An Optimal Approach to Reduce Electricity and Maintenance Costs in Cloud Data Centers

Luca Chiaraviglio, *Senior Member, IEEE,* Mohammad Shojafar, *Member, IEEE,* Claudia Canali, Fabio D'Andreagiovanni, Riccardo Lancellotti, Nicola Blefari-Melazzi, *Senior Member, IEEE*

Abstract—We target the problem of managing the power states of the servers in a Cloud Data Center (CDC) to jointly minimize the electricity consumption and the maintenance costs derived from the variation of power (and consequently of temperature) on the servers. More in detail, we consider a set of VMs and their requirements in terms of CPU and memory across a set of Time Slot (TSs). We then model the consumed electricity by taking into account the VMs processing costs on the servers, the costs for transferring data between the VMs, and the costs for migrating the VMs across the servers. In addition, we employ a material-based fatigue model to compute the maintenance costs needed to repair the servers, as a consequence of the variation over time of their power states. We then optimally formulate the Optimal Maintenance and Electricity Costs (OMEC) problem, which performs the allocation of VMs to the servers for each TS. Our results, obtained over different scenarios, show that previous energy-aware solutions not considering the maintenance costs tend to notably increase the total costs. On the other hand, OMEC is always able to minimize the sum of maintenance plus electricity costs.

Index Terms—Cloud Computing, Cloud Data Center, Maintenance Costs, Electricity Costs, Fatigue, Energy-efficiency.

1 Introduction

Data Centers (DCs) have become a key aspect of the Information and Communication Technology (ICT) sector. Historically, the idea of exploiting DCs for computing tasks has its roots during the first half of the 19th century, where different prominent researchers define the concept of global brain [1], [2], with the goal of providing encyclopedic ways of knowledge. Since then, the incredible growth in the ICT sector, including the improvements in HardWare (HW) manufacturing, as well as the almost-infinite features provided by SoftWare (SW), have completely revolutionized the possibility of exploiting DCs for computing purposes. Nowadays, DCs are intensely widespread worldwide to sustain a variety of applications, such as web browsing, streaming, high definition videos, and cloud storage. Not surprisingly, DCs generally adopt the cloud computing paradigm [3], [4], according to which the virtualized applications (and entire operating systems) are run over a set of distributed physical servers, which may be even located in different continents. Hence, the management of a Cloud Data Center (CDC) is an aspect of fundamental importance for the DC owner (which is referred as a content provider from here on).

In an era where the amount of computing information is constantly growing [5], a primary need for a content provider is to efficiently manage CDCs. Apart from the fixed costs, that are related to the installation of CDCs equipment [6], a big worry for a content provider is how to deal with the CDCs power consumption and the related electricity costs [7]. In this context, the content provider has to face the large amount of power consumed by its own CDCs. As a result, the decrease of power consumption in CDCs has been traditionally a hot topic (see e.g. the survey [8]). In this context, different works (see e.g., [9], [10]) target the reduction of power for the servers in a CDC through the management of their power states. Among them, the application of a Sleep Mode (SM) state to a subset of servers is a very promising approach [11], [12]. More in detail, thanks to the fact that the traffic from users is not constant, and in general it varies across the different hours of the day, it is possible to put in SM different servers in a CDC, and to concentrate the users traffic on a subset of servers, which remain in an Active Mode (AM). In this way, a reduction of power, and consequently of the associated electricity costs paid by the content provider, is achieved.

Although the application of SM is able to ensure lower electricity costs compared to the case in which all the servers are always powered on, the transitions between SM and AM, especially when they are applied over periods of months and years, tend to have a negative effect on the maintenance costs paid by the content provider [13]. More in detail, when the server is put in SM, a prompt decrease in the temperature of its components (especially for CPU and memories) is observed [14]. Specifically, the temperature drops from pretty high values (typically higher than 70°-80° [Celsius]) to the room temperature, which is typically cooled and kept around 20° [Celsius]. On the other hand, the opposite effect on the temperature is observed when the server passes from SM to AM. The variation of temperature on the electronics components, especially when it is repeated over time, tends to introduce thermal fatigue effects [15], [16]. This phenomenon is similar to the mechanical fatigue experienced by an airplane fuselage, subject to cabin pressurization and depressurization over different flights, which may deteriorate it in the long term [17]. In a similar way, the HW equipment, when it is subject to large temperature transitions, tends to increase its failure rate. More in detail, fatigue (and crack) effects are experienced e.g. by the

L. Chiaraviglio and N. Blefari-Melazzi are with the Dept. of Electronic Engineering, University of Rome Tor Vergata (Italy), M. Shojafar is with CNIT (Italy), Claudia Canali and Riccardo Lancellotti are with the Department of Engineering Enzo Ferrari, University of Modena and Reggio Emilia (Italy) Fabio D'Andreagiovanni is with CNRS (France) and the Department of Computer Science, Sorbonne University - UTC (France)
E-mail: luca.chiaraviglio@uniroma2.it, mohammad.shojafar@cnit.it, claudia.canali@unimore.it, d.andreagiovanni@hds.utc.fr, riccardo.lancellotti@unimore.it, nicola.blefari-melazzi@uniroma2.it
Manuscript received August 12th, 2017.

solder joints connecting the CPU/memories to the motherboard [18]. As a consequence, a server subject to frequent AM/SM transitions will experience failure events more often, compared to the case in which it is always left in AM, thus increasing the associated maintenance costs in order to fix and/or replace the failed components. In the worst case, the maintenance costs will be even larger than the electricity saved from the application of SM, thus producing a monetary loss to the content provider [13].

In this context, different natural questions arise, such as: What is the impact of the maintenance costs on the total ones? Is it beneficial to trade between electricity consumption and maintenance costs? How to optimally formulate the problem? How to properly evaluate it? The goal of this paper is the shed light on these issues. More in detail, we first present a simple (yet effective) model to compute the maintenance costs, given the variation over time of the power states for a set of servers. In addition, we adopt a detailed model to compute the power consumed by the CDC. Specifically, our power model takes into account the CPU-related electricity costs of the servers, the costs for transferring data among the servers, and the costs for migrating the Virtual Machines (VMs) running on the servers. We then optimally formulate the problem of jointly reducing the CDC electricity consumption and the related maintenance costs. Our results, obtained by optimally solving the proposed problem on a realistic case study, clearly show that our solution is able to wisely trade between maintenance and electricity costs in order to provide monetary savings for the content provider. On the other hand, we show that energy-aware strategies, targeting solely the electricity bill reduction, tend to notably increase the maintenance costs. To the best of our knowledge, none of the previous works in the CDC research field has conducted a similar analysis.

Although the results reported in this paper are promising, we point out that other costs than the ones considered here may increase the maintenance bill. Specifically, the cost of regular updates, due to HW/SW upgrades, may have an impact on the maintenance costs paid by the content provider. In addition, the adoption of renewable energy sources may also vary the electricity bill. Both these issues, which are not considered in this work, can be potentially added in our framework.

The rest of the paper is organized as follows. Related works are reviewed in Sec. 2. The reference CDC architecture is briefly overviewed in Sec. 3. Sec. 4 presents the considered models to compute the maintenance costs and the electricity costs in a CDC. The problem of jointly managing the electricity and the maintenance costs triggered by fatigue processes is formulated in Sec. 5. The considered scenarios and the input parameters to our solution are detailed in Sec. 6. Results are reported in Sec. 7. Finally, Sec. 8 concludes our work.

2 Related Work

In the following, we will briefly discuss the main literature in CDC related to our work. We first describe solutions targeting the management of energy and/or electricity in CDCs. Then, we move our attention to researches targeting the management of CDC failures.

2.1 Energy and Electricity Management in CDCs

Features such as electricity, power, as well as computing and network management tasks are addressed in [19], [20], [21], [22], [23], [24], [25], [26], [27], [28], [29]. A detailed overview of energy consumption models in Data Center (DC) is provided by Dayarathna *et al.* in [19]. In this context, several works target the management of a CDC by: i) providing algorithms for VM live migrations [20], [21], [22], ii) considering distributed server/CDC applications [23], [24], [25], [26], iii) focusing on business process management [27], and iv) detailing memory and storage management solutions [28], [29].

