Energy-aware joint management of networks and Cloud infrastructures

Bernardetta Addis [a], Danilo Ardagna [b], Antonio Capone [b,*], Giuliana Carello [b]

[a] *LORIA (UMR 7503 CNRS), Université de Lorraine, INRIA Nancy – Grand Est, France*
[b] *Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy*

1. Introduction

The debate on climate change and carbon dioxide emission reduction is fostering the development of new "green policies" aimed at decreasing the environmental impact of human activities. How to contrast global warming and how to enhance energy efficiency have been put on top of the list of the world global challenges. ICT plays a key role in this greening process. Since the beginning, ICT applications have been considered as part of the solution as they can greatly improve the environmental performance of all the

* Corresponding author. Tel.: +39 02 2399 3449; fax: +39 02 2399 3413.
 E-mail address: antonio.capone@polimi.it (A. Capone).

other sectors of the world economy. More recently, the awareness of the potential impact of the carbon emissions of the ICT sector itself has rapidly increased.

Overall, the combination of the energy consumption of service centers and communication networks accounts for 2–4% of global CO_2 emissions (comparable, e.g., to the emissions due to the global air traffic) and it is projected to reach up to 10% in 5–10 years, fueled by the expected massive adoption of Cloud computing [1,2]. Service centers investment grew by 22.1% during 2012 and it is expected it will further grow by another 14.5% in 2013 [1]. So, one of the main challenges for Cloud computing is to be able to reduce its carbon and energy footprints, while keeping up with the high growth rate of storage, server and communication infrastructures.

Even if computing and networking components of the system have been designed and managed quite independently so far, the current trend is to have them more strongly integrated for improving performance and efficiency of Cloud services offered to end users [3]. The integration of computing and networking components into a new generation of Cloud systems can be used not only to provide service flexibility to end users, but also to manage in a flexible way resources available in geographically-distributed computing centers and in the network interconnecting them.

A key enabler of Cloud/network cooperation is the use of geographically distributed service centers. Distributed Cloud service provisioning allows to better balance the traffic/computing workload and to bring computing and storage services closer to the end users, for a better application experience [4]. Moreover, from an energy point of view, having geographically distributed service centers allows Cloud providers to optimize energy consumption by exploiting load variations and energy cost variations over time in different locations.

The level of flexibility in the use of resources in different service centers depends on the application domains and basically comes from the geographic distribution of virtual machines hosting service applications, the use of dynamic geographically load redirect mechanisms, and the intelligent use of storage systems with data partitioning and replication techniques. This flexibility in service centers management also has a relevant impact on the communication network mainly for two reasons. First, since the service requests from users are delivered through an "access network" to service centers hosting the virtual machines, moving dynamically the workload among service centers can completely change the traffic pattern observed by the network. Second, an "interconnection network" is used to internally connect service centers and redirect end-user request or even move virtual machines and data with, again, a non-negligible impact on the traffic load when reconfiguration decisions are taken by the Cloud management system. Access and interconnection networks can be actually implemented in several different ways as private Intranets or using the public Internet, depending on the Cloud provider policies and the specific application domains. In any case, there is a large number of possible interaction models among telecommunications and Cloud providers that regulate their service agreements from a technical and economical perspective.

The main contribution of this paper is to explore the possibility of jointly and optimally managing service centers and the network connecting them, with the aim of reducing the Cloud energy cost and consumption. We develop an optimization framework that is able to show the potential savings that can be achieved with the joint management and to point out the relevant parameters that impact on the overall system performance. Concerning the network, we propose two representations, a high level approximated model and a detailed one. We show that the energy consumption obtained with the approximated version is very close to the one obtained with the detailed network representation and that the approximated version requires significantly smaller computational effort. Thus, it is suitable to assess the importance and impact of the joint optimization. The resource utilization and load allocation scheduling is performed on a daily basis assuming a central decision point (one of the service centers) and the availability of traffic patterns for different time periods. However, since the computational time is quite small (order of a few minutes), the time period length can be decreased, and the granularity can be finer, so as to follow better unpredictable traffic variation.

The approach proposed is based on a Mixed Integer Linear Programming (MILP) model which is solved to optimality with a state-of-the-art solver. The model assumes a Cloud service provider adopting the PaaS (Platform as a Service) approach and optimizes the load allocation to a set of geographically distributed Service Centers (SCs) where virtual machines (VMs) are assigned to physical servers in order to serve requests belonging to different classes. The goal is minimize the total energy cost consid-ering the time-varying nature of energy costs and the availability of green energy at different locations. The traf-fic can be routed to SCs using a geographical network whose capacity constraints and energy consumption are accounted for. We formally prove that the problem is NP-hard since it is equivalent to a Set Covering Problem.

We present a set of numerical results on a realistic case study that considers the real worldwide geographical distribution of SCs of a Cloud provider and the variations of energy cost and green energy availability in different world regions. Moreover, we show some results to characterize the scalability of the optimization framework proposed and the sensitivity to workload and green energy prediction errors.

Even if a large literature on the improvement of energy efficiency of service centers (see e.g. [5]) and communication networks (see e.g. [6]) exists, very few studies have considered so far the cooperation between Cloud and network providers for a joint energy management, and, to the best of our knowledge, no one has proposed a joint optimization framework able to exploit low energy modes in physical servers and networking devices (see Section 5 for a detailed analysis of related work). From a practical point of view, we assume that the economic advantages coming from the energy savings can be managed through service agreements between Cloud and network operators
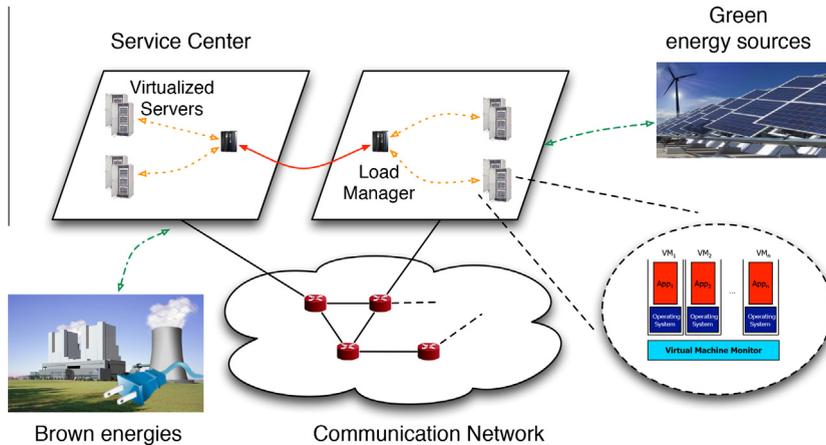
Fig. 1. Reference Cloud system architecture.

taking into account the contributions of different system components that can be quantified by the proposed model.

The paper is organized as follows. Section 2 describes the integrated approach and the problem addressed. Section 3 describes the proposed MILP models. Section 4 reports on the experimental tests and the obtained results. Section 5 overviews other literature approaches. Conclusions are finally drawn in Section 6.

2. The integrated framework for sustainable Clouds

In this work we consider a PaaS provider operating a virtualized service infrastructure comprising multiple Service Centers (SCs) distributed over multiple physical sites. This scenario is frequent nowadays. Indeed, Cloud providers own multiple geographically distributed SCs, each including thousands of physical servers. For example, Amazon offers EC2 services in nine worldwide regions (located in USA, South America, Europe, Australia, and Asia) and each region is further dispersed in different availability zones [7]. We denote with $\mathcal{N}$ the set of SCs. Physical sites are connected by a wide area network operated by a partner network provider (see Fig. 1). The network is modeled considering virtual paths connecting the SCs (a full mesh of end-to-end paths is assumed).

The PaaS provider supports the execution of multiple transactional services (e.g., e-commerce sites), each representing a different customer application. For the sake of simplicity, we assume that each VM hosts a single service, and multiple VMs implementing the same service can be run in parallel on different servers.

The hosted services can be heterogeneous with respect to computing demands, workload intensities, and bandwidth requirements. Services with different computing and workload profiles are categorized into independent request classes, where the system overall serves a set $\mathcal{K}$ of request classes.

Given the application workload profiles, the aim is to decide which SC serves each request, taking into account performance and energy as primary factors. As SC energy consumption is considered, the application workload can

be forwarded from one SC to another so as to exploit cheaper or green energy where it is available. However, this has an impact on the network energy consumption. The overall problem consists in assigning application workload to SCs in order to jointly minimize SCs and network energy costs.

The problem is solved on a daily basis: one day in advance, a prediction of the application workload is considered (as in [8–11]) and requests are assigned to service centers for the next 24 h. The considered 24 h horizon is discretized and divided into time bands. We denote with $\mathcal{T}$ the set of time bands. Each SC is characterized by a specific traffic profile for each time band: the local arrival rate for requests of class $k \in \mathcal{K}$ at the SC $i \in \mathcal{N}$ at time $t \in \mathcal{T}$ is denoted with λ_{ik}^t and we assume that the workload profile is periodic [12].[1]

Furthermore, as in [13–15] we assume that the SCs implement a distributed file system and data are replicated in multiple geographical locations for availability reasons. Each SC has a set of different types of VMs $\mathcal{L}$ that can serve incoming requests. Among the many resources, we focus on the CPU and bandwidth as representative resources for the resource allocation problem, consistently with [16–20]. Different request classes need different kinds of VMs and have different bandwidth and CPU requirements. Three parameters are used to model such features. Parameter m_{kl} is equal to 1 if request of class k can be served by a VM of type l, parameter b_k represents the bandwidth requirement of class k requests, while D_k represents the overall CPU demanding time [21] for serving a request of class k on a VM of capacity 1. VMs and SCs have capacity limits: P_{il} is the capacity of a VM of type l at the SC i, while C_i is the overall SC i computing capacity.

---

[1] It is worth noting that the computational times, discussed in Section 4, are such that even a finer granularity planning can be performed with our model. Besides, it is possible to solve the model several times during the day, thus taking into account unexpected traffic variations that can modify the predicted 24 h profile. However, the analysis of more fine grained time scales and their interrelationship with the one day ahead planning is out of the scope of this paper.

Energy consumption varies for different VMs. Furthermore, switching on or off a VM with respect to the previous time band also requires a certain amount of energy. Parameter α_{il} denotes the energy consumption for running a type l VM in SC i at peak load (in kWh), while η_{il} and θ_{il} are the energy consumption for turning on and off a VM, respectively.

In order to take into account service centers cooling energy costs, power distribution and uninterruptible power supply efficiency, we consider the Power Usage Effectiveness for SC i, denoted by ρ_i. The Power Usage Effectiveness is defined as the total service center power divided by the IT equipment power.

Concerning the network, we assume that it is possible to jointly optimize its energy consumption with that of service centers. In the case of multiple autonomous entities involved (like in the case of Cloud provider and one or more network providers) we assume that the optimal point we find is also a convenient solution for all of them. Other relevant scenarios where objectives of different entities are conflicting and the system can work on equilibrium points are out of the scope of this work.

As for the parameters, we assume that the energy consumption of Cloud service on an end-to-end path is zero if there is no traffic, while in the presence of traffic it has a fixed component and a load proportional component. In order to better estimate these components we consider the number of physical hops along the path and the energy consumption of routers interconnecting them (obviously the mapping of paths to physical links and routers depends on the network operator policies). We consider two network descriptions: a high level approximated model and a detailed one. In the approximated description paths are precomputed and we assume that the number of routers R_{ij} in the path connecting each pair of SCs i and j is given, which affects the energy consumption of connecting i to j. In the detailed representation, paths are the result of the optimization process. Furthermore, a maximum amount of traffic Q_{ij} can be sent from i to j. In fact a maximum bandwidth is available on the links belonging to the path connecting i and j, which cannot handle additional traffic when at full capacity. The energy consumption, in terms of kW h, for running and keeping idle a single router are denoted with γ_{ij} and δ_{ij}, respectively. Switching on and off a path consumes energy as well: τ_{ij} and ξ_{ij}. For each unit of energy consumed by the path connecting i to j at time t a cost f_{ij}^t must be paid. We point out here that this model is rather general and can be adapted to the energy profile of specific network devices. The assumption of a linear energy profile has been adopted in most of the previous literature on green networking and verified by a number of experimental results.

Renewable energy is considered as well, with cost and availability depending on the sites. We denote with c_i^t and g_i^t the cost for brown and green energy at each SC in the time band t, respectively. Γ_i^t is the renewable energy available at SC i at time t. As for the application workload, green energy sources can be evaluated by relying on prediction techniques [22–24].

The overall problem consists of assigning request classes to service centers, so as to guarantee that all requests

Table 1
Sets and parameters.

