Figure 12 shows that the load of DCs adapts to the energy price variations. For example, at the time labeled as 60 (12 am UTC of the third day), the most loaded DCs are DC 1 and DC 2, because in the preceding hours they have been the DCs with the lowest energy price – see Table III – and have then attracted VMs from the other two DCs.

Figure 13 shows the energy costs of the four DCs. The inter-DC migration process makes costs closer to each other, as can be observed by comparing this figure to Figure 8. Most importantly, the total cost notably reduces: Figure 14 reports the total energy cost obtained with two different values of

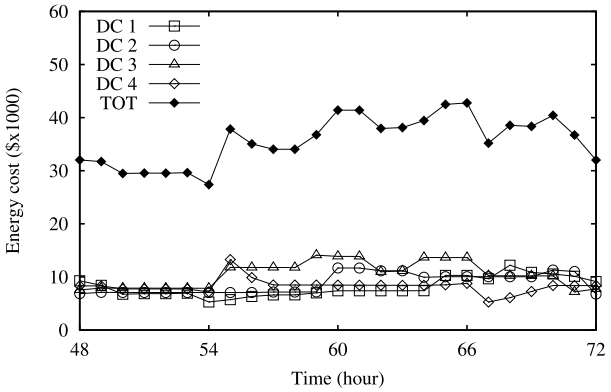

Fig. 13. Energy cost per hour vs. time during the third day after the initial assignment.

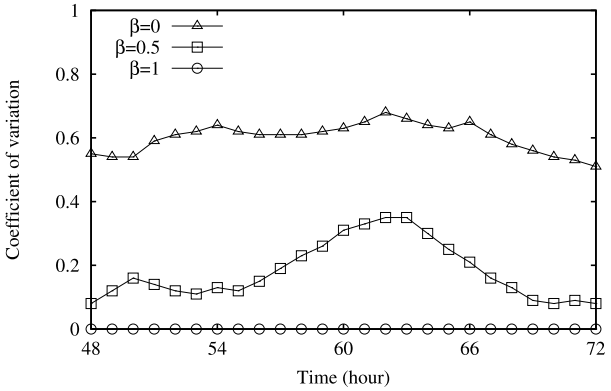

Fig. 16. Coefficient of variation in the third day of operation for different values of β.

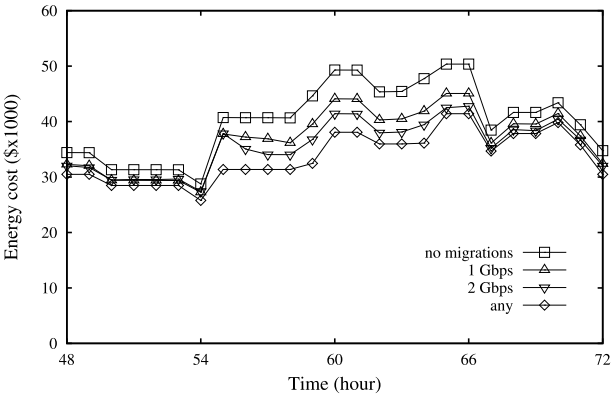

Fig. 14. Total energy cost per hour vs. time during the third day after the initial assignment, with different allowed migration rates.

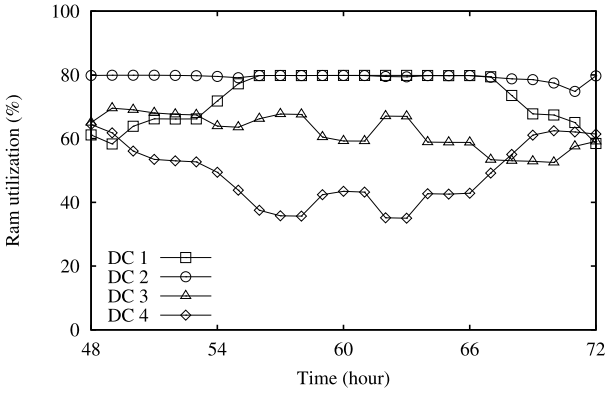

Fig. 17. Utilization of DCs vs. time during the third day after the initial assignment. Overall load Λ=75%.