Focusing on the aspect of VM live migrations, Voorsluys *et al.* [20] adopt live migration of VMs, with to goal of reducing energy in the CDC while guaranteeing the performance to applications. However, this work does not consider the server maintenance costs. Moreover, the costs of VMs migration and as well as data transferring between VMs in a CDC environment are not taken into account. Moreover, Liu *et al.* in [22] present a cost-aware learned knowledge method and an adaptive network bandwidth management, by applying VM live migration estimation to achieve power saving in the CDC. In addition, Soni *et al.* in [23] derive computing cost models for the CDC such that they try to cover the VMs' over/under loadings based on priority and states. Indeed, their proposed algorithm is able to manage load distribution among various applications running in each VM. Besides, Bi *et al.* in [25] present a queue-aware multi-tier applications model inside the CDC. In addition, they computed the number of servers that must be allotted to each tier in order to meet the response time per application per server. Moreover, they considered the CPU resources per-VM in the CDC. However, a live VMs transferring is not performed. On the other hand, Han *et al.* in [26] present an adaptive cost-aware elasticity method in order to scale up/down multi-tier cloud applications to meet run-time varying application demands. Nevertheless, the complexity of the proposed model in computational management is quadratic per-application. Focusing then on the memory and storage management, Song *et al.* in [29] employs power performance information to estimate the desired storage and memory parameters in order to preserve energy and costs in the CDC. It is important to note that their quasi-analytical performance modeling can be accurate, but it requires a deep understanding of each individual application running on the VM and the server. Therefore, several information is preliminary needed. As a result, the pre-processing time of the problem may be increased.

2.2 Failure Management in CDCs

Server failure is recognized as one of the costs for the cloud by Greenberg *et al.* [30]. In this context, different works target the reduction of the impact of the failure events by proposing efficient DC architectures. In particular, Guo *et al.* propose Dcell [31], a scalable and recursive architecture which is also fault-tolerant. Greenberg *et al.* [32] present VL2, a scalable and flexible DC network which is tolerant to failures experienced by networking equipment. Guo *et al.* [33] details BCube, an architecture for modular DCs, which is able to guarantee a graceful performance degradation as the server failure rate increases. Moreover, according to Kliazovich *et al.* [34], when the DC temperatures are not kept within their operational limits the HW reliability is decreased, thus bringing to a potential violation of Service Level Agreements (SLAs). In addition, the optimization of thermal states and cooling system operation is recognized as a challenge by Beloglazov *et al.* [10]. Moreover, a detailed failure analysis of failures in a DC is performed by Gill *et al.* [35]. However, their work is mainly focused on network devices and not on

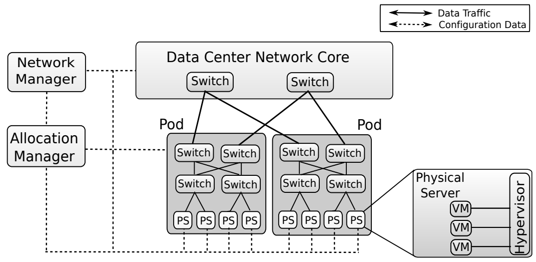


Fig. 1. Cloud Data Center Architecture.

servers like in our case. Eventually, a characterization of the HW components of the servers in terms of reliability is performed by Vishwanath *et al.* [36]. In particular, this work reports that the failure in one of the server HW components is a common event experienced in large DCs. In addition, Zhang *et al.* [37] propose Venice, an availability-aware framework able to guarantee availability requirements to the service providers. Finally, Jhawar and Piuri [38] propose an approach to measure the effectiveness of fault tolerance mechanisms in Infrastructure as a Service (Iaas) cloud, by also providing a solution to select the best mechanism satisfying the users requirements.

3 Cloud Data Center Architecture

Fig. 1 reports the main building blocks of the considered CDC architecture. More in detail, the CDC is composed of VMs, hypervisors, Physical Servers (PSs), switches and management entities. Each VM is hosted in a PS. The set of VMs in a PM is managed by an hypervisor. Moreover, the PSs are grouped in Pods. The interconnection between PSs in the same Pod is realized by means of a set of redundant switches and physical links. In addition, a DC network, again composed of switches and physical links, provides connectivity among the different Pods. Moreover, a centralized network manager (top left part of the figure) is then in charge of managing the set of networking devices, e.g. by providing software-defined functionalities. Finally, an allocation manager (mid left part of the figure) distributes the VMs over the PSs, by ensuring that each VM receive the required amount of CPU and memory from the PS hypervisor.

Focusing in more detail on the tasks performed by the allocation manager, this element is in charge of running the proposed VMs allocation algorithm, which is able to trade between electricity costs and maintenance ones by acting on the PSs power states. In our work, we assume that time is discretized in Time Slots (TSs), and that the allocation algorithm is run for every TS. Given the power state of the PSs at previous TS (AM or SM), the allocation of VMs at previous TS, and the VMs requests in terms of CPU and memory for the current TS, the allocation manager computes the allocation of VMs for the current TS. Eventually, the allocation manager notifies the servers that need to be put in AM/SM for the current TS. In case a server was in AM at previous TS and needs to be put in SM at current TS, the allocation manager interacts with the server operating system to gracefully halt the machine.

4 Costs Models

We first consider the computation of the maintenance and electricity costs for a generic TS duration, denoted with δ [h]. We initially present the model to compute the maintenance costs in a CDC subject to fatigue effects. Next, we detail the model the compute the electricity costs. Finally, we discuss the interdependence between the two models.

4.1 Maintenance Cost Model

We first introduce a failure model in order to take into account the impact of power transitions on the PS. We start from [13], in which authors present a generic model that can be applied to computing (and networking) equipment. We focus on a generic PS i in the CDC. The total Failure Rate (FR) for PS i is defined as:

$$\phi_i^{TOT} \triangleq \underbrace{\phi_i^{AM}\left(1-\frac{\tau_i^{SM}}{T}\right)}_{\text{FR due to AM state}} + \underbrace{\phi_i^{SM}\cdot\frac{\tau_i^{SM}}{T}}_{\text{FR due to SM state}} + \underbrace{\frac{\eta_i}{N_i^F}}_{\text{FR due to power transitions}} \tag{1}$$

where ϕ_i^{AM} [1/h] is the Failure Rate (FR) of the PS when it is always kept in AM (i.e., no SM is applied), τ_i^{SM} [h] is the amount of time the PS has spent in SM (from the beginning of the simulation up to current TS), T [h] is the total amount of time under consideration, ϕ_i^{SM} [1/h] is the server FR when it is always left in SM (i.e. no AM is applied), η_i [1/h] is the frequency of power state transitions between SM and AM, and N_i^F is the number of AM-SM cycles before a failure occurs. As reported in [13], the main assumptions of this model are that the failures are assumed to be statistically independent of each other and that their effect is additive. By observing in more detail Eq. (1), we can notice two different effects. Specifically, when the amount of time in SM τ_i^{SM} is increased, the resulting FR ϕ_i^{TOT} tends to the value ϕ_i^{SM}, which is, in general, lower than ϕ_i^{AM} (thanks to the fact that the temperature in SM is much lower compared to the AM case). On the other hand, the number of transitions between AM and SM tends to increase with time, thus increasing the last term of Eq. (1), and consequently the total FR ϕ_i^{TOT}. This last term tends to dominate the FR, especially when the amount of time under consideration T is in the order of months/years.

In the following, we introduce a simple metric, called Acceleration Factor (AF), to better capture the model features. More in detail, the AF, which is a metric commonly adopted in material fatigue researches [16], [39], is defined as the ratio between the observed FR ϕ_i^{TOT} and the FR by keeping the PS always in AM, i.e., ϕ_i^{AM}. More formally, we have:

$$AF_i^{TOT} \triangleq \frac{\phi_i^{TOT}}{\phi_i^{AM}} = 1 - \underbrace{\left(1-AF_i^{SM}\right)\frac{\tau_i^{SM}}{T}}_{\text{Lifetime increase (short term)}} + \underbrace{\Psi_i\cdot\rho_i}_{\text{Lifetime decrease (long term)}} \tag{2}$$

where AF_i^{SM} is defined as $\frac{\phi_i^{SM}}{\phi^{AM}}$ (which is typically lower than 1 as the FR in SM is lower than the one in AM), ρ_i is the total number of power state transitions up to TS i and Ψ_i is a weight parameter. When $AF_i^{TOT} < 1$ the PS lifetime (i.e., the time between two failure events) is higher compared to the case in which the PS is always left in AM. On the other hand, when $AF_i^{TOT} > 1$ the lifetime is lower compared to the AM case. The value of AF_i^{TOT} gives exactly the amount of lifetime reduction for the PS, e.g., if $AF_i^{TOT} = 30$, the PS will experience a lifetime reduction of 30 times compared to the case in which it is always kept in AM. Clearly, the application of different power states has an impact on the values of AF_i^{TOT}. More in detail, when the observation period (i.e. the time passed from the beginning of the experiment up to the current time slot) is in the order of months/years, the term $\Psi_i \cdot \rho_i$ becomes predominant, i.e., the application of different power states tends to increase ρ_i, and consequently the AF. Finally, we can note that the AF is influenced by parameters τ_i^{SM} and ρ_i, which depend on the specific policy used to put the PS in SM/AM, and by parameters AF_i^{SM} and Ψ_i, which instead depend on the materials used to build the PS (and their strength against fatigue effects). In principle, PSs whose components exhibit higher values of Ψ_i are more prone to fatigue effects, and consequently to lifetime degradation. The actual setting of parameters AF_i^{SM} and Ψ_i will be discussed in more detail in Sec. 6.