$\mathcal{N}$	Set of service centers
$\mathcal{T}$	Set of time bands
$\mathcal{L}$	Set of VM types
$\mathcal{K}$	Set of request classes
λ_{ik}^t	Local incoming workload arrival rate for request class k at SC i at time band t
m_{kl}	VM requirement parameter: equal to 1 if class $k \in \mathcal{K}$ can be served by type l VM, 0 otherwise
b_k	Bandwidth requirement for request class k
D_k	Demanding time for serving request class k on a VM of capacity 1
P_{il}	Capacity of a VM of type l at SC i
C_i	SC i overall computing capacity
$\overline{U}$	average VMs utilization
α_{il}	Energy consumption for running a type l VM in SC i
η_{il}	Energy consumption for switching on a type l VM in SC i
θ_{il}	Energy consumption for switching off a type l VM in SC i
ρ_i	PUE value for SC i
c_i^t	Cost for energy in SC i at time band t
R_{ij}	Total number of routers in the link (i,j)
Q_{ij}	Maximum bandwidth available on path (i,j)
γ_{ij}	Energy consumption for a router in path (i,j)
δ_{ij}	Energy consumption for a router in path (i,j) in idle state
τ_{ij}	Energy consumption for switching on a router in path (i,j)
ξ_{ij}	Energy consumption for switching off a router in path (i,j)
f_{ij}^t	Cost for energy in path (i,j)
w_{il}^0	Number of active VMs of type l in SC i at time 0
z_{ij}^0	Link (i,j) status at time 0
Γ_i^t	Green energy available in SC i at time band t
g_i^t	Cost for green energy in SC i at time band t

are served by a suitable VM by minimizing SCs and network energy costs, while not exceeding SCs and network capacities. The problem solution will provide the fraction of incoming workload served at each SC and redirected by the load managers to other SCs (see Fig. 1), and the number of VMs at each SC to be allocated to each application. The main idea behind the approach is that requests coming at SC i should be forwarded to another SC if the requests redirect is more efficient in terms of energy costs than serving the requests locally. Note that, in another SC energy costs might be lower thanks to time zone differences.

The problem parameters we will use in this paper are summarized in Table 1. We assume to solve the problem on a daily basis at a central decision point (any SC). In alternative the solution can be also computed in parallel at different SCs by sharing the incoming workload predic-tion without adding a significant overhead in the system (the information is shared once a day).

3. An optimization approach for energy management

The load management problem is formulated as a Mixed Integer Linear Programming (MILP) optimization model which takes into account many other important features, such as energy consumption, bandwidth and capacity constraints, and green energy generation. For the sake of clearness, we introduce first the approximated network representation.

3.1. Variables

The problem can be formulated using several sets of variables. Continuous nonnegative variable x_{ijkl}^t represents the arrival rate of requests of class k at SC i, which are served in SC j by a VM of type l, at time t. In other terms, x_{ijkl}^t variables represent the optimal configuration of request forwarding or in-site serving of the Cloud system in the time interval t. Note that x_{iikl}^t indicates the request rate that is originated and served locally, i.e., $\sum_{l \in \mathcal{L}} x_{iikl}^t \leqslant \lambda_{ik}^t$ and $\sum_{l \in \mathcal{L}} x_{iikl}^t = \lambda_{ik}^t$ if requests are not redirected.

Integer variable w_{il}^t represents the number of VMs of type l used in SC i at time band t. Two integer variables represent the number of VMs of type l to be turned on (and off) in each SC i at time band t : $\overline{w}_{il}^t$ (and respectively, $\underline{w}_{il}^t$).

Other variables are associated to the network: z_{ij}^t is a binary variable which is equal to 1 if the path connecting i to j is active at time t, 0 otherwise. Similarly to SCs, $\overline{z}_{ij}^t$ and $\underline{z}_{ij}^t$ indicate whether each path has to be turned off with respect to time $t - 1$.

Finally, continuous variable y_i^t models the amount of green energy used in SC i at time t. The decision variables are summarized in Table 2.

3.2. Objective function

As mentioned above, the aim of the problem is to reduce energy cost along the considered time horizon and to exploit green energy where available. The following objective function accounts for these goals.

$$
\min \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{N}} \left\{ c_i^t \left[\rho_i \sum_{l \in \mathcal{L}} (\alpha_{il} w_{il}^t + \eta_{il} \overline{w}_{il}^t + \theta_{il} \underline{w}_{il}^t) - y_i^t \right] + g_i^t y_i^t \right\}
$$
$$
+ \sum_{t \in \mathcal{T}} \sum_{i,j \in \mathcal{N}} f_{ij}^t R_{ij} \left[\delta_{ij} z_{ij}^t + \tau_{ij} \overline{z}_{ij}^t + \xi_{ij} \underline{z}_{ij}^t + (\gamma_{ij} - \delta_{ij}) \frac{\sum_{k \in \mathcal{K}} b_k \sum_{l \in \mathcal{L}} x_{ijkl}^t}{Q_{ij}} \right]
$$
$$
\tag{1}
$$

The total energy required at SC i at the time band t is represented by the term $\rho_i \sum_{l \in \mathcal{L}} (\alpha_{il} w_{il}^t + \eta_{il} \overline{w}_{il}^t + \theta_{il} \underline{w}_{il}^t)$, namely the sum of the energy needed to keep the VMs running and the energy needed to switch them on and off. The cost for the total energy is split into green and brown energy costs. The amount of green energy used is simply

Table 2
Decision variables.

x_{ijkl}^t	Arrival rate for class k request redirected from SC i to SC j served with a type l VM
w_{il}^t	Number of type l VMs running in SC i at time band t
$\overline{w}_{il}^t$	Number of VMs of type l turned on with respect to time $t - 1$ in SC i
$\underline{w}_{il}^t$	Number of VMs of type l turned off with respect to time $t - 1$ in SC i
z_{ij}^t	Whether the link (i, j) is active in time band t (binary)
$\overline{z}_{ij}^t$	Whether the link (i, j) has to be turned on with respect to time $t - 1$ (binary)
$\underline{z}_{ij}^t$	Whether the link (i, j) has to be turned off with respect to time $t - 1$ (binary)
y_i^t	Green energy used in SC i at time t

given by the variable y_i^t, while the brown energy is the difference between the total energy needed in the SC and the green energy actually used.

Concerning the network, the total amount of energy cost to be minimized is computed summing each path energy cost, which depends on the cost of keeping the routers working and the cost of turning on and off the routers. Besides, the cost of the energy consumed by the routers is considered, which is proportional to the total bandwidth in use, namely the fraction with respect to the total bandwidth available. Network is assumed not to use green energy. The energy costs of SC (first term of the objective function) and network (second term) are jointly minimized.

3.3. Constraints

Several constraints are needed to guarantee a proper description of the service centers and of the network behavior.

3.3.1. Workload assignment constraints

First of all, we need to guarantee that all the requests are served, either by the SC at which they occur or by another to which they are forwarded. Further, they must be served by a suitable class of VM. The following constraints guarantee that these requirements are satisfied.

$$
\sum_{l \in \mathcal{L}} \sum_{j \in \mathcal{N}} x_{ijkl}^t = \lambda_{ik}^t \qquad \forall i \in \mathcal{N}, \quad \forall k \in \mathcal{K}, \quad \forall t \in \mathcal{T} \tag{2}
$$

$$
x_{ijkl}^t \leqslant \lambda_{ik}^t m_{kl} \qquad \forall i, j \in \mathcal{N}, \quad \forall k \in \mathcal{K}, \quad \forall l \in \mathcal{L}, \quad \forall t \in \mathcal{T} \tag{3}
$$

In details, constraints (2) make sure that all the incoming traffic is served by any of the SC. Eq. (3) ensure that each type of request in each SC is served by the suitable type of VM.

3.3.2. Service center capacity constraints

The following capacity constraints are added to ensure that service centers capacity is not exceeded.

$$
w_{il}^t \geqslant \sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{K}} \frac{D_k x_{jikl}^t}{\overline{U}} \qquad \forall i \in \mathcal{N}, \quad \forall l \in \mathcal{L}, \quad \forall t \in \mathcal{T} \tag{4}
$$

$$
\sum_{l \in \mathcal{L}} P_{il} w_{il}^t \leqslant C_i \quad \forall i \in \mathcal{N} \tag{5}
$$

Constraints (4) determines the number of VMs of type l required to serve the overall incoming workload at site i. As in other literature approaches [8,25] and currently implemented by Cloud providers (see, e.g., AWS Elastic Beanstalk [26]), these constraints force a suitable number of VMs to be active in i in time slot t, guaranteeing that the average VM utilization is less or equal to a threshold $\overline{U}$. Note that $\sum_{j \in \mathcal{N}} \sum_{k \in \mathcal{K}} \frac{D_k x_{jikl}^t}{w_{il}^t}$ equals to VMs utilization under workload sharing. In practice $\overline{U}$ is usually set around 60%. Since w_{il}^t must be integer while the right hand side may be not, the constraints may not be tight, and therefore inequalities are needed. Finally, constraints (5) ensure that the number of VMs running at SC i in time t is lower than the available resources.

3.3.3. Network constraints

A network description must be included in the model, guaranteeing that the network capacity is not exceeded

and that the network energy consumption is properly computed.

$$\sum_{k \in \mathcal{K}} b_k \sum_{l \in \mathcal{L}} x^t_{ijkl} \leqslant Q_{ij} z^t_{ij} \qquad \forall i, j \in \mathcal{N}, \quad \forall t \in \mathcal{T} \tag{6}$$

$$z^t_{ij} = z^t_{ji} \qquad \forall i, j \in \mathcal{N}, \quad \forall t \in \mathcal{T} \tag{7}$$

Formula $\sum_{k \in \mathcal{K}} b_k \sum_{l \in \mathcal{L}} x^t_{ijkl}$ compute the portion of link bandwidth used for both directions of the traffic between SC i to SC j, thus determining the bandwidth used on each path of the network in each time slot. Constraints (6) ensure that traffic on each path does not exceed the available capacity and force z^t_{ij} to be one if the amount of used bandwidth on (i,j) is strictly positive, thus determining the active paths at time t.

Furthermore, constraints (7) guarantee that if a path is active in one direction, it is also active in the other, as paths are bidirectional.

Note that parameters Q_{ij} may not be the actual bandwidth along the path between i and j, but a reduced value that can guarantee that the capacity available for traffic load is just a fraction of the total one. This is one of the typical traffic engineering instruments adopted by network operators to control the quality of service experienced by traffic flows.

3.3.4. Time continuity constraints
VMs and paths switching on and off must be computed along the considered time horizon. The following constraints represent the relations between consecutive time bands.

$$\overline{w}^t_{il} \geqslant w^t_{il} - w^{t-1}_{il} \qquad \forall i \in \mathcal{N}, \quad \forall l \in \mathcal{L}, \quad \forall t \in \mathcal{T} \tag{8}$$

$$\underline{w}^t_{il} \geqslant w^{t-1}_{il} - w^t_{il} \qquad \forall i \in \mathcal{N}, \quad \forall l \in \mathcal{L}, \quad \forall t \in \mathcal{T} \tag{9}$$

$$\overline{z}^t_{il} \geqslant z^t_{il} - z^{t-1}_{il} \qquad \forall i \in \mathcal{N}, \quad \forall l \in \mathcal{L}, \quad \forall t \in \mathcal{T} \tag{10}$$

$$\underline{z}^t_{il} \geqslant z^{t-1}_{il} - z^t_{il} \qquad \forall i \in \mathcal{N}, \quad \forall l \in \mathcal{L}, \quad \forall t \in \mathcal{T} \tag{11}$$

Concerning SCs, (8) and (9) identify how many VMs have to be switched on or off on with respect to time band $t - 1$. Concerning the network, equations (10) and (11) define which paths have to be turned on or switched to idle mode with respect to the previous time band.

3.3.5. Green energy
The following constraints concerns the green energy consumed.

$$y^t_i \leqslant \rho_i \sum_{l \in \mathcal{L}} \left(\alpha_{il} w^t_{il} + \eta_{il} \overline{w}^t_{il} - \theta_{il} \underline{w}^t_{il} \right) \qquad \forall i \in \mathcal{N}, \quad \forall t \in \mathcal{T} \tag{12}$$

$$y^t_i \leqslant \Gamma^t_i \qquad \forall i \in \mathcal{N}, \quad \forall t \in \mathcal{T} \tag{13}$$

Eq. (12) ensures that the green energy used in each SC does not exceed the total energy needed. Eqs. (13) guarantee that the availability of green energy sources in each SC is not exceeded. The lower cost of green energy (we assume $g^t_i < c^t_i$ for all i and t) forces the optimal solution to prefer those SCs which have the possibility to exploit green energy produced in-site, and to use all the green energy available in a site before starting to use brown energy. Therefore, requests will be forwarded to those sites where green energy is available, provided that the network cost is lower than the savings that would be obtained.