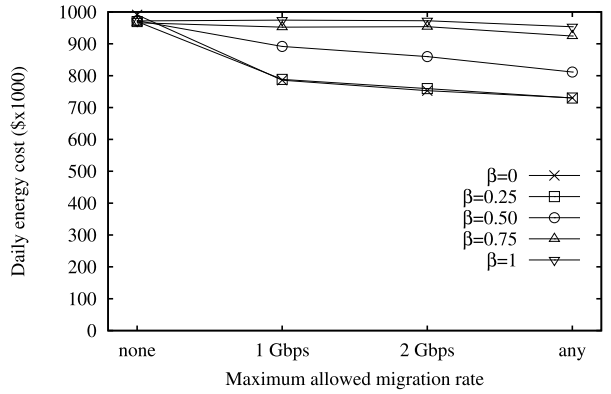

Fig. 15. Total daily energy cost in the third day vs. the allowed migration rate, for different values of β.

inter-DC bandwidth and, for the sake of comparison, in the case that migrations are disabled (curve "no migrations") and in the case that the migrations are instantaneous, taken as a theoretical limit. Cost savings clearly increase with the allowed bandwidth.

The total daily cost of energy is reported in Figure 15, for different values of inter-DC bandwidth and β. In the case examined so far, with β=0.5, the daily cost is equal to about \$973,000 if migrations are not allowed, while it is about \$860,000 when the bandwidth is 2 Gbps, resulting in a cost saving of about \$113,000, corresponding to 11%. Cost savings are even higher with β=0, since the load balancing is not taken into account, and cheaper DCs are able to attract more VMs. In this case, the daily saving increases to about \$219,000, or 21%. Conversely, with β=1, all the DCs support the same load and, since the load balance is the only objective, no inter-DC migrations are triggered even when allowed, and no cost saving can be achieved. It is also noticed that lower values of β correspond to lower values of daily cost, as expected, except when no inter-DC migrations are allowed.

Figure 16 helps to understand the effect on the load balancing objective. The figure reports the value of the coefficient of variation in the third day of operation with three different values of β. The index is computed by considering the RAM utilization of the four DCs and dividing the standard deviation by the average. With β=1 the DCs are equally loaded, as desired. With β=0 the distribution of load is completely determined by costs, so the imbalance is maximum. Finally, with β=0.5 the values are intermediate between the two extreme cases, and the large fluctuations reflect the fact that inter-DC migrations are used to dynamically redistribute the load as required by the varying values of energy price.

Finally, Figure 17 shows the utilization of the DCs for the case of a higher overall load, i.e., $\Lambda = 0.75$, for $\beta = 0.5$. It is observed that when the load is

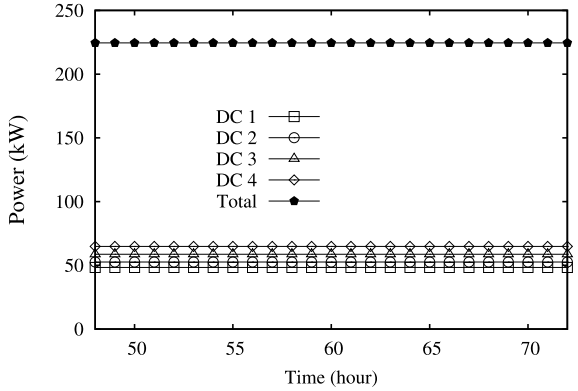

Fig. 18. Power consumption of DCs vs. time during the third day after the initial assignment, with β=1. Overall load Λ=50%.

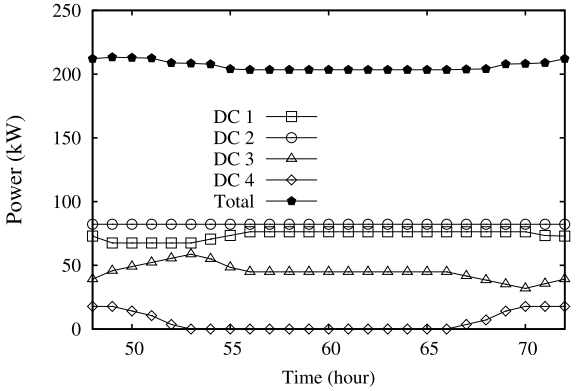

Fig. 19. Power consumption of DCs vs. time during the third day after the initial assignment, with β=0. Overall load Λ=50%.

high, the most convenient DCs may reach full utilization, but the actual load distribution still depends also on the energy cost.