Finally, we compute the total maintenance costs for all the PSs in the CDC as:

$$C_M^{TOT} = K_R \cdot \delta \cdot \sum_{i \in \mathcal{M}} \left(AF_i^{TOT} \cdot \phi_i^{AM} \right) \quad [\$] \tag{3}$$

where $\mathcal{M}$ is the set of PSs in the CDC, K_R [$] is the reparation cost for one PS (i.e., the cost for fixing the PS without the need to replace it with a new one), and δ is the duration of the considered TS. In this work, we assume that the PS failures can be repaired by, e.g., the substitution of only the failed components with new ones. We believe that this assumption is more realistic compared to the case in which a PS is always replaced with a new one each time a failure is experienced. Finally, we stress the fact that the total maintenance costs C_M^{TOT} may include also the costs for HW upgrades and SW updates, as well as scheduled maintenance operations. These terms can be added as additional costs in Eq. (3), and they are left for future work.

4.2 Electricity Cost Model

We model the electricity costs as the sum of three different contributions: i) the data processing costs on the PSs, ii) the data transferring costs among the VMs located on different PSs, and iii) the costs for migrating the VMs across different PSs. The following subsections detail the different cost components.

4.2.1 Data Processing Costs

We adopt the assumption of [10], according to which the power consumption of each PS in AM is proportional to the CPU utilization due to data processing tasks running on the hosted VMs. On the other hand, when the PS is in SM, we assume that its power consumption is negligible. We denote the total electricity costs due to processing tasks with C_E^{PROC}. More formally, we have:

$$C_E^{PROC} = K_E \cdot \delta \sum_{i \in \mathcal{M}} \left[u_i \left(P_i^{MAX} - P_i^{IDLE} \right) + O_i \cdot P_i^{IDLE} \right] \tag{4}$$

where K_E [$/Wh] is the hourly electricity cost, δ [h] is the time slot duration, u_i is the CPU utilization of the server at current TS (ranging between 0 and 1), P_i^{MAX} [W] is the power consumption of PS i when its CPU is fully utilized, P_i^{IDLE} [W] is the power consumption of the PS when its CPU is idle, and O_i is the power state of PS i (0 if it is in SM, 1 otherwise). Note that, when the PS is in SM (i.e. O_i=0), it holds that $u_i = 0$.

4.2.2 Data Transferring Costs

We then consider the electricity costs derived from the exchange of data between VMs running on different PSs. By adopting a common assumption in the literature [40], [41], we assume that the total costs due to data transferring are the sum of a static term, which considers the power consumed by the network interfaces of the PS, and a linear one, which instead takes into account the amount of data transferred between VMs. The total costs due to migrations, which are denoted with C_E^{TR}, are then expressed as:

$$C_E^{TR} = K_E \cdot \delta \sum_{i \in \mathcal{M}} \left(O_i \cdot P_i^{TR-IF} + \sum_{j \in \mathcal{M}} \sum_{k,w \in \mathcal{N}} d_{kw}^{ij} \cdot P_{ij}^{TR-NET} \right) \tag{5}$$

where $\mathcal{N}$ is the set of VMs in the CDC, P_i^{TR-IF} [W] is the power of the network interfaces of PS i, d_{kw}^{ij} [Mb/h] is the hourly amount of data traffic exchanged between VM k on PS i and VM w on PS j (which is equal to 0 if either PS i or PS j is in SM), and P_{ij}^{TR-NET} [W/Mb] is the power consumption consumed for transferring one [Mb] of information between PS i and PS j (by assuming that VM k is hosted in PS i, and that VM w is located in PS j).

4.2.3 Migration Costs

Finally, we consider the costs that are paid when the VMs are moved across the PSs. For example, a typical event requiring VM migration is the activation of SM on a PS. Before the PS applies SM, all the VMs running on it have to be moved to other PS(s). We assume that the VM migration involves the whole copy of the VM memory from the old PS to the new one.[1] Eventually, the process of copying the memory requires an additional amount of overhead power, which needs to be properly taken into account. This amount of power is driven by the fact that VM migration introduces a performance degradation, which may be even in the order of 10% according to [42]. The total migration costs, which are denoted with C_E^{MIG}, are then defined as:

$$C_E^{MIG} = K_E \cdot \sum_{i,j \in \mathcal{M}} \sum_{k \in \mathcal{N}} m_{ijk} \left(\mu_k \cdot P_{ij}^{TR-NET} + P_i^{OH} + P_j^{OH} \right) \tag{6}$$

where m_{ijk} is a binary variable taking value 1 if the k-th VM on PS i is migrated on PS j (0 otherwise), μ_k [Mb] is the amount of memory of the VM k, P_{ij}^{TR-NET} [W/Mb] is again the power consumption consumed for transferring one [Mb] of information between PS i and PS j, P_i^{OH} [W] and P_j^{OH} [W] are the amount of overhead power consumed during the migration process by PS i and j, respectively.

4.2.4 Total Electricity Costs

The total electricity costs consumed by the CDC are then computed as the sum of the considered costs:

$$C_E^{TOT} = C_E^{PROC} + C_E^{TR} + C_E^{MIG} \quad [\$] \tag{7}$$

4.3 Interdependence Between The Costs Models

The presented electricity and maintenance costs models are strictly independent of each other. Let us consider for simplicity the case in which a generic PS i was in AM at previous TS and is put in SM at current TS. In this case, the number of power state transitions ρ_i is increased. This inevitably increases the PS AF reported in Eq. (2), and consequently the reparation costs in Eq. (3). On the other hand, by imposing the SM state, O_i is set to 0. Therefore, the data processing costs in Eq. (4) and the data transferring costs in Eq. (5) are equal to 0 for PS i. On the other hand, the VMs running on the PS will be moved to other PSs, thus increasing

1. The actual amount of exchanged data may be slightly higher than the size of memory, due to the retransmission of dirty memory pages. However, the typically small size of the active page set w.r.t. the global memory space of the VM allows us to neglect this effect.

the migration costs in Eq. (6). In a similar way, the power state change from SM in the previous TS to AM in the current time slot also tends to increase the reparation costs, while also increasing the electricity costs.

In this context, a natural question is: How to set the power states for the whole set of PSs in the CDC in order to trade between the costs? To answer this question, we optimally formulate in the next section the problem of minimizing the electricity and maintenance costs in a CDC.

5 Problem Formulation

We first consider the extension of our cost model by introducing the set of TSs, which is denoted by $\mathcal{T}$. Then, for each TS $t \in \mathcal{T}$, we target the problem of jointly managing maintenance and electricity costs in the CDC. We initially detail each set of constraints, and then we provide the entire formulation.

5.1 Maintenance Costs Constraints

We first consider the constraints related to the computation of the maintenance costs. We initially introduce the variable $\tau^{ALL}(t)$ [h] to compute the total amount of time elapsed from the initial TS up to TS t. $\tau^{ALL}(t)$ is computed as:

$$\tau^{ALL}(t) = \tau^{ALL}(t-1) + \delta(t) \tag{8}$$

where $\tau^{ALL}(t-1)$ [h] is the total elapsed time up to TS $(t-1)$ and $\delta(t)$ [h] is the duration of current TS t.

We then denote with $\tau_i^{SM}(t)$ [h] the total time in SM for PS i up to TS t. $\tau_i^{SM}(t)$ is then computed as:

$$\tau_i^{SM}(t) = \tau_i^{SM}(t-1) + \delta(t)\left[1 - O_i(t)\right], \quad \forall i \in \mathcal{M} \tag{9}$$

where $\tau_i^{SM}(t-1)$ [h] is the total time in SM for PS i up to TS $(t-1)$, and $O_i(t)$ [units] is a binary variable for the power state of PS i, taking value 1 if PS i is in AM at TS t, 0 otherwise.

We then introduce the binary variable $z_i(t)$ [units], which takes value 1 if PS i has experienced a power state transition (from SM to AM, or the opposite) between TS t and TS $(t-1)$, 0 otherwise. $z_i(t)$ is formally defined as:

$$z_i(t) = \begin{cases} O_i(t) - O_i(t-1) & \text{if} \quad O_i(t-1) == 0 \\ O_i(t-1) - O_i(t) & \text{if} \quad O_i(t-1) == 1 \end{cases}, \forall i \in \mathcal{M} \tag{10}$$

We then introduce the integer variable $\rho_i(t)$ [units], which computes the total number of transitions for PS i up to TS t:

$$\rho_i(t) = \rho_i(t-1) + z_i(t), \quad \forall i \in \mathcal{M} \tag{11}$$

where $\rho_i(t-1)$ [units] is the total number of transitions for PS i up to TS $(t-1)$.