3.3.6. Domain constrains
Finally, the model is completed by defining the variables domains.

$$x^t_{ijkl} \in \mathbb{R}^+ \quad \forall i,j \in \mathcal{N}, \qquad \forall k \in \mathcal{K}, \quad \forall l \in \mathcal{L}, \quad \forall t \in \mathcal{T} \tag{14}$$

$$w^t_{il}, \quad \overline{w}^t_{il}, \quad \underline{w}^t_{il} \in \mathbb{N} \qquad \forall i \in \mathcal{N}, \quad \forall l \in \mathcal{L}, \quad \forall t \in \mathcal{T} \tag{15}$$

$$z^t_{ij}, \quad \overline{z}^t_{il}, \quad \underline{z}^t_{il} \in \{0, 1\} \qquad \forall i,j \in \mathcal{N}, \forall t \in \mathcal{T} \tag{16}$$

$$y^t_i \in \mathbb{R}^+ \quad \forall i \in \mathcal{N}, \forall t \in \mathcal{T} \tag{17}$$

In Appendix B we show that this problem is NP-hard since it is equivalent to an Set Covering Problem.

3.4. Detailed network description model

In this section we describe the formulation of the detailed network representation. With such model it is possible to take into account other specific features, such as the delay introduced in the traffic load.

To represent the network connecting service centers, a set of router nodes $\mathcal{N}_\mathcal{R}$ is introduced. Therefore, in the graph representing the network, the set of nodes is $\mathcal{N} \cup \mathcal{N}_\mathcal{R}$. The network is not fully connected. It is represented by a set of links $\mathcal{A} \subset (\mathcal{N} \cup \mathcal{N}_\mathcal{R}) \times (\mathcal{N} \cup \mathcal{N}_\mathcal{R})$. Parameters Q_{ij}, γ_{ij}, δ_{ij}, τ_{ij}, ξ_{ij} and f^t_{ij}, which are associated with predefined paths in the former model, are now associated with single links.

Continuous routing variables φ^{odt}_{ij} are added to the model: ϕ^{odt}_{ij} represents the bandwidth required by the requests redirected from SC o to SC d, such that $o \neq d$, through routers i and j. Besides, variables z^t_{ij}, $\overline{z}^t_{ij}$, and $\underline{z}^t_{ij}$ are now associated with links in $\mathcal{A}$ instead of with paths. The domain of variables z^t_{ij}, $\overline{z}^t_{ij}$ and $\underline{z}^t_{ij}$ (16) is to be changed accordingly:

$$z^t_{ij}, \quad \overline{z}^t_{ij}, \quad \underline{z}^t_{il} \in \{0, 1\} \qquad \forall (i,j) \in \mathcal{A}, \quad \forall t \in \mathcal{T} \tag{18}$$

Objective function takes into account network costs, associated with links in set $\mathcal{A}$:

$$\min \quad \sum_{t \in \mathcal{T}} \sum_{i \in \mathcal{N}} \left\{ c^t_i \left[\rho_i \sum_{l \in \mathcal{L}} (\alpha_{il} w^t_{il} + \eta_{il} \overline{w}^t_{il} + \theta_{il} \underline{w}^t_{il}) - y^t_i \right] + g^t_i y^t_i \right\}$$
$$+ \sum_{t \in \mathcal{T}} \sum_{(i,j) \in \mathcal{A}} f^t_{ij} \left[\delta_{ij} z^t_{ij} + \tau_{ij} \overline{z}^t_{ij} + \xi_{ij} \underline{z}^t_{ij} + (\gamma_{ij} - \delta_{ij}) \frac{\sum_{o,d \in \mathcal{N}} \varphi^{odt}_{ij}}{Q_{ij}} \right] \tag{19}$$

Flow balancing constraints are added which ensure that the overall rate from o served by d is sent on the network from o to d:

$$\sum_{(i,j) \in \mathcal{A}} \varphi^{odt}_{ij} - \sum_{(j,i) \in \mathcal{A}} \varphi^{odt}_{ji} = 0, \quad \forall i \in \mathcal{N}_\mathcal{R}, \, o, \, d \in \mathcal{N} : o \neq d \tag{20}$$

$$\sum_{(i,j) \in \mathcal{A}} \varphi^{odt}_{ij} - \sum_{(j,i) \in \mathcal{A}} \varphi^{odt}_{ji}$$
$$= \begin{cases} \sum_{k \in \mathcal{K}, l \in \mathcal{L}} b_k x^t_{odkl}, & i = o \\ -\sum_{k \in \mathcal{K}, l \in \mathcal{L}} b_k x^t_{odkl}, & i = d, \quad \forall i \in \mathcal{N}, o, d \in \mathcal{N} : o \neq d \\ 0, & i \neq o, d \end{cases} \tag{21}$$

Constraints (20) concern nodes belonging to $\mathcal{N}_\mathcal{R}$. As such nodes are not associated with SC, they are neither origin nor destination of any flow. Therefore, all the flow entering in each of them must leave. Instead, constraints (21) concern SC nodes. For any two data centers o and d, such that $o \neq d$, the flow, related to request type k and VM type l, entering in node $i \in \mathcal{N}$ must be equal to the rate from o to d is $i = d$. As well, the flow leaving node i must be equal to the rate redirected from o to d is $i = o$. If $i \neq o, d$, the overall flow entering in i, associated with o and d, must leave node i. Besides, constraints (6) and (7) must be replaced to take into account the activation of each link in $\mathcal{A}$:

$$\sum_{o,d \in \mathcal{N}} \varphi_{ij}^{odt} \leqslant Q_{ij} z_{ij}^t, \qquad \forall (i,j) \in \mathcal{A}, \quad \forall t \in \mathcal{T} \tag{22}$$

$$z_{ij}^t = z_{ji}^t, \qquad \forall (i,j) \in \mathcal{A}, \quad \forall t \in \mathcal{T} \tag{23}$$

The model describing not fully connected network among data center is therefore:

min (19)
$s.t.$ (2)−(5), (8)−(15), (20)−(23), (17), (18).

Besides the network connecting routers and service centers, other features of the problem can be modeled, such as delay, by setting suitable limits to network performance. For instance, let us consider delay due to the network. It can be represented by assigning a delay μ_{ij} to each link $(i,j) \in \mathcal{A}$. A maximum delay $\bar{\mu}$ is set by adding the following constraints:

$$\sum_{(i,j) \in \mathcal{A}} \mu_{ij} \varphi_{ij}^{odt} \leqslant \bar{\mu}, \qquad \forall o, d \in \mathcal{N}, \quad \forall t \in \mathcal{T}$$

4. Experimental analysis

Our resource management model has been evaluated under a variety of systems and workload configurations. Section 4.1 presents the settings for our experiments. Section 4.2 reports a comparison between the two MILP models proposed. Section 4.3 reports on the scalability results we achieved while using a commercial MILP solver. Section 4.4 presents a cost-benefit evaluation of our solution, which is compared to the case where resource allocation is performed locally and independently at each SC without workload redirection and green energy is not available.

4.1. Experimental settings

Our approach is evaluated considering a large set of randomly generated instances. The model parameters were randomly generated uniformly in the ranges reported in Table 3.

The number of SCs varies between 9 and 36 considering a world wide distribution, while the number of request classes varies between 120 and 600. In particular SCs are located in four macro-regions: West USA, East USA, Europe, and Asia, where most of the SCs belonging to each area have the same time band. In order to evaluate the impact of interaction among different geographical areas, each instance includes at least one SC in several macroregions. SCs location has an impact on the energy costs and on the availability of green energy sources. This latter also depends on the size of the SC if solar energy is considered. According to [27] we consider SCs with size in the range 300–10,000 sqm. Furthermore, the overhead of the cooling system is included in energy costs by considering service centers PUE according to the values reported in [2].

For what concerns applications incoming workload, we consider 24 time bands and we have built synthetic workloads from the trace of requests registered at the Web site of a large University in Italy. The trace contains the number of sessions registered over 100 physical servers for a one year period in 2006, on a hourly basis. Realistic workloads are built assuming that the request arrivals follow non-homogeneous Poisson processes and extracting the request traces corresponding to the days with the highest workloads experienced during the observation periods. Some noise is also added, as in [28,29,11]. Fig. 2 shows a representative instance of the normalized local arrival rate of a given class.

In order to vary also the applications peak workload we consider the total number of Internet users [30] for each world region where SCs are located and set the λ_{ik}^t peak values proportionally to this number. Overall, the peak values vary between 100 and 1000 req/s.

We considered 9 types of VMs varying their capacity in the range between 1 and 20, according to current IaaS/PaaS provider offers [7,31].

The computing performance parameters of the applications have been randomly generated uniformly in the ranges reported in Table 3 as in [32,29,11,28]. Applications network requirements have been generated considering the SPECWeb2009 industry benchmark results [33].

The switching energy consumption of VMs has been evaluated as in [34], while VMs power consumption has been generated according to the results reported for the SPECVint industry benchmark [35], considering the peak power consumption of the reference physical servers and averaging the power consumption among the VMs running during a test.

The overall SC capacity has been determined varying the number of physical servers between 5,000 and 16,000 and assuming a 1:8 ratio for the physical to virtual resources assignment (i.e., 1 physical core is assigned to 8

Table 3
Performance and energy parameters.

b_k [10, 210] KB	D_k [0.1, 1] s	α_{il} [60, 90] W h	η_{il} [2, 3] W h	θ_{il} [0.28,14.11] W h
γ_{ij} 3.84 KW h	δ_{ij} 0.768 KW h	τ_{ij} 0.128 KW h	ξ_{ij} 0.128 KW h	ρ_i [1.2, 1.7]
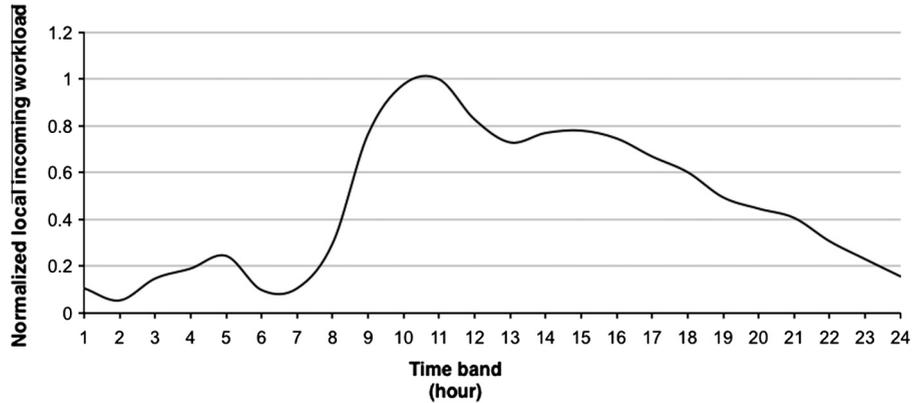
Fig. 2. Normalized local arrival rate over a day (UTC+1).

virtual cores of equal capacity) according to SPECVint recent results [35].

For what concerns the wide area network connecting remote SCs, paths capacity varies between 0.5 and 1 Gbps. A typical path connecting SCs is build up both by physical lines (such as optical fibers) and network components (such as routers and switches). Thus, we estimate the energy consumption of each path as the energy consumed by its routers, proportionally to the bandwidth in use. We have estimated the number of routers connecting different world regions through the traceroute application. Results (reported in Table 11 of Appendix B) show that, the number of hops within the same region is often one, while the largest number (17), is achieved when connecting Europe with West USA and Asia with East USA.

Due to the difficulty in obtaining detailed energy consumption profiles of commercial routers and the heterogeneity of devices used by network providers, we have considered a simplified scenario taking into account a single reference router. The model considered is the Juniper E320 [36], which is quite popular and whose power parameters are publicly available and reported in Table 3. Obviously, the model proposed is general and results can be easily extended to other router energy profiles.

Finally, from the cost point of view, we assumed that for each path (i, j) energy cost is equal to the average energy cost between service centers i and j.

The cost of energy per MW h varies between 10 and 65 Euro/MW h. Energy costs data have been obtained by considering several energy Market Managers responsible for those country in which SCs are located, taking into account also the energy price variability depending on the time of the day. The list of energy Market Managers we have considered is reported in Table 12 of Appendix B; for China, where there is no data disclosure on energy prices we have considered an unofficial source [37]. Fig. 3 shows the energy cost trend in the four geographical macro-regions with respect to time zone UTC+1.

Regarding green energies, we have considered a subset of SCs able to produce green energy or close to clean energy facilities. For example, Google and other Cloud market leaders have launched initiatives aiming at increasing the amount of electricity derived from renewable sources, rather than relying on electricity produced only with coal [38]. In [39], Google stated that energy coming from renewable sources represents the 25% of the overall Google's electricity use in 2010 and it is argued that this percentage could grow up to 35% in 2012.