C. Energy Consumption

The business objectives of the data center administrators are established through the choice of the appropriate components in the expression of the assignment function (5) and through the tuning of the corresponding weights. In (6), the two chosen objectives are the reduction of energy costs and the load balance. However, other metrics can be affected indirectly. For example, the cost of the consumed energy depends on both the price of electricity and the PUE, as stated in (3), since the amount of consumed energy depends on the PUE. If the data center administrators operate to reduce the energy costs, they can also achieve a significant reduction in the energy consumption, though the latter objective is not specifically declared.

This beneficial effect is indeed observed in the examined scenario. Figure 18 shows the amount of energy that is consumed by the four data centers, and the total consumption, in the scenario examined in Section IV-B, with $\Lambda = 50\%$ and $\beta = 1$. Since the data centers are equally loaded ($\beta = 1$), the energy consumption at the data centers is proportional to the values of the PUE index. The overall energy consumed in the data centers in one day of operation is 5388 kWh. Figure 19 reports the energy consumed with $\beta = 0$, i.e., when the objective is the reduction of energy costs only. In this case, the energy consumption is reduced because less load is assigned to data centers with higher values of the PUE. The energy savings are lower than cost savings, both because energy saving is not the the primary objective and because the PUE values are not so different among each other. However, the energy saving is significant: the overall amount of energy consumed in one day reduces to 4950 kWh when $\beta = 0$, which corresponds to an energy saving of 8.12%.

The migration of VMs leads to incremental energy consumption. We now quantify this extra-consumption and show that it is negligible with respect to the total amount. In [34], an accurate model for energy consumption due to VM migration is presented[7]: the consumed energy E_{mig}, expressed in Joules, can be computed as:

$$E_{mig} = m \cdot V_{mig} + n \tag{13}$$

In this expression, V_{mig} is the amount of migrated data, measured in megabytes, and m and n are parameters whose values depend on the approach adopted for live migration. To make the migration procedure transparent to the user, the VMware virtualization platform uses the *precopying* algorithm, i.e., the memory pages are pushed across the network to the new destination while the source VM continues running, and the memory pages get dirtied during the migration are iteratively re-sent to ensure memory consistency. The parameters are trained in [34] using linear regression and ordinary least squares estimation, and the obtained values are $m = 0.512$ and $n = 20.165$, which we use for our estimation.

In Section IV-C, the amount of data involved in VM migrations was estimated for the case that, after the values of the function f_{assign} of the data centers have reached a steady state and are equal to each other, a variation ΔC of the energy cost is experienced at one of the data centers. In such a case, the amount of data to migrate to return to the equilibrium condition, D_{mig}, is:

$$D_{mig} = -\Delta C_i \cdot \frac{U^*}{C_{max}} \cdot \frac{1-\beta}{\beta} \tag{14}$$

where U^* is the overall RAM utilization of the most loaded data center and C_{max} is the maximum cost of energy among the four data centers.

For example, let us take the case of the experiment discussed in Section IV-B, with $\Lambda = 50\%$ and $\beta = 0.5$. After the initial assignment, the energy price of DC 1 decreases from 0.15 \$/*kWh* to 0.11 \$/*kWh* (see Table III). The PUE of DC 1 is equal to 1.56 (Table II), so the corresponding variation ΔC of the energy cost is $(0.11 - 0.15) \cdot 1.56 = -0.0624$ \$/kWh.

[7]The model derived in [34] only considers the energy drawn by each migration side, while it ignores the energy consumed by the switching fabric during the migration. The reason is that in a geographical scenario the network connecting the two ends can be very complex, so the consumed energy is hard to quantify. Moreover, since the network elements are typically owned by multiple Internet and telecommunications companies, it is questionable if the consumed energy should be accounted in the energy balance of the data centers.

The value of D_{mig} is then[8]:

$$D_{mig} = 0.0624 \cdot \frac{9.9 \cdot 10^6 \cdot 0.694}{0.234} \cdot \frac{1 - 0.5}{0.5} MB = 1.856 \cdot 10^6 MB \quad (15)$$

The energy consumed to migrate this amount of data, according to (13), is about 950,000 Joules, or about 0,264 kWh. Notice that this is an approximate computation mainly because the values of the parameters m and n in (13) are taken from [34] and suited to represent the scenario that is considered there; their values should be evaluated for other considered environments. However, it is clear that the expected consumed energy is very low with respect to the overall energy consumed in the data centers, and can be neglected in this context.