In the following, we denote with $AF_i^{TOT}(t)$ [units] a continuous variable storing the value of AF for PS i up to TS t. The total AF is computed as:

$$AF_i^{TOT}(t) = 1 - \left(1 - AF_i^{SM}\right)\frac{\tau_i^{SM}(t)}{\tau^{ALL}(t)} + \Psi_i \cdot \rho_i(t), \quad \forall i \in \mathcal{M} \tag{12}$$

where AF_i^{SM} [units] is the AF of the PS when it is always kept in SM, and Ψ_i [units] is the weight parameter for the number of power state transitions $\rho_i(t)$ [units].

Finally, we introduce the variable $C_M^{TOT}(t)$ [\$] to store the maintenance costs of the CDC up the TS t. The total maintenance costs are computed as:

$$C_M^{TOT}(t) = C_M^{TOT}(t-1) + K_R \cdot \delta(t) \sum_{i \in \mathcal{M}} AF_i^{TOT}(t) \cdot \phi_i^{AM} \tag{13}$$

where $C_M^{TOT}(t-1)$ [\$] are the maintenance costs up to TS $(t-1)$, K_R [\$] is the reparation cost for one PS, and ϕ_i^{AM} [1/h] is the FR of PS i when it is always kept in AM.

5.2 Electricity Costs Constraints

In the following, we consider the computation of the different terms of the electricity costs. More in detail, we start by computing the CPU utilization of each PS. We denote with $u_i(t)$ [units] a continuous variable storing the CPU utilization of PS i at TS t. $u_i(t)$ [units] is expressed as the summation of the CPU consumed by the VMs running on PS i, normalized by the total CPU available on the PS. More formally, we have:

$$u_i(t) = \sum_{k \in \mathcal{N}} x_{ik}(t) \cdot \frac{\gamma_k(t)}{\gamma_i^{MAX}}, \quad \forall i \in \mathcal{M} \tag{14}$$

where $x_{ik}(t)$ [units] is a binary variable taking the value 1 if VM k is assigned to PS i (0 otherwise), $\gamma_k(t)$ [units] is the CPU request of VM k at TS t, and γ_i^{MAX} [units] is the maximum CPU utilization of PS i.

Given the CPU utilization $u_i(t)$ [units], we then compute the total electricity costs due to CPU processing at TS t, which are denoted as $C_E^{PROC}(t)$ [\$]. The total processing costs are defined as:

$$C_E^{PROC}(t) = K_E \cdot \delta(t) \sum_{i \in \mathcal{M}} \left[u_i(t)\left(P_i^{MAX} - P_i^{IDLE}\right) + O_i(t) \cdot P_i^{IDLE}\right] \tag{15}$$

where K_E [\$/Wh] is the electricity cost per Watt-hour, P_i^{MAX} [W] is the maximum power consumption of the PS and P_i^{IDLE} [W] is the idle power consumption of the PS.

In the following step, we compute the amount of data exchanged between VM k located on PS i and VM k located on PS w at TS t, which we denote with $d_{kw}^{ij}(t)$ [Mb/h]. This variable is equal to the amount of data traffic $D_{kw}(t)$ [Mb/h] exchanged by the VMs k and w at TS t, if k and w are located on different PSs. On the other hand, if k and w are located on the same PS, $d_{kw}^{ij}(t)$ is set to 0. More formally, we have:

$$d_{kw}^{ij}(t) = \begin{cases} p_{kw}^{ij}(t) \cdot D_{kw}(t) & \text{if} \quad i \neq j \\ 0 & \text{if} \quad i = j \end{cases}, \forall i,j \in \mathcal{M}, \forall k,w \in \mathcal{N} \tag{16}$$

where $p_{kw}^{ij}(t) = x_{ik}(t) \cdot x_{jw}(t)$ is a non linear product. In order to linearize this term, we introduce the following linear constraints:

$$p_{kw}^{ij}(t) \leq x_{ik}(t), \quad \forall i,j \in \mathcal{M}, \forall k,w \in \mathcal{N} \tag{17}$$

$$p_{kw}^{ij}(t) \leq x_{jw}(t), \quad \forall i,j \in \mathcal{M}, \forall k,w \in \mathcal{N} \tag{18}$$

$$p_{kw}^{ij}(t) \geq x_{ik}(t) + x_{jw}(t) - 1, \quad \forall i,j \in \mathcal{M}, \forall k,w \in \mathcal{N} \tag{19}$$

$$p_{kw}^{ij}(t) \geq 0, \quad \forall i,j \in \mathcal{M}, \forall k,w \in \mathcal{N} \tag{20}$$

The total data transferring costs at TS t, denoted as $C_E^{TR}(t)$, are then defined as:

$$C_E^{TR}(t) = K_E \cdot \delta(t) \sum_{i \in \mathcal{M}} \left[O_i(t) \cdot P_i^{TR-IF} + \sum_{j \in \mathcal{M}} \sum_{k,w \in \mathcal{N}} d_{kw}^{ij}(t) \cdot P_{ij}^{TR-NET}\right] \tag{21}$$

where P_i^{TR-IF} [W] is the power consumption of network interfaces of PS i, and P_{ij}^{TR-NET} [W/Mb] is the power cost to transfer one [Mb] of data between PS i and PS j.

In the next part, we compute the costs due to VM migrations across the PSs. Specifically, we first introduce the binary variable

TABLE 1
Main Notation

	Symbol	Definition	Type - Unit	Appears in Eq.
Input Parameters	$\mathcal{M}$ ($\lvert\mathcal{M}\rvert$)	Set (Number) of PSs	-	-
	$\mathcal{N}$ ($\lvert\mathcal{N}\rvert$)	Set (Number) of VMs	-	-
	$\mathcal{T}$ ($\lvert\mathcal{T}\rvert$)	Set of TSs	-	-
	$\tau^{ALL}(t-1)$	Total time up to TS $(t-1)$	[h]	(8),(12)
	$\delta(t)$	Current TS duration	[h]	(8),(9),(13),(15),(21)
	$\tau_i^{SM}(t-1)$	Total amount of time in SM for PS i up to TS $(t-1)$	[h]	(9),(12)
	$O_i(t-1)$	Power state of the PS at TS $(t-1)$: 1 AM, 0 SM	[units]	(10)
	$\rho_i(t-1)$	Total number of transitions for PS i up to TS $(t-1)$	Integer - [units]	(11)
	AF_i^{SM}	AF always in SM for PS i	[units]	(12)
	Ψ_i	Weight for power state transitions for PS i	[units]	(12)
	$C_M^{TOT}(t-1)$	Total maintenance costs for the CDC up to TS $(t-1)$	[$]	(13)
	K_R	PS maintenance costs	[$]	(13)
	ϕ_i^{AM}	FR for PS i always in AM	[1/h]	(13)
	$\gamma_k(t)$	CPU utilization of VM k at TS t	[units]	(14),(27)
	γ_i^{MAX}	Maximum CPU utilization for PS i	[units]	(14),(27)
	K_E	Electricity cost for one [Wh]	[$/Wh]	(15),(21),(23)
	P_i^{MAX}	Maximum power consumption for PS i	[W]	(15)
	P_i^{IDLE}	Idle power consumption for PS i	[W]	(15)
	$D_{kw}(t)$	Amount of data exchanged between VM k and VM w	[Mb/h]	(16)
	P_i^{TR-IF}	Power consumption of the network interfaces for PS i	[W]	(21)
	P_{ij}^{TR-NET}	Power consumption per [Mb] of information exchanged between PS i and PS j	[W/Mb]	(21),(23)
	$x_{ik}(t-1)$	VM k to PS i assignment at TS $(t-1)$	[units]	(22)
	P_i^{OH}	Overhead power for PS i	[W]	(23)
	$\mu_k(t)$	Memory consumption of VM k at TS t	[Mb]	(23),(28)
	μ_i^{MAX}	Maximum memory consumption on PS i	[Mb]	(28)
Variables	$\tau^{ALL}(t)$	Total time up to TS (t)	[h]	(8),(12)
	$\tau_i^{SM}(t)$	Total amount of time in SM for PS i up to TS t	Continuous - [h]	(9),(12)
	$O_i(t)$	Power state for PS i: 1 AM, 0 SM	Binary - [units]	(9),(10),(15),(21), (26)
	$z_i(t)$	Power transition for PS i: 1 if the PS has experienced a transition between t and $(t-1)$, 0 otherwise	Binary - [units]	(10),(11)
	$\rho_i(t)$	Total number of transitions for PS i up to TS t	Integer - [units]	(11),(12)
	$AF_i^{TOT}(t)$	Total AF for PS i at TS t	Continuous - [units]	(12),(13)
	$C_M^{TOT}(t)$	Total maintenance costs for the CDC up to TS t	Continuous - [$]	(13)
	$u_i(t)$	CPU utilization for PS i at TS t	Continuous - [units]	(14),(15)
	$x_{ik}(t)$	VM k to PS i assignment at TS t: 1 if VM k is assigned to PS k at TS t, 0 otherwise	Binary - [units]	(14),(17),(18),(19),(22),(25), (26),(27),(28)
	$p_{kw}^{ij}(t)$	1 if VM k is assigned to PS i and VM w is assigned to PS i (0 otherwise)	Binary - [$]	(16),(17),(18),(19),(20)
	$C_E^{PROC}(t)$	Data processing costs at TS t for the CDC	Continuous - [$]	(15),(24)
	$d_{kw}^{ij}(t)$	Amount of data exchanged between VM k on PS i and VM w on PS j at TS t	Continuous - [Mb/h]	(16),(21)
	$C_E^{TR}(t)$	Data transferring costs at TS t for the CDC	Continuous - [$]	(21),(24)
	$m_{ijk}(t)$	Migration of VM k from PS i to PS j at TS t: 1 if VM k is migrated from PS i to PS j at TS t, 0 otherwise	Binary - [units]	(22),(23)
	$C_E^{MIG}(t)$	Migration costs at TS t for the CDC	Continuous - [$]	(23),(24)
	$C_E^{TOT}(t)$	Total electricity costs for the CDC at TS t	Continuous - [$]	(24)