According to [40–42], we have derived the availability of renewable energy sources (namely, solar, wind, and geothermal energy), in the four macro-regions. Details are reported in Fig. 4.

Finally, for what concerns green energy costs, we assume that energy is produced locally within the SC, so
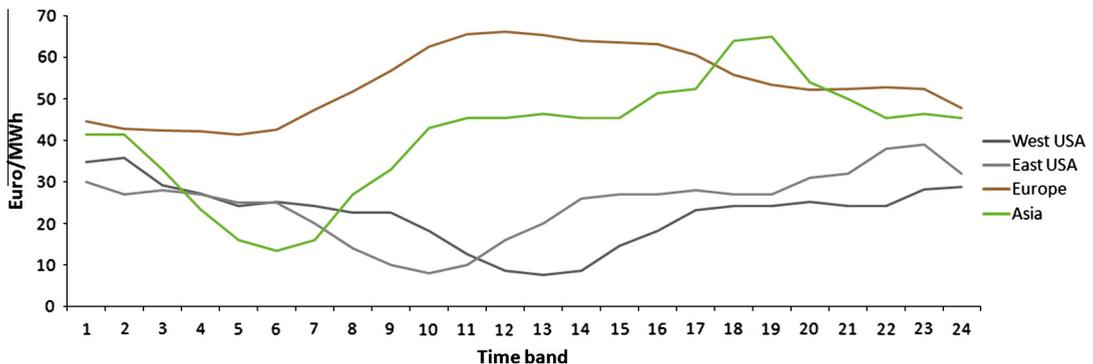
Fig. 3. Service centers energy cost during each time band (UTC+1).

Fig. 4. Green SCs location.

that the only cost to be considered is the capital cost required to realize the facilities. Consequently, parameters g_i^t are set to zero.

All tests were run on an Intel Nehalem dual socket quad-core CPUs @2.4 GHz with 24 GB of RAM running Ubuntu Linux 2011.4. CPLEX 12.2 was use as MILP optimization solver [43]. CPLEX was installed within a VirtualBox VM running Ubuntu Linux 2011.4 with 8 physical cores dedicated and 20 GB of memory reserved.

4.2. Approximated vs. detailed models comparison

In order to evaluate the impact of detailed network modeling, we consider a small example with a 4 service centers, including Atlanta, Mountain View, Pleasanton, and Chicago. Green energy is not available. We consider a network from the SNDLib [44] Library, Nobel US. The Nobel US network has 14 nodes. We integrate the Nobel US network with the 4 service centers. We collapse two service center nodes with two close router nodes belonging to $\mathcal{N_R}$. We added the remaining two service centers, and connected each of them to three close router nodes. Therefore the example has $|\mathcal{N_R}| = 12$ and $|\mathcal{N}| = 4$. The set of nodes and arcs are reported in Table 4 and 5. Each link in $\mathcal{A}$ has one router. All the other parameters have been derived in the same way as for previous examples. Results are reported in Table 6.

The model with the approximate network representation is solved to optimality in less than one second. Instead, the full network description model is not solved to optimality within one hour. The provided optimality gap, computed as $\frac{UB-LB}{LB}$, where UB is the best feasible solution found, and LB is the best relaxation found, is 3.6%.

Although the impact of network energy cost on the overall consumption is greater if the network is fully described, the increase is balanced by a decreasing in the SC consumption, resulting in a difference between objective functions for the two cases of about 0.5%. Therefore, the network approximate model can be profitably used

Table 4
Set of nodes.

DC-Mountain View
DC-Pleasanton
DC-Chicago
DC-Atlanta
(Palo-Alto merged with DC-Mountain View)
San-Diego
Boulder
Washington
(Atlanta merged with DC-Atlanta)
Urbana–Champaign
Ann-Arbor
Lincoln
Princeton
Ithaca
Pittsburgh
Houston
Salt-Lake-City
Seattle

to evaluate the overall impact of redirecting, as it provides reliable insights in negligible computational time. Besides, network cost is not increasing when including a full network description.

If green energy is assumed to be available in two service centers the problem with full network description becomes more time consuming, and the optimality gap rises up to about 13% with a one hour time limit.

The approximated model is based on precomputed paths, which are usually applied to obtain heuristic solutions when enumerating all the possible paths is too expensive [45]. The approximated model allows to achieve a good trade-off between cost accuracy and optimization time required for managing jointly the network and multiple distributed service centers, as the difference between the global optimal solution of the detailed network model and the approximated one is around 0.5%, which is a good approximation for any heuristic approach. Thus, in the following we will only present the results of the approximated model.

Table 5
Set of arcs.

DC-Mountain View	San-Diego
DC-Mountain View	Salt-Lake-City
DC-Mountain View	Seattle
San-Diego	Houston
San-Diego	Seattle
Boulder	Lincoln
Boulder	Houston
Boulder	Salt-Lake-City
Washington	Princeton
Washington	Ithaca
Washington	Houston
DC-Atlanta	Pittsburgh
DC-Atlanta	Houston
Urbana–Champaign	Lincoln
Urbana–Champaign	Pittsburgh
Urbana–Champaign	Seattle
Ann-Arbor	Princeton
Ann-Arbor	Ithaca
Ann-Arbor	Salt-Lake-City
Princeton	Pittsburgh
Ithaca	Pittsburgh
DC-Pleasanton	Seattle
DC-Pleasanton	DC-Mountain View
DC-Pleasanton	San-Diego
DC-Chicago	Ann-Arbor
DC-Chicago	Pittsburgh
DC-Chicago	Washington

Table 7
CPLEX solver execution time.

Test set	$|\mathcal{N}|$	$|\mathcal{K}|$	Exe time (s)
S1	11	150	28.40
	11	300	69.29
	11	450	90.20
	11	600	135.63
S2	10	300	58.10
	15	300	124.84
	20	300	177.46
	25	300	385.29
S3	9	120	15.37
	17	180	38.60
	24	300	115.51
S4	11	150	28.29
	19	300	88.66

SCs in about seven minutes. The size of the solved instances is limited by the available memory of the hosting VirtualBox VM. Therefore, our MILP-based approach is suitable for the one day ahead planning. Besides, it could be applied to solve the problem several times during one day, so as to exploit updated data. In fact, thanks to the small computational effort, the approach can be extended quite easily to scenarios where load predictions on a 24 h horizon are not reliable and predictions must be adjusted according to real time traffic measurements. The proposed approach can be modified using a sliding time window (24 h or shorter) that is shifted ahead every time a new load prediction is available. At each update the optimization is re-executed. Depending on the accuracy of the traffic values, also the time granularity of the window can be modified. However, such analysis and use of our approach is out of the scope of this paper.

4.3. Scalability analysis

In order to evaluate the scalability of commercial MILP solvers in dealing with our approximated optimization model, we performed a large set of experiments.

We generated four sets of randomly generated instances. The rational was to include several time zones in the analysis and to be representative of real SCs location around the globe according to the main Cloud providers practice (see also the following section).

In the first set S1, the service centers number and location is fixed in the four macro-regions, namely West America, East America, Europe, Asia, and the scalability of the solver varying the number of classes is investigated. The set S2, vice versa, investigates the scalability of our approach varying the number of SCs in East America and Europe, while keeping constant the number of classes. Finally, in sets S3 and S4 the number of SCs and the number of classes are jointly varied by considering three (i.e., West America, East America, and Asia) and the four macro-regions.

Table 7 reports the average optimization time required by CPLEX. The average values reported in this Section have always been computed by considering 35 instances with the same size. Results show that CPLEX is very efficient and can solve problem instances up to 600 classes and 25

4.4. A case study

In this section we perform a cost-benefit evaluation of our solution with respect to a *base scenario* in which the system cannot exploit the network infrastructure to move requests between sites, which is the approach currently used for Cloud resource allocation. Analyses evaluate the costs savings that can be achieved through load redirection, exploiting energy costs variability and green energy availability in multiple locations.

Furthermore, we also evaluate the greenhouse gas emission reduction with respect to a *brown scenario* where no green energy is available. CO_2 emission has been computed through the U.S. Environmental Protection Agency web tool [46].

In the following, we consider a case study inspired by the Google infrastructure where we also varied the number of searches to be performed per day. Although Google

Table 6
Comparison between full and approximate network representations.

Network representation	DC energy consumption			Network energy consumption		
	Overall	Computing	Switching	Overall	Fixed and switching	Flow
Approximate	15.62	15.17	0.45	2.49	1.52	0.83
Full	14.11	13.80	0.31	3.92	2.45	1.38

tends to be quite secretive about its SCs technology, the geographical location of its SCs is known. Google owns 36 SCs spread all over the world [47], in order to have a huge amount of computational resources to satisfy the tens of billions of user requests per day. SCs have been setup near large urban areas close to the end-users in order to reduce network latency. Fig. 13 and Table 13 in Appendix B details where SCs are placed.

For what concerns SCs capacity, since no specific data is available but only aggregated values have been undisclosed [48], we assume that the servers are almost spread uniformly among SCs (a Gaussian random deviation from the mean for each SC is added).

For the PUE value, [49] reports an average value close to 1.8. However, considering SCs geographic location, we can adopt a lower value of PUE for those SCs placed in regions with suitable climatic conditions, like the ones placed in North Europe. In fact, in most moderate climates with temperatures lower than 13 °C for 3000 or more hours per year, new cooling techniques based on free air can eliminate the majority of chillers runtime. For example, the SC in Belgium eliminates chillers, running on free cooling 100% of the time, and reaching in this way a PUE value close to 1.2 [49]. Fig. 14 in Appendix B shows the PUE value assumed for each SC.

Finally, in order to estimate the total amount of green energy produced by each SC during a single day (parameters Γ_i^t), we assumed that the green energy was proportional to the SC area. The values we used are reported in Appendix B by Fig. 15, while details on SC location and time zone are reported in Table 13.

Fig. 5 shows the cost savings that can be achieved by our solution with respect to the base scenario varying the number of searches to be performed per day. The savings are very significant, ranging between 40% and 56% for the lightest loaded instance. This can be expected, since only a small set of SCs can rely on green energy. Our model forwards as many requests as possible to these sites, until the VMs utilization threshold is reached or the upstream bandwidth is saturated. Additional requests cannot be served and need to be routed elsewhere, possibly to a nongreen-enabled SC. Therefore, saving are smaller when the overall workload is higher. This argument is confirmed also by the data reported in Figs. 6 and 7, which show how the energy consumption is split among the network and the SCs. Whilst the network energy consumption fraction is almost constant and around 15–20%, the test cases with a larger number of daily requests are characterized by a higher usage of brown energy. As it can be expected, with a larger amount of green energy available, a Cloud provider would experience not only savings (due to the lower cost assumed for green energy), but also massive reduction of CO_2 emission, as will be further discussed in the following. It is worth noticing that for the heaviest load instance the
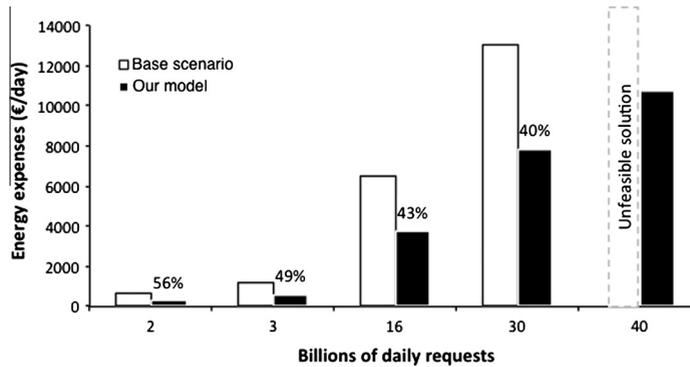
Fig. 5. Overall cost comparison with respect to the base scenario.
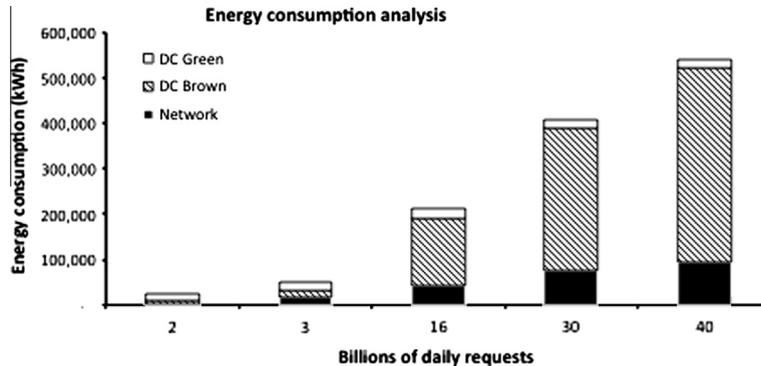
Fig. 6. Energy split for our model.
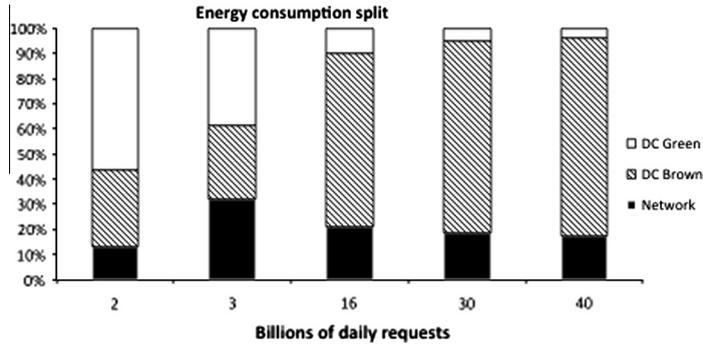
Fig. 7. Percentage energy split for our model.

base scenario cannot find a feasible solution. Hence, the requests redirection allows also to exploit remote SCs available capacity when local resources are saturated.