VI. Conclusion

The paper focused on the challenging task of workload management in multi-site data centers. A new hierarchical approach, named EcoMultiCloud, was presented and evaluated. The proposed solution is based on a function that defines the cost of running some workload on the various sites of the distributed data center. The function can be tailored to properly trade-off the various possible management goals, such as energy cost reduction and load balance. Moreover, the solution preserves the autonomy of the sites for the internal management. The presented results show that the proposed solution, despite being simple and requiring a very limited information exchange among the sites, smoothly adapts the workload distribution to variations of the working conditions, such as changes of the energy cost and daily load fluctuations. Future research will be devoted to the definition of techniques based on a differentiated management of different classes of VMs, both in the assignment phase and in the migration phase. For example, the migration of CPU-intensive VMs can be particularly useful to reduce energy consumption and energy costs, because these VMs can be migrated more easily than disk- and RAM-intensive VMs, and because energy consumption is more sensitive to the variations of CPU utilization than to the variations of other hardware resources' utilization.

References

[1] A. Forestiero, C. Mastroianni, M. Meo, G. Papuzzo, and M. Sheikhalishahi, "Hierarchical approach for green workload management in distributed data centers," in *Proc. Euro-Par Parallel Process. Workshops*, vol. 8805. 2014, pp. 323–334.

[2] A. Khosravi, S. K. Garg, and R. Buyya, "Energy and carbon-efficient placement of virtual machines in distributed cloud data centers," in *Proc. Euro-Par Parallel Process.*, vol. 8097. 2013, pp. 317–328.

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

[4] M. Cardosa, M. R. Korupolu, and A. Singh, "Shares and utilities based power consolidation in virtualized server environments," in *Proc. 11th IFIP/IEEE Integr. Netw. Manag. (IM)*, Long Island, NY, USA, Jun. 2009, pp. 327–334.

[8]The most loaded data center is DC 2, utilized at 69.4%, as reported in the second row of Table IV, and the total RAM capacity of the same DC is $9.9 \cdot 10^6$ MB. The data center with the highest cost of energy is DC 1, as can be derived from the first rows of Tables II and III, so $C_{max} = 0.234$ \$/kWh.

[5] P. Graubner, M. Schmidt, and B. Freisleben, "Energy-efficient virtual machine consolidation," *IT Prof.*, vol. 15, no. 2, pp. 28–34, 2013.

[6] K. Schröder and W. Nebel, "Behavioral model for cloud aware load and power management," in *Proc. Int. Workshop Hot Topics Cloud Services (HotTopiCS)*, Prague, Czech Republic, 2013, pp. 19–26.

[7] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Gener. Comput. Syst.*, vol. 28, no. 5, pp. 755–768, 2012.

[8] M. Sheikhalishahi, R. M. Wallace, L. Grandinetti, J. L. Vazquez-Poletti, and F. Guerriero, "A multi-capacity queuing mechanism in multi-dimensional resource scheduling," in *Adaptive Resource Management and Scheduling for Cloud Computing* (LNCS 8907). Cham, Switzerland: Springer, pp. 9–25.

[9] Y. Chen *et al.*, "Managing server energy and operational costs in hosting centers," *ACM SIGMETRICS Perform. Eval. Rev.*, vol. 33, no. 1, pp. 303–314, Jun. 2005.

[10] M. Mazzucco, D. Dyachuk, and R. Deters, "Maximizing cloud providers' revenues via energy aware allocation policies," in *Proc. 10th IEEE/ACM Int. Symp. Clust. Comput. Grid (CCGrid)*, Melbourne VIC, Australia, May 2010, pp. 131–138.

[11] 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. 1st Int. Conf. Self Organizing Architect. (SOAR)*, Cambridge, U.K., Sep. 2010, pp. 127–151.

[12] C. Mastroianni, M. Meo, and G. Papuzzo, "Probabilistic consolidation of virtual machines in self-organizing cloud data centers," *IEEE Trans. Cloud Comput.*, vol. 1, no. 2, pp. 215–228, Jul./Dec. 2013.

[13] F. Kong and X. Liu, "A survey on green-energy-aware power management for datacenters," *ACM Comput. Surveys*, vol. 47, no. 2, pp. 1–38, 2014.