$m_{ijk}(t)$, which takes value 1 if VM k is moved from PS i to PS j at TS t, 0 otherwise. We set $m_{ijk}(t)$ with the following constraint:

$$m_{ijk}(t) = \begin{cases} x_{ik}(t-1) \cdot x_{jk}(t) & \text{if} \quad i \neq j \\ 0 & \text{if} \quad i = j \end{cases} \qquad \forall i, j \in \mathcal{M}, \forall k \in \mathcal{N} \quad (22)$$

We then store the total VM migration costs at TS t in the variable $C_E^{MIG}(t)$, which is defined as:

$$C_E^{MIG}(t) = K_E \cdot \sum_{i,j \in \mathcal{M}} \sum_{k \in \mathcal{N}} m_{ijk}(t) \left[\mu_k(t) \cdot P_{ij}^{TR-NET} + P_i^{OH} + P_j^{OH} \right] \quad (23)$$

where $\mu_k(t)$ [Mb] is the amount of allocated memory to VM k at TS t, while P_i^{OH} [W] and P_j^{OH} [W] are the overhead power consumption of PS i and PS j due to VM migrations, respectively.

The total electricity costs are then computed as the sum of the current costs, plus the costs at previous TS $C_E^{TOT}(t-1)$ [$]:

$$C_E^{TOT}(t) = C_E^{TOT}(t-1) + C_E^{PROC}(t) + C_E^{TR}(t) + C_E^{MIG}(t) \quad (24)$$

5.3 Additional Constraints

We then introduce a set of additional constraints in our problem. Specifically, we first impose than each VM has to be allocated to only one PS:

$$\sum_{i \in \mathcal{M}} x_{ik}(t) = 1, \quad \forall k \in \mathcal{N} \quad (25)$$

In addition, when a VM is assigned to a PS, the PS has to be powered on. We ensure this condition with the following constraint:

$$\sum_{k \in \mathcal{N}} x_{ik}(t) \leq |\mathcal{N}| \cdot O_i(t), \quad \forall i \in \mathcal{M} \quad (26)$$

where on the right side we have exploited the Big-M method by multiplying $O_i(t)$ for $|\mathcal{N}|$ (see [48] for the explanation of the general Big-M methodology).

In the following, we consider the fact that the CPU consumed by the VMs running on each PS i has to be lower than the CPU available on the PS. More formally, we have:

$$\sum_{k \in \mathcal{N}} \gamma_k(t) \cdot x_{ik}(t) \leq \gamma_i^{MAX}, \quad \forall i \in \mathcal{M} \quad (27)$$

TABLE 2
Settings for the Input Parameters

Parameter	Value	Reference	Appears in Eq.
$\lvert\mathcal{M}\rvert$	{3,5}	-	-
$\lvert\mathcal{N}\rvert$	{9,11,13,15}	-	-
$\mathcal{T}$	1 TS every 15 [min] for $\lvert\mathcal{T}\rvert = 1$ [year]	-	-
$\tau^{ALL}(t-1)$	Given by previous TS (set to 0 [min] for the initial TS)	-	(8),(12)
$\delta(t)$	15 [min]	[43]	(8),(9),(13),(15),(21)
$\tau_i^{SM}(t-1)$	Given by previous TS (set to 0 [min] for the initial TS)	-	(9),(12)
AF_i^{SM}	0.5 $\forall i \in \mathcal{M}$	Arrhenius Law with parameters from [14], [44]	(12)
Ψ_i	[0.01-0.1] $\forall i \in \mathcal{M}$	Computed from ϕ_i^{AM} in [13]	(12)
$C_M^{TOT}(t-1)$	Given by previous TS (set to 0 [$] for the initial TS)	-	(13)
K_R	380 [$] (reparation costs requiring two crew members, each of them costing 190 [$])	[13]	(13)
ϕ_i^{AM}	1.14×10^{-5} [1/h] $\forall i \in \mathcal{M}$	[13]	(13)
$\gamma_k(t)$	Traces from e-Health CDC	[43]	(14),(27)
γ_i^{MAX}	Maximum utilization, corresponding to 2x6 cores Xeon 5760 @ 2.93 MHz $\forall i \in \mathcal{M}$	[43]	(14),(27)
K_E	0.00016 [$/Wh]	[13]	(15),(21),(23)
P_i^{MAX}	328.2 [W] $\forall i \in \mathcal{M}$	[45]	(15)
P_i^{IDLE}	197.6 [W] $\forall i \in \mathcal{M}$	[45]	(15)
$D_{kw}(t)$	Traces from e-Health CDC	[43]	(16)
P_i^{TR-IF}	42.7 [W]	[46], [47] $\forall i \in \mathcal{M}$	(21)
P_{ij}^{TR-NET}	Dataset from [43]	[43]	(21),(23)
$x_{ik}(t-1)$	Given by previous TS (set to 0 $\forall i \in \mathcal{M}, k \in \mathcal{N}$ for the initial TS)	-	(22)
P_i^{OH}	3.28 [W] (1% of P_i^{MAX}) $\forall i \in \mathcal{M}$	[43]	(23)
$\mu_k(t)$	Traces from e-Health CDC	[43]	(23),(28)
μ_i^{MAX}	128 [GB] $\forall i \in \mathcal{M}$	[45]	(28)
C_E^{TOT}	Given by previous TS (set to 0 [$] for the initial TS) -	(24)	

Similarly, we impose a limit also for the amount of memory consumed by the VMs on each PS:

$$\sum_{k \in \mathcal{N}} \mu_k(t) \cdot x_{ik}(t) \leq \mu_i^{MAX} \quad \forall i \in \mathcal{M} \tag{28}$$

where μ_i^{MAX} [Mb] is the maximum memory consumption allowed on PS i.

5.4 Overall Formulation

The Optimal Maintenance and Electricity Costs (OMEC) problem, which aims at minimizing the costs for each TS t, is formulated as follows:

$$\min C^{TOT}(t) = \left[C_M^{TOT}(t) + C_E^{TOT}(t)\right] \tag{29}$$

subject to:

$$\begin{array}{ll} \text{Maintenance Costs Computation} & (8)-(13), \\ \text{Electricity Costs Computation} & (14)-(24), \\ \text{VM Allocation Constraint} & (25), \\ \text{PS Activation Constraint} & (26), \\ \text{Maximum CPU Capacity} & (27), \\ \text{Maximum Memory Capacity} & (28). \end{array} \tag{30}$$

under control variables: $x_{ij}(t) \in \{0,1\}$, $O_i(t) \in \{0,1\}$.

Finally, Tab. 1 reports the main notation introduced so far.

6 Scenarios and Input Parameters

Tab. 2 reports the values of the input parameters, their references, and the equations of the OMEC model where they appear. More in detail, we consider different settings for the number of PSs $\lvert\mathcal{M}\rvert$ and VMs $\lvert\mathcal{N}\rvert$. In addition, a total period of time $\lvert\mathcal{T}\rvert$ equal to 1 [year] is considered. We assume that the TS duration is equal to $\delta(t) = 15$ [min]. Moreover, the failure rate of a PS in AM ϕ_i^{AM} is set equal to 1.14×10^-5 [1/h], in accordance to the FR values expressed in Failure in Time (FIT) in [13].