While our approach can achieve lower cost with respect to the base scenario, it may cause a larger energy consumption. As shown in Fig. 8, this is due to the network transfers and routers crossed along the path between two sites. However, our model especially fosters the maximum possible amount of "clean" energy usage, unlike the base scenario, which is limited in relying on local energy only. This consideration is confirmed by Fig. 9, which reports the percentage of green energy used in our model and in the base scenario. The plot clearly shows how our model performs better on all accounts, i.e., it is able to exploit as much green energy as possible by forwarding the requests to the green SCs.

Furthermore, as a representative example, Fig. 10 shows the number of active servers for each time band,
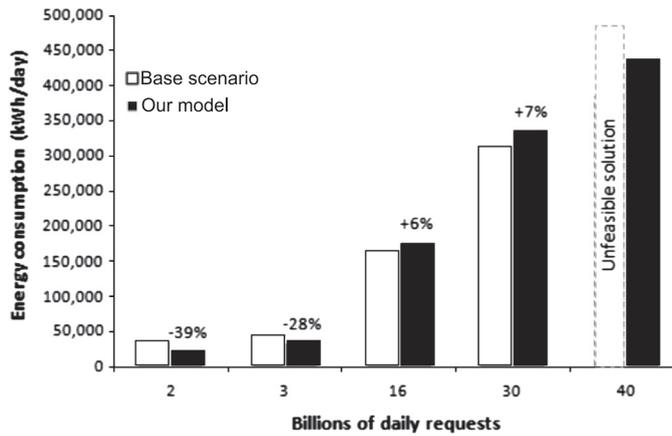
Fig. 8. Energy consumption comparison with the base scenario.
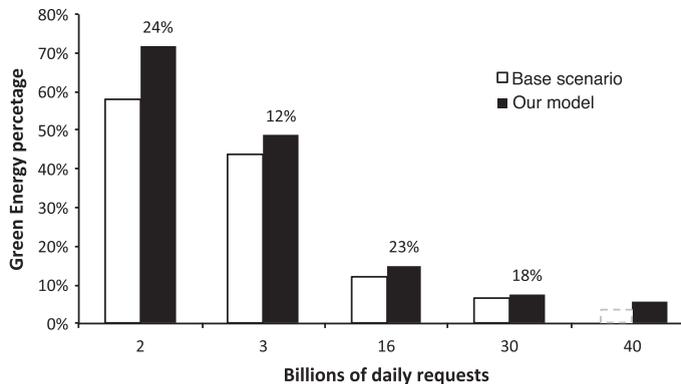
Fig. 9. Percentage of green energy usage.
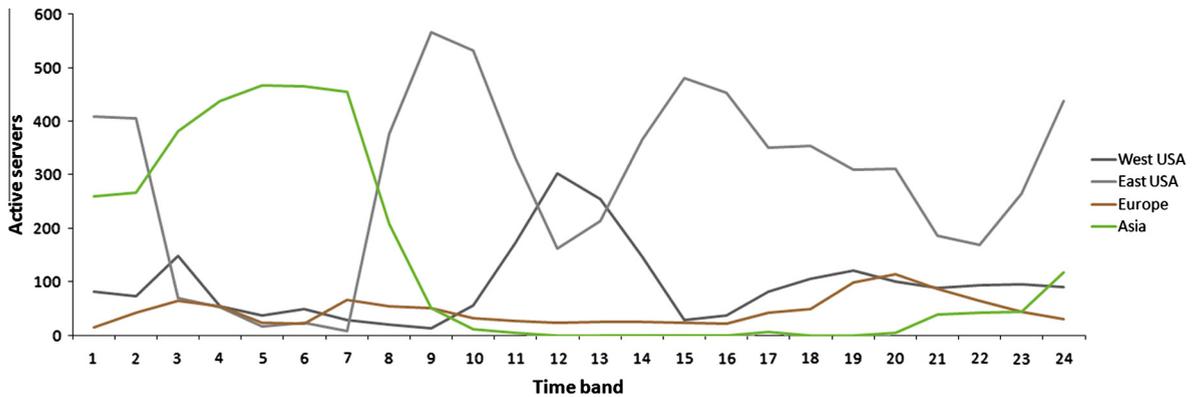
Fig. 10. Number of active servers for each time band (UTC+1).

distinguishing the four macro-regions where SCs are located, while serving 30 billion requests a day. As it could be expected, the trend of each macro-region appears opposite to the energy cost, presented in Fig. 3: the number of active servers increases when the energy cost decreases, and vice versa. For example, requests are executed in Asia only in the first time bands, where in fact the energy cost has a minimum, while in most of the other time bands the majority of the requests are forwarded to East and West USA, since the energy cost appears always lower with respect to the other areas. However, since also the network costs and the network and SC capacities have to be taken into account, a small amount of requests is served locally also in areas where the energy cost is not optimal.

An interesting comparison can also be made between our model and the brown scenario (see Fig. 11, where the overall energy consumption is reported). When the workload is light, the energy consumption of our model is higher, while being almost equal in the other cases. It has to be reminded that, for our model, more than 50% of the energy consumed in lightly loaded instances comes from green sources, thus the CO_2 emission results to be lower, as will be discussed in the following. The main explanation for a higher energy consumption is that, considering in particular the 2 billion requests per day scenario, our model consumes more energy for the network, since it

has to forward more requests (in percentage) to remote SCs to exploit green energy sources. In the other instances, this problem is marginal, since all the green energy is saturated. Requests are forwarded according to the same criteria as the brown model, thus consuming the same amount of energy.

Finally, Fig. 12, reports the CO_2 emissions of the brown scenario and of our model. The maximum absolute reduction (around 25 tons of CO_2) occurs in the 40 billion requests per day instance. Our model is able to reduce up to 57% the environment pollution derived from greenhouse gases for the lightest loaded scenario, while this percentage reduces with the workload, advocating a larger adoption of green energy sources for the largest Cloud providers.

4.5. Sensitivity analysis for the case study

We performed some tests to evaluate the impact on the problem solution of an error in the prediction of the workload or the amount of green energy produced.

We consider a total workload around 2 billions requests/day (λ^*), which corresponds to the worst case scenario for the green energy sensitivity (recall from Fig. 9 that most of the energy consumed by SCs is green in that case).
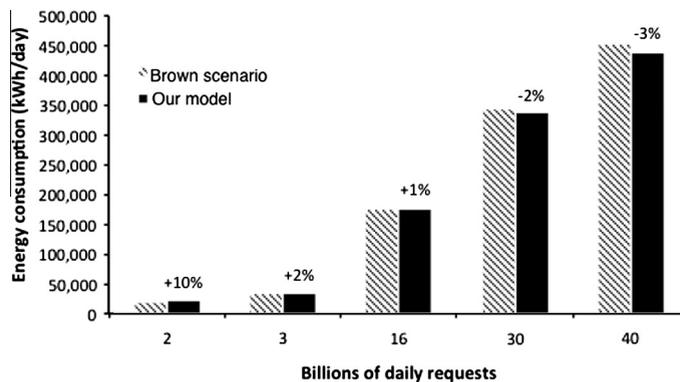
Fig. 11. Comparison of the energy consumption of our model with the brown scenario.
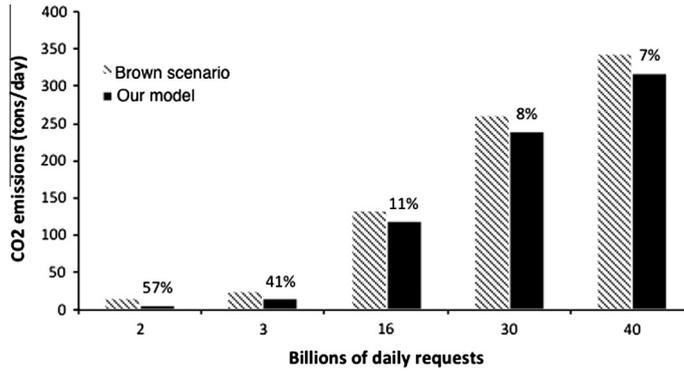
Fig. 12. CO_2 gas emissions comparison with respect to the brown scenario.
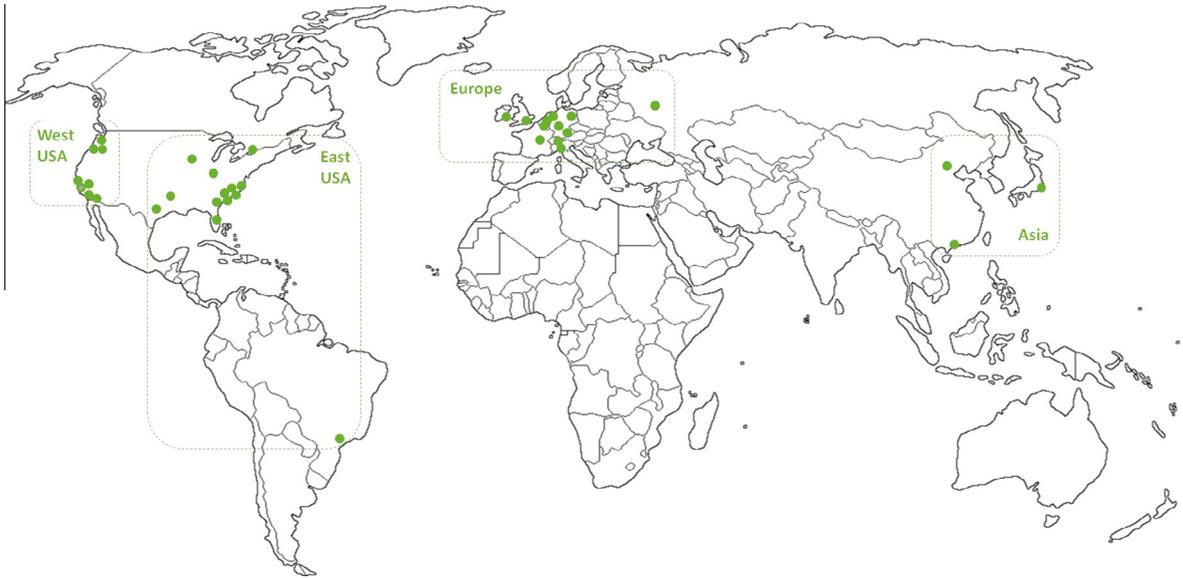
Fig. 13. Map of Google SCs location considered in our model.
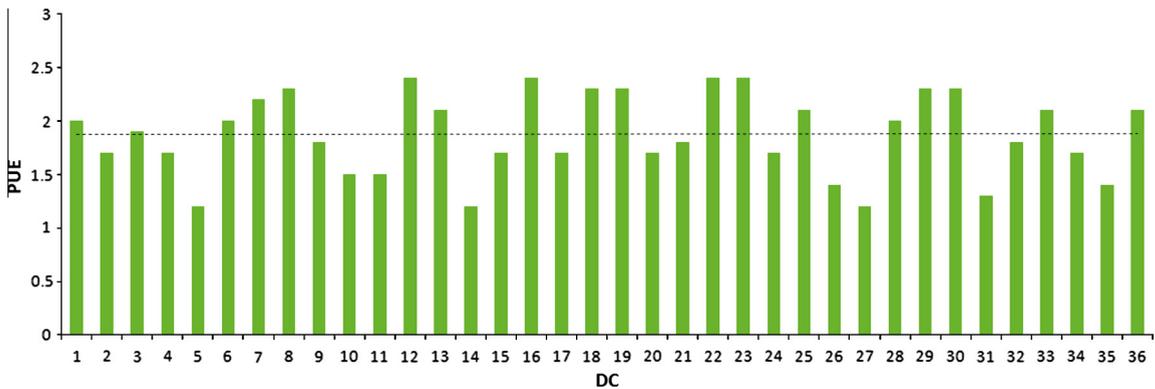
Fig. 14. Google SC PUE values considered in our analyses.