[14] Z. Liu, M. Lin, A. Wierman, S. Low, and L. L. H. Andrew, "Greening geographical load balancing," *IEEE/ACM Trans. Netw.*, vol. 23, no. 2, pp. 657–671, Apr. 2015.

[15] A. Qureshi, R. Weber, H. Balakrishnan, J. Guttag, and B. Maggs, "Cutting the electric bill for Internet-scale systems," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 123–134, 2009.

[16] D. Xu and X. Liu, "Geographic trough filling for Internet datacenters," in *Proc. IEEE INFOCOM*, Orlando, FL, USA, Mar. 2012, pp. 2881–2885.

[17] Y. Yao, L. Huang, A. B. Sharma, L. Golubchik, and M. J. Neely, "Power cost reduction in distributed data centers: A two-time-scale approach for delay tolerant workloads," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 1, pp. 200–211, Jan. 2014.

[18] Y. Guo, Z. Ding, Y. Fang, and D. Wu, "Cutting down electricity cost in Internet data centers by using energy storage," in *Proc. IEEE Glob. Telecommun. Conf. (GLOBECOM)*, Houston, TX, USA, Dec. 2011, pp. 1–5.

[19] L. Rao, X. Liu, L. Xie, and W. Liu, "Minimizing electricity cost: Optimization of distributed Internet data centers in a multi-electricity-market environment," in *Proc. INFOCOM*, 2010, pp. 1145–1153.

[20] H. Shao *et al.*, "Optimal load balancing and energy cost management for Internet data centers in deregulated electricity markets," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 10, pp. 2659–2669, Oct. 2014.

[21] D. Lučanin and I. Brandic, "Pervasive cloud controller for geotemporal inputs," *IEEE Trans. Cloud Comput.*, vol. 4, no. 2, pp. 180–195, Apr./Jun. 2016.

[22] L. Yu, T. Jiang, Y. Cao, and Q. Zhang, "Risk-constrained operation for Internet data centers in deregulated electricity markets," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 5, pp. 1306–1316, May 2014.

[23] J. Luo, L. Rao, and X. Liu, "Data center energy cost minimization: A spatio-temporal scheduling approach," in *Proc. IEEE INFOCOM*, Turin, Italy, Apr. 2013, pp. 340–344.

[24] W. Li, P. Svärd, J. Tordsson, and E. Elmroth, "Cost-optimal cloud service placement under dynamic pricing schemes," in *Proc. 6th IEEE/ACM Int. Conf. Util. Cloud Comput.*, Dresden, Germany, 2013, pp. 187–194.

[25] I. Goiri, K. Le, J. Guitart, J. Torres, and R. Bianchini, "Intelligent placement of datacenters for Internet services," in *Proc. 31st Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Minneapolis, MN, USA, Jun. 2011, pp. 131–142.

[26] J. Doyle, R. Shorten, and D. O'Mahony, "Stratus: Load balancing the cloud for carbon emissions control," *IEEE Trans. Cloud Comput.*, vol. 1, no. 1, pp. 116–128, Jan. 2013.

[27] A. Gupta, U. Mandal, P. Chowdhury, M. Tornatore, and B. Mukherjee, "Cost-efficient live VM migration based on varying electricity cost in optical cloud networks," *Photon Netw. Commun.*, vol. 30, no. 3, pp. 376–386, 2015.