In order to set the AF in SM AF_i^{SM}, we recall that this term is equal to ϕ_i^{SM}/ϕ_i^{AM}, where ϕ_i^{SM} [1/h] is the FR in SM, which is expressed by the Arrhenius law [49]:

$$\phi_i^{SM} = e^{\frac{-E_a}{(\mathcal{K} \cdot T_{SM})}} \tag{31}$$

where E_a [joule/mol] is the activation energy, $\mathcal{K} = 8.314472$ [joule / (mol kelvin)] is the Boltzmann constant, and T_{SM} [kelvin] is the temperature in SM. In our case, we have set $E_a = 30500$ [joule/mol] in accordance to the values measured for chip components in [44], $T_{SM} = 303.15$ [kelvin], corresponding to 30 [Celsius], in accordance to the real measurement performed on a PS in [14]. As a result, we get $AF_i^{SM} \approx 0.5$.

In the following, we focus on the Ψ_i parameter, which instead is the weight for the power state transitions $\rho_i(t)$. More in depth, Ψ_i is defined as $\Psi_i = 1/(\phi_i^{AM} N_i^F)$ where N_i^F is the number of cycles to failures. In our case, we consider the interval $N_i^F = [8.77 \cdot 10^5 - 8.77 \cdot 10^6]$. In particular, we set N_i^F to values higher than the ones measured under stressful conditions, i.e., between a maximum and a minimum temperature (such as the testing methodology of [50]), due to the fact that we are only applying a SM procedure, which is supposed to be less aggressive for the lifetime of the components than the test in [50]. As a result, we consider a range of Ψ_i values in the interval $[0.01 - 0.1]$.

Given these parameters, we then focus on a scenario in which the VMs requests in terms of CPU $\gamma_k(t)$, memory $\mu_k(t)$ and exchanged data $D_{kw}(t)$ vary across the set of TSs. In particular, these input parameters are derived from real measurements from an e-Health CDC [43]. Moreover, the dimensioning of the PSs in terms of power, CPU, and memory is performed considering a realistic server machine [43], [45]. Finally, the remaining parameters are set in accordance to Tab. 2.

7 Performance Evaluation

We compare the OMEC model against a classical Optimal Consolidation (OC) strategy, which is described in [9] and

TABLE 3
Costs breakdown after 1, 6 and 12 months for OMEC, OC, and OEA with $|\mathcal{M}| = 5$, and $|\mathcal{N}| = 15$.

Algorithms			$C_E^{PROC}(T')$ [$]	$C_E^{TR}(T')$ [$]	$C_E^{MIG}(T')$ [$]	$C_E^{TOT}(T')$ [$]	$C_M^{TOT}(T')$ [$]	$C^{TOT}(T')$ [$]	Ranking
15 VMs	$T' =$ 1 [month]	OC	**140.09**	90.86	26.04	256.99	25.74	282.73	#3
		OEA	140.83	**80.72**	**9.19**	**230.73**	**20.48**	**251.18**	#1
		OMEC	141.32	82.16	9.74	233.20	24.39	257.60	#2
	$T' =$ 6 [months]	OC	**851.67**	**552.23**	157.64	1561.82	**509.51**	**2071.33**	#1
		OEA	907.71	553.27	78.63	**1539.74**	694.62	2234.36	#3
		OMEC	926.29	559.44	**77.90**	1563.81	642.27	2206.08	#2
	$T' =$ 12 [months]	OC	**1703.24**	**1102.25**	315.25	**3121.48**	1859.57	4981.05	#3
		OEA	1897.79	1134.01	158.09	3190.49	1740.59	4931.08	#2
		OMEC	1916.37	1140.18	**157.36**	3214.56	**1510.92**	**4725.48**	#1

optimally formulated in [51]. The goal of OC is to minimize the energy consumed by PSs in each TS. More formally, the OC problem can be sketched as follows:

$$\min C_E^{PROC}(t) \tag{32}$$

subject to: (14), (15), (25), (26), (27), (28) and the control variables $x_{ij}(t) \in \{0,1\}$, $O_i(t) \in \{0,1\}$. In addition, we consider as an additional term of comparison the energy-aware approach of [43], which we denote as OPTIMAL ELECTRICITY COSTS ALGORITHM (OEA). The goal of OEA is to minimize the total electricity costs consumed by the PSs, including the terms due to data transferring and migrations. More formally, OEA can be sketched with the following optimization problem:

$$\min C_E^{TOT}(t) \tag{33}$$

subject to: (14)-(24), (25), (26), (27), (28) and the control variables $x_{ij}(t) \in \{0,1\}$, $O_i(t) \in \{0,1\}$.

The OMEC, OEA and OC formulations have been coded with the AMPL software, and solved over the considered scenarios with the CPLEX optimization solver on a powerful server with 16 cores and 24 [GB] of RAM. Unless otherwise specified, we have initially considered a set of $|\mathcal{M}| = 5$ PSs and $|\mathcal{N}| = 15$ VMs. In this way, we have focused on a scenario in which the PSs are always pretty loaded. We have then considered the impact on the costs by running OMEC, OEA and OC for $|\mathcal{T}| = 1$ [year], and we have selected the obtained solutions after $T' = 1, 6, 12$ [months], respectively. Tab. 3 reports the obtained results, by detailing the different costs and the final ranking in terms of total costs $C_E^{TOT}(T')$. Focusing first on the 1-month period, OEA is able to minimize the total costs. Not surprisingly, OC is able to minimize the processing costs $C_E^{PROC}(T')$, but this comes at the price of generally increasing the migrations costs $C_E^{MIG}(T')$ and the data transferring costs $C_E^{TR}(T')$ w.r.t. OEA. Moreover, we can see that OMEC requires only an additional cost of less than 6 [$] compared to OEA. Moreover, we can note that the maintenance costs $C_M^{TOT}(T')$ are much lower than the electricity costs $C_E^{TOT}(T')$. This is an expected result, since the AF, which governs $C_M^{TOT}(T')$, is mainly impacted by the number of power state transitions $\rho_i(t)$, which is still low for all the strategies at the end of the 1-month period. Eventually, when the 6-months period is taken into account, the costs are minimized by OC, which is able to: i) reduce the number of PSs powered on, and consequently decrease $C_E^{PROC}(T')$, and ii) increase the total time in SM, and consequently limit the increase in the maintenance costs $C_M^{TOT}(T')$. However, we can notice that OMEC achieves the second position in the ranking. Interestingly, when the 12-months period is considered, the best solution turns to be OMEC. Specifically, both OEA and OC have notably increased the maintenance costs, which have been instead wisely taken into account by OMEC. Overall, OMEC is able to save more than 250 [$] and more than 200 [$] w.r.t. OC and OC, respectively. Therefore, these results confirm our intuition that, when the period of time under consideration is sufficiently long (i.e., one year), a solution jointly targeting the maintenance costs and the energy ones, like done by OMEC, is the best one.

TABLE 4
Breakdown of different variables after 1, 6 and 12 months for OMEC, OC, and OEA with $|\mathcal{M}| = 5$, $|\mathcal{N}| = 15$.

Algorithms			$\sum_i \tau_i^{SM}(T')$ [h]	$\sum_i \rho_i(T')$	$\sum_i AF_i^{TOT}(T')$	$\sum_{t\in\mathcal{T}'} \sum_{i,j,k} m_{ijk}(t)$
15 VMs	$T' =$ 1 [Month]	OC	**720**	**402**	**12.54**	5193
		OEA	697	488	14.28	**3792**
		OMEC	681.50	632	17.12	3967
	$T' =$ 6 [Months]	OC	**4382**	2208	48.66	32462
		OEA	2610	2477	54.24	19858
		OMEC	2075	**2021**	**45.19**	**18778**
	$T' =$ 12 [Months]	OC	**8764**	4412	92.74	64915
		OEA	2610	2477	54.39	35082
		OMEC	2075	**2021**	**45.30**	**34002**