Let us first consider the impact of a variation of work-load estimate. As in [29], we solve the model cost with one hour of time limit for a traffic equal to λ^*, and we compare it with the solution obtained within the same time limit when the amount of workload is perturbed. Workloads of all SCs are reduced (or increased) by the
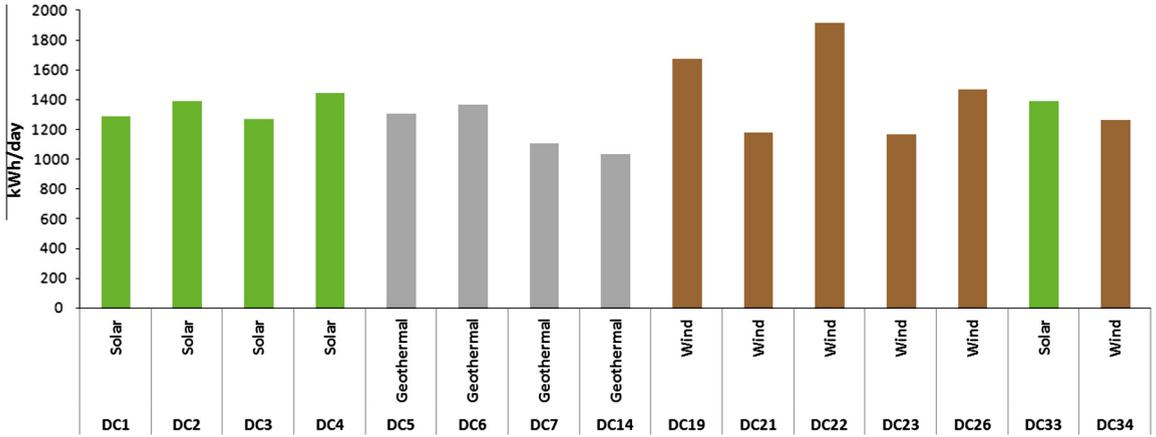
Fig. 15. Total kW h produced by green Google Service Centers during a single day.

same parameter ζ. For instance, if we consider $\zeta = 1.1$ the perturbed workload associated to SC i, time slot t and request type k is $1.1\lambda_{ik}^{*t}$. Two values of ζ are considered: 0.9 and 1.1.

The ratio between the objective function associated with the perturbed workload and the original one shows that the variation of the workload has a small impact on objective function variation. The variation is almost linearly proportional to the perturbation. In fact, as the number of VMs is large, it behaves almost as a continuous variable, and therefore VMs cost varies linearly with the workload. Besides, the link switching costs are the only stepwise costs, and they are usually smaller than the VMs costs. Therefore, the variation of the objective function depends in a linear way on the workload variation, unless one link is switched on or off when the workload changes.

We also considered the case where workloads are perturbed independently in a random way, with their value uniform in the interval $\pm10\%$ around the predicted level. We compared the optimal solution value obtained solving the model with the nominal workloads and the values obtained according to the following procedure. We compute the optimal solution with the nominal workload, then we fix the values of the routing and network variables (i.e., the maximum traffic between any pair of service centers and the network activation variables) and recompute the objective function considering the perturbed workloads. This comparison scenario corresponds to the practical case where the load balancing between service centers is fixed according the optimal solution computed with the predicted workloads and any unexpected variation of the workload is handled locally in each service center reducing or increasing the local load. Results show that the difference is always below 4% over 100 randomly generated scenarios.

Then we evaluated the impact of an error in the prediction of green energy availability. We consider the conservative case in which the available green energy is smaller than the forecasted one and the percentage error is the same for all SCs provided with green energy generation. Starting with a known solution, we reduce the amount of

Table 8
Impact of green availability prediction error.

Green energy reduction (%)	Objective function increase (%)
100	68.98
50	41.35
25	24.00

green energy available in each SC. Then, keeping the loads for each SC as they are in the starting solution, we compute the increase of brown energy use, if reduced green energy available in a SC in not enough to deal with SC load. Due to such increase, we compute the increase in the objective function. The percentage objective function increase is reported in Table 8 for three values of percentage prediction error: it is the same for all the SCs and it is equal to 100%, 50% and 25%.

Results show that, although we consider a very conservative case in which the prediction error affects simultaneously all the SC, and green energy availability is never greater than the forecasted value, the impact of the overall energy cost is limited.

5. Related work

As discussed before, for many years energy management of Cloud systems has been studied considering separately the service centers and the network components.

From the service center side, three main approaches have been developed: (i) *control theoretic feedback loop techniques*, (ii) *adaptive machine learning approaches*, and (iii) *utility-based optimization techniques*.

A main advantage of a control theoretic feedback loop is system stability guarantees. Upon workload changes, these techniques can also accurately model transient behavior and adjust system configurations within a transitory period, which can be fixed at design time. A previous study [28] implemented a limited lookahead controller that determines servers in the active state, their operating frequency, and the allocation of VMs to physical servers. However, this implementation considers the VM

placement and capacity allocation problems separately, and scalability of the proposed approach is not considered. A recent study [50,51] proposed hierarchical control solutions, particularly providing a cluster-level control architecture that coordinates multiple server power controllers within a virtualized server cluster [50]. The higher layer controller determines capacity allocation and VM migration within a cluster, while the inner controllers determine the power level of individual servers.

Machine learning techniques are based on live system learning sessions, without a need for analytical models of applications and the underlying infrastructure. A previous study [52] applied machine learning to coordinate multiple autonomic managers with different goals, integrating a performance manager with a power manager in order to satisfy performance constraints, while minimizing energy expenses exploiting server frequency scaling. Recent studies provide solutions for server provisioning and VM placement [53] and propose an overall framework for autonomic Cloud management [54]. An advantage of machine learning techniques is that they accurately capture system behavior without any explicit performance or traffic model and with little built-in system-specific knowledge. However, training sessions tend to extend over several hours [52], retraining is required for evolving workloads, and existing techniques are often restricted to separately applying actuation mechanisms to a limited set of managed applications.

Utility-based approaches have been introduced to optimize the degree of user satisfaction by expressing their goals in terms of user-level performance metrics. Typically, the system is modeled by means of a performance model embedded within an optimization framework. Optimization can provide global optimal solutions or sub-optimal solutions by means of heuristics, depending on the complexity of the optimization model. Optimization is typically applied to each one of the five problems separately. Some research studies address admission control for overload protection of servers [55]. Capacity allocation is typically viewed as a separate optimization activity which operates under the assumption that servers are protected from overload. VM placement recently has been widely studied [18,56,57]. A previous study [8] has presented a multi-layer and multiple time-scale solution for the management of virtualized systems, but the trade-off between performance and system costs is not considered. A hierarchical framework for maximizing Cloud provider profits, taking into account server provisioning and VM placement problems, has been similarly proposed [58], but only very small systems have been analyzed. The problem of VM placement within an IaaS provider has been considered [59], providing a novel stochastic performance model for estimating the time required for VMs startup. Finally, frameworks for the co-location of VMs into clusters based on an analysis of 24-h application workload profiles have been proposed [60,61], solving the server provisioning and capacity allocation problems.

Due to the increasing trend towards more communication intensive applications in data centers, the bandwidth between VMs is rapidly growing. These features raise new interest in the specific study of VMs migration

strategies, and in the joint optimization of the network underling the Cloud and the data center structure [62–65]. In [66] the authors propose a data center virtualization architecture called SecondNet. The central idea it is to allocate as a unit of resource a virtual data center, instead of single/groups of VMs. The authors propose a greedy algorithm based on clustering of servers to simplify the problem and reduce the dispersion of VM inside the real data center. The method first assign the set of VM to servers, then construct the communication network. In [67] authors propose dynamic programming algorithm for computing the optimal embedding on special structured networks, namely both network and requests topologies are trees. They showed that the problem of embedding it is still NP-hard, but proposed a specialized method to deal with this subproblem, representative of a large class of service and enterprise workloads.

All these works are devoted to consider the bandwidth usage within the Cloud, but in recent years the increase of Cloud dimensions and requests applications to take into account aroused the interest in considering geographical Clouds. In this context the consumption of the network between Clouds and the communication delays introduced by migration of VMs among different Clouds must be considered to manage efficiently the overall system. An energy-aware traffic engineering in the geographical network is the optimization instrument necessary for this purpose.

From the network side, research on improving energy efficiency has been attracting a growing attention during the last years [6], but a limited number of works have been devoted to energy-aware traffic engineering. Some recent studies evaluate the potentialities and the effective applicability of energy-aware routing procedures [68–74]. Main differences between considered approaches are related to the use of flow-based label-switching techniques, or shortest path routing. In [75] a MPLS (Multi Protocol Label Switching) framework is considered and traffic routing is optimized according to a multi-period schedule in order to switch off routers and/or their line cards for saving energy. The approach proposed in [76] aims at switching off the line-cards (network links) guaranteeing QoS constraints (maximum utilization and maximum path length constraints) in a scenario where an hybrid MPLS/OSPF (Open Shortest Path First) scheme is adopted. In [77] the authors describe some heuristics that, given a traffic matrix and a fully powered network, are able to switch off nodes and links while respecting traffic constraints. In [78] some on-line Energy-Aware Traffic Engineering (EATe) techniques are presented for optimizing links and routers power consumption, by considering their rate-dependent energy profiles. In [79], models and algorithms for setting link weighs (routing metric) used for shortest path calculation are used in order to optimize traffic and minimize the energy consumption.

To the best of our knowledge, very few approaches has provided an integrated framework for service centers and network, and no one has included an accurate model of the network energy management. On the other side, several recent papers have considered the problem of geographical load balancing with the aim to reduce energy

cost by exploiting differences of electricity price and weather conditions across different regions (see e.g.,[13–15]). In [80] the focus is on the impact of local temper-ature on the energy efficiency on the flexibility allowed by batch workload in shifting the energy consumption. The problem of interactive workload routing and capacity allo-cation for batch workload is formulated as a MILP model and a distributed algorithm is proposed. The interesting issue of the interaction between the smart grid and data centers is investigated in [81,82] where the different objec-tives of data centers interested in cost minimization and grid interested in power load balancing over time and space are taken into account.

The energy cost is a central point in managing Cloud system, however optimizing energy costs without taking into account energy consumption, can lead to the collateral consequence of increasing the amount of energy. For this reason, some new works are devoted and centered on greening the network consumption and reducing energy amount instead of reducing merely energy costs. In [83] the problem of managing the energy usage in data centers that can exploit on-site power production is considered and dynamic algorithms that are able to optimize the energy consumption and the use of energy from local sources and from the electric grid are defined. In [84,85] authors consider the problem of geographical load balanc-ing with the aim of utilize green energies. They propose a discrete-time model where the two decision taken are the amount of traffic routed from a given source and a given data center and the number of servers active at each data center. The delay is modeled as a network delay (depending on the couple source and data center) and a queueing delay depending on the traffic served by the data center and the number of active servers. The objective function takes into account energy consumption and total delays, the model is linearized leading to a continuous optimization model. In [86] authors present an optimization based framework to manage the consumption of energy in multi-data-center services. The framework manages to assign requests across data centers with the aim of reducing the total energy costs, but at the same time respecting Service Level Agreements (SLAs). The frame-work take into account energy costs and sources, and the possibility to introduce caps on brown energy. The model of the data centers and SLA is very simplified. Different decisions are taken at different time scales. At time scale of one year the proportion of green and brown energy used is decided; at time scale of one hour, distribution of requests with a simplified model that consider all requests of the same type. The resulting model is a non-linear optimization model and it is solved using a simulated annealing strategy with online re-optimization. The network connecting the data centers is not considered, data center capacity is evaluated by the algorithm, and after that serv-ers are managed as a consequence (servers are kept on to manage 20% more capacity then the one evaluated by the algorithm). In [87], the same authors modified their model to take into account the use of UPS batteries to manage dif-ferent costs of energy in different times of the day. In this work they do not consider any more brown vs. green energy, but use different costs of energy for different geographical regions and time of the day (cost data are drown from real life).

Differently from these previous papers, our work pro-vides a fully integrated management of service centers and communication network considering variable traffic, energy cost, and green energy availability. We proposed a new MILP model for the integrated problem and we show that it can be solved to the optimum in relatively short time (order of minutes) for realistic size instances without the need to rely on heuristic algorithms. Our extended model is the first one that model accurately the energy consumption of the network even in the presence of man-agement strategies that can put in sleep some parts of the network following load variations.