[28] S. Ren, Y. He, and F. Xu, "Provably-efficient job scheduling for energy and fairness in geographically distributed data centers," in *Proc. IEEE 32nd Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2012, pp. 22–31.
[29] E. Kayaaslan, B. B. Cambazoglu, R. Blanco, F. P. Junqueira, and C. Aykanat, "Energy-price-driven query processing in multi-center Web search engines," in *Proc. 34th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval (SIGIR)*, Beijing, China, 2011, pp. 983–992.
[30] K. Le *et al.*, "Reducing electricity cost through virtual machine placement in high performance computing clouds," in *Proc. Int. Conf. High Perform. Comput. Netw. Stor. Anal.*, Seattle, WA, USA, 2011, p. 22.
[31] S. Akoush, R. Sohan, A. Rice, A. W. Moore, and A. Hopper, "Free lunch: Exploiting renewable energy for computing," in *Proc. 13th USENIX Conf. Hot Topics Oper. Syst. (HotOS)*, 2011, p. 17.
[32] E. Feller, L. Rilling, and C. Morin, "Snooze: A scalable and autonomic virtual machine management framework for private clouds," in *Proc. 12th IEEE/ACM Int. Symp. Clust. Cloud Grid Comput. (CCGrid)*, Ottawa, ON, Canada, May 2012, pp. 482–489.
[33] X. Xiang, C. Lin, F. Chen, and X. Chen, "Greening geo-distributed data centers by joint optimization of request routing and virtual machine scheduling," in *Proc. 7th IEEE/ACM Int. Conf. Util. Cloud Comput.*, London, U.K., Dec. 2014, pp. 1–10.
[34] H. Liu, H. Jin, C.-Z. Xu, and X. Liao, "Performance and energy modeling for live migration of virtual machines," *Clust. Comput.*, vol. 16, no. 2, pp. 249–264, 2013.

Agostino Forestiero received the Laurea and Ph.D. degrees in computer engineering from the University of Calabria, Italy, in 2002 and 2007, respectively. He has been a Researcher with the Institute of High Performance Computing and Networking, Italian National Research Council, ICAR-CNR, Cosenza, Italy, since 2010. He has co-authored over 50 papers published in international journals, among which the IEEE/ACM TON, the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, and ACM TAAS, and conference proceedings. His areas of interest are multi-agent systems, Cloud computing, P2P, bio-inspired algorithms, and cyber security. He is the Co-Founder of the Eco4Cloud Company.

Carlo Mastroianni (M'08) received the Laurea and Ph.D. degrees in computer engineering from the University of Calabria, Italy, in 1995 and 1999, respectively. He has been a Researcher with the Institute of High Performance Computing and Networking, Italian National Research Council, ICAR-CNR, Cosenza, Italy, since 2002. He was with the Computer Department, Prime Minister Office, Rome. He has co-authored over 100 papers published in international journals, among which IEEE/ACM TON, the IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, the IEEE TRANSACTIONS ON CLOUD COMPUTING, and ACM TAAS, and conference proceedings. He edited special issues for several journals such as the IEEE TRANSACTIONS ON CLOUD COMPUTING and *Future Generation Computer Systems*. His areas of interest are Cloud and grid computing, P2P, bio-inspired algorithms, Internet of Things, and smart cities. He is the Co-Founder of the Eco4Cloud Company.

Michela Meo (S'94–M'03) received the Laurea degree in electronic engineering and the Ph.D. degree in electronic and telecommunications engineering from the Politecnico di Torino, Italy, in 1993 and 1997, respectively. Since 2006, she has been a Professor with the Politecnico di Torino. She has co-authored about 200 papers and edited a book with Wiley and six special issues of international journals, including *ACM Monet*, *Performance Evaluation*, and *Computer Networks*. Her research interests include performance evaluation and modeling, green networking, and traffic classification and characterization. She chairs the Steering Committee of the IEEE OnlineGreenComm and the International Advisory Council of ITC. She is an Associate Editor of the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS—GREEN SERIES and was an Associate Editor of the IEEE TRANSACTIONS OF NETWORKING. She was a Program Co-Chair of several conferences among which ACM MSWiM, the IEEE Online GreenComm, the IEEE ISCC, the IEEE Infocom Miniconference, and ITC.

Giuseppe Papuzzo received the Laurea degree in computer engineering from the University of Calabria, Cosenza, Italy, in 2004. Since 2004, he has been in collaboration with the Institute of High Performance Computing and Networks, Italian National Research Council, ICAR-CNR, Cosenza, Italy. He has co-authored 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 the Co-Founder of the Eco4Cloud Company.

Mehdi Sheikhalishahi received the Ph.D. degree in energy efficient computing from the University of Calabria, Italy, in 2012. He has 12 years of experience in Web technology, network security, and distributed computing technologies. He is currently a Post-Doctoral Researcher with CREATE-NET, Italy. His main research interests are job scheduling, cloud, green computing, and applications of cloud computing in scientific disciplines. He has co-authored several papers on grid, cloud computing, and energy efficiency. In addition, he served as the Reviewer of several conferences and journals such as the *Journal of Optimization Methods and Software* and a Special Issue of the IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS on Many Task Computing.