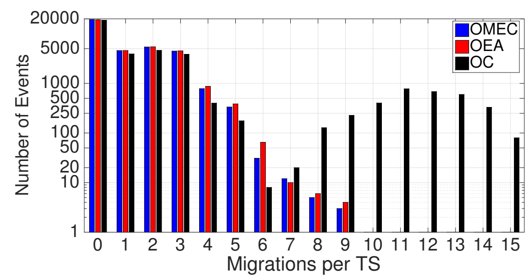
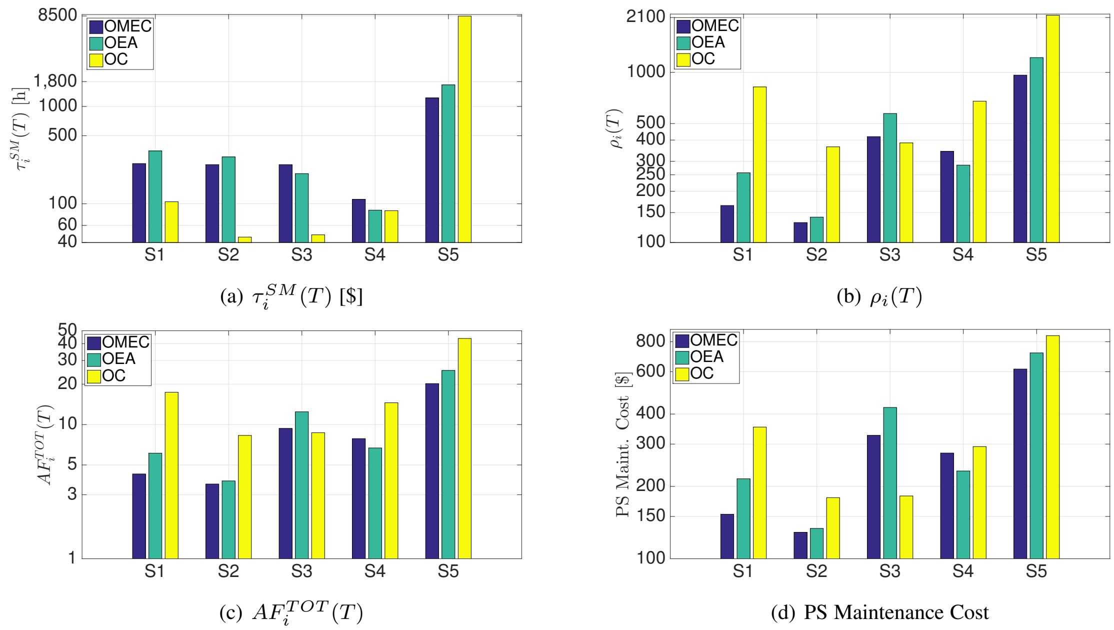


Fig. 2. Histogram of the occurrence of migrations events for OMEC, OC and OEA with $|\mathcal{M}| = 5$, $|\mathcal{N}| = 15$ and $|\mathcal{T}| = 1$ [year].

To give more insight, we report in Tab. 4 the following variables: total time in SM $\sum_i \tau^{SM}(T')$, total number of transitions $\sum_i \rho_i(T')$, total AF $\sum_i AF_i^{TOT}(T')$, total number of migrations $\sum_{t\in T'} \sum_{i,j,k} m_{ijk}(t)$. The analysis is again repeated with the three strategies and by considering $T' = 1, 6, 12$ [months], respectively. As expected, OC always maximizes $\sum_i \tau^{SM}(T')$. However, while this choice is wise for 1-month and 6-months time periods, the total number of transitions $\sum_i \rho_i(T')$ is consistently increased at the end of the 12-months period. As a result, the AF of OC is clearly higher than OMEC when $T' = 12$ [months]. Interestingly, the solution achieving the lowest AF values after the 12-months period is OMEC. Moreover, we can observe that this strategy is able also to reduce $\sum_{t\in\mathcal{T}'} \sum_{i,j,k} m_{ijk}(t)$ compared to OMEC and OEA. Therefore, OMEC is able to: i) limit the increase in the maintenance costs $C_M^{TOT}(t)$, by properly governing the AF, and ii) limiting also the number the migrations, and the associated costs $C_E^{MIG}(t)$.

In order to better investigate the impact of the different



(a) $\tau_i^{SM}(T)$ [$]

(b) $\rho_i(T)$

(c) $AF_i^{TOT}(T)$

(d) PS Maintenance Cost

Fig. 3. Time in SM $\tau_i^{SM}(T)$, number of transitions $\rho_i(T)$, Acceleration Factor $AF_i^{TOT}(T)$,maintenance costs for each PS $i \in \mathcal{M}$ with $|\mathcal{M}| = 5$, $|\mathcal{N}| = 15$, $T = 1$ [year].

TABLE 5
Electricity Cost Breakdown, Total Electricity Costs, Total Maintenance Costs, Total Costs for OMEC, OC, and OEA vs. the variation of $|\mathcal{N}|$.

		Algorithms	$C_E^{PROC}(T)$ [$]	$C_E^{TR}(T)$ [$]	$C_E^{MIG}(T)$ [$]	$C_E^{TOT}(T)$ [$]	$C_M^{TOT}(T)$ [$]	$C^{TOT}(T)$ [$]
5 Servers	$|\mathcal{N}| =$ 11	OC	**1458.97**	656.80	443.64	2559.52	20930.16	23489.68
		OEA	1610.32	**643.50**	79.371	**2333.76**	1510.17	3843.93
		OMEC	1651.78	646.19	**78.620**	2377.15	**1082.66**	**3459.81**
	$|\mathcal{N}| =$ 13	OC	**1418.06**	449.47	1322.49	3188.66	23819.3	27002.96
		OEA	1482.57	**441.80**	129.57	**2053.05**	6051.31	8104.36
		OMEC	1835.97	564.84	**121.52**	2521.65	**2848.98**	**5370.63**
	$|\mathcal{N}| =$ 15	OC	**1703.24**	**1102.25**	315.251	**3121.48**	1859.57	4981.05
		OEA	1897.79	1134.01	158.09	3190.49	1740.59	4931.08
		OMEC	1916.37	1140.18	**157.36**	3214.56	**1510.92**	**4725.48**

strategies in terms of migrations, Fig. 2 reports the histograms of the occurrence of migrations events $\sum_{ijk} m_{ijk}(t) \quad \forall t \in \mathcal{T}$ (with $T = 12$ [months]). More in detail, a number of 0 migrations is counted for almost 20000 TSs for the three strategies, corresponding to 57% of the total time. We recall that we are considering a scenario in which the PSs are pretty loaded. Hence, it is not always possible to put PSs in SM. From the figure, we can see that OC tends to spread the occurrence of migration events, by requiring even the entire set of 15 VMs to be migrated in the same TS. This is due to the fact that this strategy does not consider the migration costs. Therefore, it can be very aggressive in terms of migrations that are concurrently performed. On the other hand, both OMEC and OEA are able to reduce the occurrence of migration events and also the maximum number of migrations, which is equal to 9 in this case.

At this point a natural question is then: what is the impact of the different strategies on the single PSs? In order to shed light on this issue, Fig. 3 reports the time in SM $\tau_i^{SM}(T)$, the number of transitions $\rho_i(T)$, the Acceleration Factor $AF_i^{TOT}(T)$, and the maintenance costs for each server $i \in \mathcal{M}$ by applying OC, OEA and OMEC with $T = 1$ [year]. All the figures report the y-axis in logarithmic scale for the sake of clarity. Focusing on $\tau_i^{SM}(T)$, we can clearly see from Fig. 3(a) that the amount of time in SM is notable increased in PS 5 with OC w.r.t. to the other strategies. However, this has a cost in terms of number of transitions $\rho_i(T)$, which is clearly higher for PS 5 with OC compared to OMEC and OEA, as reported in Fig. 3(b). Eventually, also high values of $\rho_i(T)$ are experienced with OC for PS 1,2,4, despite their relatively low values of $\tau_i^{SM}(T)$. As a result, high values of AF and maintenance costs are experienced by the PSs when OC is employed, as shown in Fig. 3(c) and Fig. 3(d), respectively. Moreover, while in general the amount of time in SM is pretty similar both with OMEC and OEA (Fig. 3(a)), the former generally performs better than the latter, by wisely limiting ρ_i for PSs 1,2,3,5 (Fig. 3(b)). Consequently, also the AF of OMEC is in general lower than OEA (Fig. 3(c)), thus bringing lower maintenance costs for all the PSs - except from PS 4 (Fig. 3(d)).

In the following, we investigate the impact of varying the number of VMs $|\mathcal{N}|$, while keeping the number of PSs fixed to $|\mathcal{M} = 5|$. Tab. 5 reports the results, which are obtained by running each strategy with $T = 1$ [year]. Interestingly, in all cases OMEC clearly outperforms both OEA and OC, by ensuring the lowest values of $C_E^{TOT}(T)$. More in depth, the gap between OMEC and OC in terms of total costs is more than 20000 [$] when $|\mathcal{N}|$ is equal to 11 and 13. In particular, as the number of VMs is decreased, OC exploits the fact that the PSs are less loaded in order to trigger more frequently the SM state on them. While this strategy is in general beneficial from the perspective of processing costs, which tend to be reduced, the maintenance costs $C_M^{TOT}(T)$ are negatively impacted. In addition, we can see

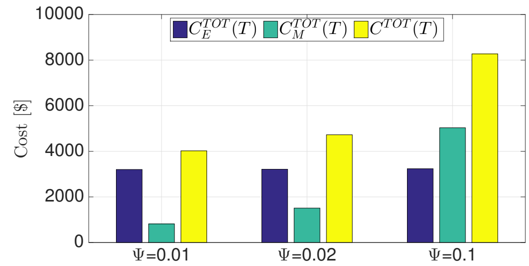
TABLE 6
Maintenance costs, electricity costs, and total costs for OMEC, OC, and OEA vs. the variation of $|\mathcal{M}|$.