6. Conclusions and open issues

In this paper we propose a new optimization frame-work for the management of the energy usage in an inte-grated system for Cloud services that includes both service centers and communication networks for accessing and interconnecting them. The optimization framework considers a PaaS scenario where VMs serving an applica-tion can be allocated to a set of SCs geographically distrib-uted and traffic load coming from different world regions can be assigned to VMs in order to optimize the energy cost and minimize CO_2 emissions. We propose two repre-sentations of the network interconnecting SCs. We have shown that the approximate description provides a very close energy consumption value and therefore can be applied to provide very good approximate solution in rea-sonable computational time. We have shown that, despite its quite good level of detail in modeling the energy con-sumption of different system components and the compu-tation and bandwidth constraints, the proposed problem can be solved to optimality in relatively short time (order of minutes) with realistic size instances.

The numerical results presented in previous Sections provide interesting insight in the factors that influence optimal energy management policies. In particular, it is worth pointing out that the availability of green energy has a remarkable impact on the energy management and that in most cases exploiting clean energy comes at the cost of an higher global energy consumption mainly because of the impact of the communication network. We expect that in more complex scenarios than that con-sidered here, where also the impact on the network of VMs migration and data replication is considered, this effect may be even more evident.

We believe that the optimization instrument proposed in this paper is flexible and efficient enough to be used for exploring possible energy management policies in real Cloud systems beyond the results presented here.

Appendix A. Problem complexity

We prove that our approximated problem is NP-hard reducing the Set Covering Problem to it. A set of elements $M = \{1, \ldots, m\}$ and a set of subsets $\{N = \{N_1, \ldots, N_n\}$ $(N_j \subset M)$ are given. A *cover* is a set $F \subseteq N$, such that $\cup_{N_j \in F} N_j = M$. The decision version of Set Covering is defined as follows: Is there a cover F such that its cardinality is at most equal to a given threshold κ?

The decision version of Set Covering can be reduced in polynomial time to the considered problem. Let consider that $\Gamma_i^1 = 0$, $\forall i \in \mathcal{N}$ (green energy is not available).

Each item of the set M is represented by a SC. Each subset N_j is represented by a SC, as well. Let us call $\mathcal{N}_1$ the set of SCs representing the elements of M, and $\mathcal{N}_2$ the set of SCs representing the elements of N. Thus, the set of SCs $\mathcal{N} = \mathcal{N}_1 \cup \mathcal{N}_2$. Since, by construction, $\mathcal{N}_1 \cap \mathcal{N}_2 = \emptyset$, $|\mathcal{N}| = |M| + |N|$. We consider $|\mathcal{T}| = 1$, $|\mathcal{L}| = 1$, $|\mathcal{K}| = 1$, $m_{11} = 1$, $b_1 = 1$, $\mathcal{U} = 2|M|$. Parameters associated with SCs are defined in a different way for the two subsets of SCs and are summarized in Tables 9 and 10.

The arrival rate λ in SCs representing elements of M is equal to 1, while it is 0 for SCs representing elements in N. As the SCs belonging to $\mathcal{N}_1$ have no capacity, and VM energy consumption equal to 10κ, in any feasible solution, whose value is lesser or equal than the threshold κ, the arrival rate of SCs belonging to $\mathcal{N}_1$ must be assigned to SCs belonging to $\mathcal{N}_2$. On the other hand, the average utilization rate of SC belonging to $\mathcal{N}_2$ is high enough to deal with all the requests. If a SC belonging to $\mathcal{N}_2$ is activated, at most one VM can be activated, due to the value of C_i.

The corresponding energy consumption is equal to 1: thus the energy consumption associated to active VMs corresponds to the cardinality of activated SCs. Switching on and off cost are assumed with value zero.

Network costs are used to represents the coverage matrix. Each SC $j \in \mathcal{N}_2$ corresponds to one subset N_j. Only the requests of SCs in $\mathcal{N}_1$ associated to the elements of N_j can be served by SC j. Therefore, the energy cost on the network are set so as to guarantee that any unfeasible solution for the Set Covering has a cost higher than the threshold. We consider only costs associated to activation, while switching cost are considered with value zero. The number of routers between two SCs i and j is always 1. The energy cost is 10κ for any pair of SCs which represents an unfeasible assignment for the Set Covering Problem, while it is equal to zero if the assignment is feasible.

If a solution of the considered problem is found, whose cost is below κ, the corresponding solution of Set Covering, which has the same cost, can be derived as follows. The set of selected subsets F is the set of SCs in $\mathcal{N}_2$ in which one VM is active; each element in M is covered by at least one of the selected subsets, as each element belonging to $\mathcal{N}_1$ is redirected toward SCs belonging to $\mathcal{N}_2$ with a network cost equal to 0. On the other hand, if no solution of the considered problem is found whose cost is below the threshold, then no feasible solution for Set Covering with cardinality at most κ exists. In fact, either one SC in $\mathcal{N}_1$ is activated or a SC $i \in \mathcal{N}_1$ is assigned to $j \in \mathcal{N}_2$ such that $i \notin N_j$.

Appendix B. Model parameters details

In this section we report the details of the parameters we used for performing the paper numerical analyses.

Table 9
Reduction parameters: nodes.

Parameter	$i \in \mathcal{N}_1$	$i \in \mathcal{N}_2$
λ_i	1	0
C_i	0	1
α_i	10κ	1
D	1	1
P_i	1	1
η_i	0	0
θ_i	0	0
ρ_i	1	1
c_i	1	1

Table 10
Reduction parameters: links.

Parameter	$i \in \mathcal{N}_1$, $j \in \mathcal{N}_1$	$i \in \mathcal{N}_1$, $j \in \mathcal{N}_2$	$i \in \mathcal{N}_2$, $j \in \mathcal{N}_1$	$i \in \mathcal{N}_2$, $j \in \mathcal{N}_2$
R_{ij}	1	1	1	1
Q_{ij}	1	1	1	1
γ_{ij}	1	1	1	1
δ_{ij}	1	1	1	1
τ_{ij}	0	0	0	0
ξ_{ij}	0	0	0	0
f_{ij}^t	10κ	0 if $i \in N_j$, 10κ if $i \notin N_j$	10κ	10κ

Table 11
Number of network hops among world regions.

Region	Region	Number of hops
Europe	West USA	17
Europe	East USA	11
Europe	Asia	15
Asia	East USA	11
Asia	West USA	14
West USA	East USA	11

Table 12
Energy market managers considered in the experimental analyses.

Country	Market manager
Brazil	Electric Energy Commercialization Chamber
Italy	GME (Gestore dei Mercati Energetici)
Canada	IESO (Independent Electricity System Operator)
France	Powernext
Germany	EEX (European Energy Exchange)
Ireland	SEMO
Japan	JEPX (Japan Electric Power Exchange)
Netherlands, UK, and Belgium	APX-ENDEX
Russia	ATS (Trade System Administrator)
USA	California ISO, New England Market, and PJM

Table 13
SC locations details.

Service center	City	Country	Time zone
DC1	Mountain View	California (USA)	UTC−08
DC2	Pleasanton	California (USA)	UTC−08
DC3	San Jose	California (USA)	UTC−08
DC4	Los Angeles	California (USA)	UTC−08
DC5	Palo Alto	California (USA)	UTC−08
DC6	Seattle	Washington (USA)	UTC−08
DC7	Portland	Oregon (USA)	UTC−08
DC8	The Dalles	Oregon (USA)	UTC−08
DC9	Chicago	Illinois (USA)	UTC−05
DC10	Atlanta	Georgia (USA)	UTC−05
DC11	Reston	Virginia (USA)	UTC−05
DC12	Ashburn	Virginia (USA)	UTC−05
DC13	Virginia Beach	Virginia (USA)	UTC−05
DC14	Houston	Texas (USA)	UTC−05
DC15	Miami	Florida (USA)	UTC−05
DC16	Lenoir	North Carolina (USA)	UTC−05
DC17	Goose Creek	South Carolina (USA)	UTC−05
DC18	Pryor	Oklahoma (USA)	UTC−05
DC19	Council Bluffs	Iowa (USA)	UTC−05
DC20	Toronto	Canada	UTC−05
DC21	Berlin	Germany	UTC+01
DC22	Frankfurt	Germany	UTC+01
DC23	Munich	Germany	UTC+01
DC24	Zurich	Switzerland	UTC+01
DC25	Groningen	Netherlands	UTC+01
DC26	Mons	Belgium	UTC+01
DC27	Eemshaven	Netherlands	UTC+01
DC28	Paris	France	UTC+01
DC29	London	England	UTC+00
DC30	Dublin	Ireland	UTC+00
DC31	Milan	Italy	UTC+01
DC32	Moscow	Russia	UTC+03
DC33	San Paolo	Brazil	UTC−03
DC34	Tokyo	Japan	UTC+09
DC35	Hong Kong	China	UTC+08
DC36	Beijing	China	UTC+08

References

[1] Data Center Dynamic. Industry Census 2012: Emerging Data Center Markets, 2013. <https://www.datacenterdynamics.com/blogs/industry-census-2012-emerging-data-center-markets>.

[2] Greenpeace. How Clean is your Cloud? <http://www.greenpeace.org/international/Global/international/publications/climate/2012/iCoal/HowCleanisYourCloud.pdf>.

[3] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy Katz, Andy Konwinski, Gunho Lee, David Patterson, Ariel Rabkin, Ion Stoica, Matei Zaharia, A view of cloud computing, Commun. ACM 53 (4) (2010) 50–58.

[4] Albert Greenberg, James Hamilton, David A. Maltz, Parveen Patel, The cost of a cloud: research problems in data center networks, SIGCOMM Comput., Commun. Rev. 39 (1) (2008) 68–73.

[5] A. Beloglazov, R. Buyya, Y.C. Lee, A. Zomaya, A taxonomy and survey of energy-efficient data centers and cloud computing systems, Adv. Comput. 82 (2) (2011) 47–111.

[6] R. Bolla, R. Bruschi, F. Davoli, F. Cucchietti, Energy efficiency in the future internet: a survey of existing approaches and trends in energy-aware fixed network infrastructures, Commun. Surv. Tutorials, IEEE 13 (2) (2011) 223–244.

[7] Amazon Inc. Amazon Elastic Cloud. <http://aws.amazon.com/ec2/>.

[8] X. Zhu, D. Young, B. Watson, Z. Wang, J. Rolia, S. Singhal, B. McKee, C. Hyser, D. Gmach, R. Gardner, T. Christian, L. Cherkasova, 1000 islands: an integrated approach to resource management for virtualized data centers, J. Clust. Comput. 12 (1) (2009) 45–57.

[9] S. Casolari, M. Colajanni, Short-term prediction models for server management in internet-based contexts, Decis. Support Syst. 48 (1) (2009).

[10] D. Ardagna, S. Casolari, M. Colajanni, B. Panicucci, Dual time-scale distributed capacity allocation and load redirect algorithms for cloud systems, J. Parall. Distrib. Comput. 72 (6) (2012) 796–808.

[11] Bernardetta Addis, Danilo Ardagna, Barbara Panicucci, Mark Squillante, Li Zhang, A hierarchical approach for the resource management of very large cloud platforms, IEEE Trans. Dependable Secure Comput. 10(5) (2013) 253–272.

[12] Robert Birke, Lydia Y. Chen, Evgenia Smirni, Data centers in the cloud: a large scale performance study, in: IEEE CLOUD, 2012.

[13] Minghong Lin, A. Wierman, L.L.H. Andrew, E. Thereska, Dynamic right-sizing for power-proportional data centers, in: INFOCOM, 2011 Proceedings IEEE, 2011, pp. 1098–1106.

[14] Lei Rao, Xue Liu, Le Xie, Wenyu Liu, Minimizing electricity cost: optimization of distributed internet data centers in a multi-electricity-market environment, in: INFOCOM, 2010 Proceedings IEEE, 2010, pp. 1–9.

[15] Patrick Wendell, Joe Wenjie Jiang, Michael J. Freedman, Jennifer Rexford, Donar: decentralized server selection for cloud services, SIGCOMM Comput. Commun. Rev. 41 (4) (2010).

[16] J. Rolia, L. Cherkasova, C. McCarthy, Configuring workload manager control parameters for resource pools, in: 10th IEEE NOMS, Vancouver, Canada, April 2006.

[17] C. Tang, M. Steinder, M. Spreitzer, G. Pacifici, A scalable application placement controller for enterprise data centers, in: WWW2007, 2007.

[18] D. Carrera, M. Steinder, I. Whalley, J. Torres, E. Ayguadé, Autonomic placement of mixed batch and transactional workloads, IEEE Trans. Parall. Distrib. Syst. 23 (2) (2012) 219–231.

[19] Claudia Canali, Riccardo Lancellotti, Automated clustering of vms for scalable cloud monitoring and management, in: 2012 20th International Conference on Software, Telecommunications and Computer Networks (SoftCOM), 2012.

[20] Claudia Canali, Riccardo Lancellotti, Automated clustering of virtual machines based on correlation of resource usage, J. Commun. Softw. Syst. (JCOMSS) 8 (4) (2013).

[21] G. Scott Graham, E.D. Lazowska, J. Zahorjan, K.C. Sevcik, Quant. Syst. Perform., Comput. Syst. Anal. Queueing Netw. Models (1984).

[22] Baris Aksanli, Jagannathan Venkatesh, Liuyi Zhang, Tajana Rosing, Utilizing green energy prediction to schedule mixed batch and service jobs in data centers, in: Proceedings of the 4th Workshop on Power-Aware Computing and Systems, HotPower '11, 2011.

[23] Zhenhua Liu, Yuan Chen, Cullen Bash, Adam Wierman, Daniel Gmach, Zhikui Wang, Manish Marwah, Chris Hyser, Renewable and cooling aware workload management for sustainable data centers, in: Proceedings of the 12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS '12.

[24] Ulrich Focken, Matthias Lange, Hans-Peter Waldl, Previento – a wind power prediction system with an innovative upscaling algorithm, in: Proc. of the 2002 Global Windpower Conference, SIGMETRICS '12.

[25] A. Wolke, G. Meixner, Twospot: a cloud platform for scaling out web applications dynamically, in: ServiceWave, 2010.

[26] Amazon Inc., AWS Elastic Beanstalk. <http://aws.amazon.com/elasticbeanstalk/>.

[27] L.A. Barroso, U. Hölzle, The datacenter as a computer: an introduction to the design of warehouse-scale machines, Synthesis Lect. Comput. Archit. (2009).

[28] D. Kusic, J.O. Kephart, N. Kandasamy, G. Jiang, Power and performance management of virtualized computing environments via lookahead control, in: ICAC, 2008.

[29] D. Ardagna, B. Panicucci, M. Trubian, L. Zhang, Energy-aware autonomic resource allocation in multitier virtualized environments, IEEE Trans. Serv. Comput. 5 (1) (2012) 2–19.

[30] Miniwatts Marketing Group, Internet World Statistics, 2011. <http://www.internetworldstats.com/stats.htm>.

[31] Microsoft Corporation, Microsoft Windows Azure Virtual Machines. <http://www.windowsazure.com/en-us/home/features/virtual-machines/>.

[32] J. Almeida, V. Almeida, Da. Ardagna, I. Cunha, C. Francalanci, M. Trubian, Joint admission control and resource allocation in virtualized servers, J. Parall. Distrib. Comput. 70 (4) (2010) 344–362.

[33] SPEC, SPECWeb 2009, 2009. <http://www.spec.org/web2009/>.

[34] Andrew J. Younge, Gregor von Laszewski, Lizhe Wang, Sonia Lopez-Alarcon, Warren Carithers, Efficient resource management for cloud computing environments, in: Proceedings of the International Conference on Green Computing, GREENCOMP '10, 2010.

[35] SPEC, SPECvirt_sc2010. <http://www.spec.org/virt_sc2010/>.

[36] Juniper Networks, E120 and E320 Hardware Guide, March 2011.

[37] China Energy Intelligence and Communication Forum. <http://www.energychinaforum.com/>.

[38] Google, Google's Green PPAs: What, How, and Why, Revision 2, April 2011.

[39] Google, Google Green – The Big Picture. <http://www.google.com/green/>.

[40] National Renewable Energy Laboratory, Geothermal Resource of the United States, October 2009. <http://www.nrel.gov/gis/images/geothermal_resource2009-final.jpg>.

[41] National Renewable Energy Laboratory, Photovoltaic Solar Resource of the United States, October 2008. <http://www.nrel.gov/gis/images/map_pv_national_lo-res.jpg>.

[42] National Renewable Energy Laboratory, Wind Resource Map of the United States. <http://www.nrel.gov/gis/pdfs/windsmodel4pub1-1-9base200904enh.pdf>.

[43] IBM, IBM ILOG CPLEX Optimizer. <http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/>.

[44] S. Orlowski, M. Pióro, A. Tomaszewski, R. Wessäly, SNDlib 1.0 – survivable network design library, in: Proc. of INOC 2007, Spa, Belgium, 2007.

[45] B. Addis, A. Capone, G. Carello, L.G. Gianoli, B. Sansò, Robust Energy Management for Green and Survivable IP Networks. arXiv:1401.1977, 2014.

[46] U.S. Environmental Protection Agency, Greenhouse Gas Equivalencies Calculator, July 2011. <http://www.epa.gov/cleanenergy/energy-resources/calculator.html>.

[47] R. Miller, Google Data Center Frequently Asked Questions, March 2008. <http://www.datacenterknowledge.com/archives/2008/03/27/google-data-center-faq/>.

[48] J. Koomey, Growth in Data Center Electricity Use 2005 to 2010, Technical Report, Analytics Press, Oakland, CA, 2011.

[49] R. Miller, Google's Chiller-less DC, July 2009. <http://www.datacenterknowledge.com/archives/2009/07/15/googles-chiller-less-data-center>.

[50] X. Wang, Y. Wang, Coordinating power control and performance management for virtualized server clusters, IEEE Trans. Parall. Distrib. Syst. 22 (2) (2011) 245–259.

[51] X. Wang, M. Chen, C. Lefurgy, T.W. Keller, Ship: a scalable hierarchical power control architecture for large-scale data centers, IEEE Trans. Parall. Distrib. Syst. 23 (1) (2012) 168–176.

[52] J. Kephart, H. Chan, R. Das, D. Levine, G. Tesauro, F. Rawson, C. Lefurgy, Coordinating multiple autonomic managers to achieve specified power-performance tradeoffs, in: ICAC, 2007.

[53] H. Li, S. Venugopal, Using reinforcement learning for controlling an elastic web application hosting platform, in: ICAC, 2011.

[54] C.Z. Xu, J. Rao, X. Bu, URL: a unified reinforcement learning approach for autonomic cloud management, JPDC 72 (2) (2012) 95–105.

[55] V. Gupta, M. Harchol-Balter, Self-adaptive admission control policies for resource-sharing systems, in: SIGMETRICS, 2009.

[56] N.M. Calcavecchia, O. Biran, E. Hadad, Y. Moatti, VM placement strategies for cloud scenarios, in: IEEE CLOUD, 2012.

[57] E. Feller, C. Rohr, D. Margery, C. Morin, Energy management in IaaS clouds: a holistic approach, in: IEEE CLOUD, 2012.

[58] J. Cao, K. Hwang, K. Li, A.Y. Zomaya, Optimal multiserver configuration for profit maximization in cloud computing, IEEE Trans. Parall. Distrib. Syst. 99 (PrePrints) (2012).

[59] H. Khazaei, J. Misic, V.B. Misic, S. Rashwand, Analysis of a pool management scheme for cloud computing centers, IEEE Trans. Parall. Distrib. Syst. 99 (PrePrints) (2012).

[60] Z. Gong, X. Gu, Pac: pattern-driven application consolidation for efficient cloud computing, in: MASCOTS, 2010.

[61] A. Gandhi, Y. Chen, D. Gmach, M. Arlitt, M. Marwah, Minimizing data center sla violations and power consumption via hybrid resource provisioning, IGCC (2011).

[62] Meng Wang, V. Pappas, Li Zhang, Improving the scalability of data center networks with traffic-aware virtual machine placement, in: INFOCOM, 2010 Proceedings IEEE, 2010.

[63] Joe Wenjie Jiang, Tian Lan, Sangtae Ha, Minghua Chen, Mung Chiang, Joint VM placement and routing for data center traffic engineering, in: INFOCOM'12, 2012, pp. 2876–2880.

[64] Meng Wang, Xiaoqiao Meng, Li Zhang, Consolidating virtual machines with dynamic bandwidth demand in data centers, in: INFOCOM, 2011 Proceedings IEEE, April 2011, pp. 71 –75.

[65] D. Ardagna, C. Francalanci, M. Trubian, Joint optimization of hardware and network costs for distributed computer systems, IEEE Trans. Syst. Man Cybernet., Part A 38 (2008) 470–484.

[66] C. Guo, G. Lu, H.J. Wang, S. Yang, C. Kong, P. Sun, W. Wu, Y. Zhang, Secondnet: a data center network vitualization architecture with bandwidth guarantees, in: ACM CoNext, 2010.

[67] D. Dutta, M. Kapralov, I. Post, R. Shinde, Optimal Bandwidth-Aware VM Allocation for Infrastructure-as-a-Service, CoRR, abs/1202.3683, 2012.

[68] M. Baldi, Y. Ofek, Time for a greener internet, in: IEEE International Conference on Communications Workshops, 2009, ICC Workshops, pp. 1–6.

[69] J.C.C. Restrepo, C.G. Gruber, C.M. Machuca, Energy profile aware routing, in: IEEE International Conference on Communications Workshops, 2009, ICC Workshops, pp. 1–5, 2009.

[70] A.P. Bianzino, C. Chaudet, F. Larroca, D. Rossi, J. Rougier, Energy-aware routing: a reality check, in: IEEE GLOBECOM Workshops (GC Wkshps), 2010, pp. 1422–1427.

[71] R. Bolla, R. Bruschi, K. Christensen, F. Cucchietti, F. Davoli, S. Singh, The potential impact of green technologies in next-generation wireline networks – is there room for energy saving optimization?, IEEE Commun Mag. (COMMAG) (2011).

[72] L. Chiaraviglio, D. Ciullo, M. Mellia, M. Meo, Modeling sleep modes gains with random graphs, in: 2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), IEEE, 2011, pp. 355–360.

[73] B. Addis, A. Capone, G. Carello, L.G. Gianoli, B. Sansò, Multi-period traffic engineering of resilient networks for energy efficiency, in: IEEE Online Conference on Green Communications (GreenCom) 2012, 2012, pp. 14–19.

[74] E. Amaldi, A. Capone, L.G. Gianoli, L. Mascetti, Energy management in ip traffic engineering with shortest path routing, in: 2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011, pp. 1–6.

[75] B. Addis, A. Capone, G. Carello, L.G. Gianoli, B. Sansò, Energy management through optimized routing and device powering for greener communication networks, IEEE/ACM Trans. Netw. 22 (1) (2014) 313–325.

[76] M. Zhang, C. Yi, B. Liu, B. Zhang, GreenTE: power-aware traffic engineering, 2010.

[77] L. Chiaraviglio, M. Mellia, F. Neri, Minimizing ISP network energy cost: formulation and solutions, IEEE/ACM Trans. Netw. PP (99) (2011).

[78] N. Vasić, D. Kostić, Energy-aware traffic engineering, in: Proc. of the 1st International Conference on Energy-Efficient Computing and Networking, ACM, 2010, pp. 169–178.

[79] E. Amaldi, A. Capone, L.G. Gianoli, Energy-aware {IP} traffic engineering with shortest path routing, Comput. Netw. 57 (6)(2013) 1503–1517.

[80] Hong Xu, Chen Feng, Baochun Li, Temperature aware workload management in geo-distributed datacenters, in: Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS '13, ACM, New York, NY, USA, 2013, pp. 373–374.

[81] Zhenhua Liu, Adam Wierman, Yuan Chen, Benjamin Razon, Niangjun Chen, Data center demand response: avoiding the coincident peak via workload shifting and local generation, in: Proceedings of the ACM SIGMETRICS/International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS '13, ACM, New York, NY, USA, 2013, pp. 341–342.

[82] Hao Wang, Jianwei Huang, Xiaojun Lin, Hamed Mohsenian-Rad, Exploring smart grid and data center interactions for electric power load balancings, in: ACM Greenmetrics, Pittsburg, PA, June 213, 2013.

[83] Jinlong Tu, Lian Lu, Minghua Chen, Ramesh K. Sitaraman, Dynamic provisioning in next-generation data centers with on-site power production, in: Proceedings of the Fourth International Conference on Future Energy Systems, e-Energy '13, ACM, New York, NY, USA, 2013, pp. 137–148.

[84] Zhenhua Liu, Minghong Lin, Adam Wierman, Steven H. Low, Lachlan L.H. Andrew, Greening geographical load balancing, in: Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS '11, ACM, New York, NY, USA, 2011, pp. 233–244.

[85] Zhenhua Liu, Minghong Lin, Adam Wierman, Steven H. Low, Lachlan L.H. Andrew, Geographical load balancing with renewables, SIGMETRICS Perform. Eval. Rev. 39 (3) (2011) 62–66.

[86] K. Le, R. Bianchini, T.D. Nguyen, O. Bilgir, M. Martonosi, Capping the brown energy consumption of internet services at low cost, 2010, pp. 3–14.

[87] M. Etinski, M. Martonosi, K. Le, R. Bianchini, T.D. Nguyen, Optimizing the use of request distribution and stored energy for cost reduction in multi-site internet services, in: Sustainable Internet and ICT for Sustainability (SustainIT), 2012, pp. 1–10, 2012.