Algorithms		$C_M^{TOT}(T)$ [\$]	$C_E^{TOT}(T)$ [\$]	$C^{TOT}(T)$ [\$]
$\vert\mathcal{M}\vert = 3$	OC	8466.39	1989.38	10455.77
	OEA	1723.44	1673.82	3397.26
	OMEC	**1195.11**	**1682.10**	**2877.21**
$\vert\mathcal{M}\vert = 5$	OC	1859.57	3121.48	4981.05
	OEA	1740.59	3190.49	4931.08
	OMEC	**1510.92**	**3214.56**	**4725.48**

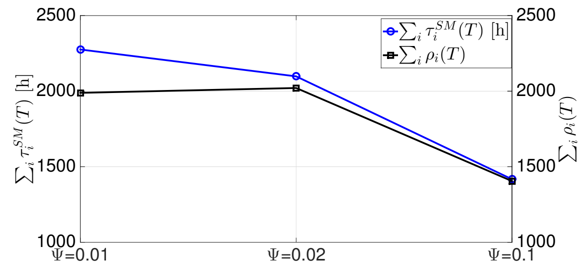
that also the OEA strategy performs consistently worse that OEA for all the values $|\mathcal{N}|$. On the other hand, the proposed OMEC solution is able to trade between the different costs, and to achieve the best performance in terms of total cost $C^{TOT}(T)$.

We then evaluate the performance of the algorithms when the number of servers $|\mathcal{M}|$ is varied. In order to provide a fair comparison, we set $|\mathcal{N}| = 3 \times |\mathcal{M}|$ for each considered value of $|\mathcal{M}|$. Tab. 6 reports the obtained results. Not surprisingly, the total energy costs $C_E^{TOT}(T)$ are increased as $|\mathcal{M}|$ is increased, for all the considered strategies. However, the maintenance costs $C_M^{TOT}(T)$ of OC tend to increase when $|\mathcal{M}|$ is decreased. By further investigating this issue, we have found that, when $|\mathcal{M}|$ is decreased, OC tends to change the power state to a smaller number of PSs, resulting in a large increase of their AFs, and consequently of $C_M^{TOT}(T)$. On the other hand, OMEC ensures the lowest values of $C^{TOT}(T)$ when $|\mathcal{M}|$ is decreased, by wisely changing the power states of the PSs.

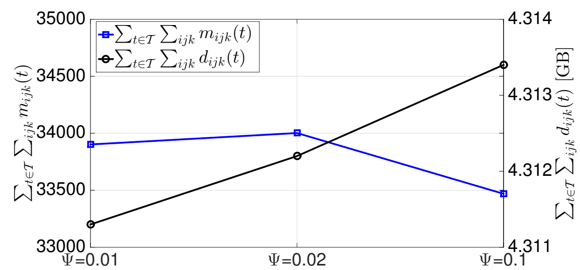
In the final part of our work, we have considered the impact of the Ψ_i parameter, which acts as a weight for the power state transitions $\rho_i(t)$ appearing in Eq. (12). Fig. 4 reports the results obtained from the variation of Ψ_i with the OMEC strategy. Fig. 4(a) reports the impact on the costs $C_E^{TOT}(T)$, $C_M^{TOT}(T)$, and $C^{TOT}(T)$. As expected, the total costs $C^{TOT}(T)$ are increased when Ψ_i is increased, due to the increase in the maintenance costs $C_M^{TOT}(T)$. To give more insight, Fig. 4(b) reports the time in SM $\sum_i \tau_i^{SM}(T)$ and the total number of transitions $\sum_i \rho_i(T)$. Clearly, when Ψ_i is increased, both $\sum_i \tau_i^{SM}(T)$ and $\sum_i \rho_i(T)$ tend to be reduced. Finally, Fig. 4(c) reports the impact of Ψ_i on the total number of migrations $\sum_{t\in\mathcal{T}} \sum_{ijk} m_{ijk}(t)$ and the total amount of transferred data $\sum_{t\in\mathcal{T}} \sum_{ijk} d_{ijk}(t)$. In particular, the total number of migrations tends to decrease when Ψ_i is increased, as a consequence of the decrease of $\rho_i(t)$. On the other hand, the amount of transferred data between VMs tends to increase when Ψ_i is increased.



(a) Electricity Costs $C_E^{TOT}(T)$, Maintenance Costs $C_M^{TOT}(T)$, and total costs $C^{TOT}(T)$



(b) Time in SM $\sum_i \tau_i^{SM}(T)$ and total number of transitions $\sum_i \rho_i(T)$



(c) Number of migrations $\sum_{t\in\mathcal{T}} \sum_{ijk} m_{ijk}(t)$ and amount of transferred data $\sum_{t\in\mathcal{T}} \sum_{ijk} d_{ijk}(t)$

Fig. 4. Impact of Ψ_i with OMEC $\mathcal{M} = 15$ and $\mathcal{N} = 5$.

8 Conclusion and Future Directions

We have investigated the problem of jointly managing the maintenance costs and the electricity consumption in a CDC by acting on the PSs power states and the VMs allocation. After reporting a model to compute the maintenance costs given the power (and consequently the temperature) variation between AM and SM states for each PS, we have introduced the electricity cost computation, which is based on the CPU processing, the amount of transferred data, and the VMs migrations. Consequently, we have optimally formulated the OMEC problem, which jointly takes into account the aforementioned terms. In addition, we have also detailed the linearization for the non linear constraints. Results, obtained over different scenarios, demonstrate that, at the end of the considered time period of 1 [year], OMEC outperforms both the OC and OEA strategies in all cases, thus always ensuring the lowest values of total costs. We believe that this work can be the first step towards research directions. More in detail, the definition of efficient heuristics, able to solve the problem for very large CDCs, can be an interesting direction. In addition, we plan also to solve the optimal problem by considering multiple TSs jointly together. Finally, the application of our approach to a set of CDCs, each of them subject to different electricity prices (e.g., due to different CDC locations), can be another interesting aspect to be investigated.

References

[1] H. Wells, *World Brain*. Methuen & Co., London, 1938.
[2] V. Bush *et al.*, "As we may think," *The atlantic monthly*, vol. 176, no. 1, pp. 101–108, 1945.
[3] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, *et al.*, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.
[4] P. Mell, T. Grance, *et al.*, "The nist definition of cloud computing," 2011.
[5] M. Hilbert and P. López, "The worlds technological capacity to store, communicate, and compute information," *science*, vol. 332, no. 6025, pp. 60–65, 2011.
[6] C. D. Patel and A. J. Shah, "Cost model for planning, development and operation of a data center," 2005.
[7] J. Koomey, "Growth in data center electricity use 2005 to 2010," *A report by Analytical Press, completed at the request of The New York Times*, vol. 9, 2011.

[8] K. Bilal, S. U. R. Malik, O. Khalid, A. Hameed, E. Alvarez, V. Wijaysekara, R. Irfan, S. Shrestha, D. Dwivedy, M. Ali, *et al.*, "A taxonomy and survey on green data center networks," *Future Generation Computer Systems*, vol. 36, pp. 189–208, 2014.

[9] S. Srikantaiah, A. Kansal, and F. Zhao, "Energy aware consolidation for cloud computing," in *Proceedings of the 2008 conference on Power aware computing and systems*, vol. 10, pp. 1–5, San Diego, California, 2008.

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

[11] C. Mastroianni, M. Meo, and G. Papuzzo, "Probabilistic consolidation of virtual machines in self-organizing cloud data centers," *IEEE Transactions on Cloud Computing*, vol. 1, no. 2, pp. 215–228, 2013.

[12] W. Fang, X. Liang, S. Li, L. Chiaraviglio, and N. Xiong, "Vmplanner: Optimizing virtual machine placement and traffic flow routing to reduce network power costs in cloud data centers," *Computer Networks*, vol. 57, no. 1, pp. 179–196, 2013.

[13] L. Chiaraviglio, P. Wiatr, P. Monti, J. Chen, J. Lorincz, F. Idzikowski, M. Listanti, and L. Wosinska, "Is green networking beneficial in terms of device lifetime?," *IEEE Communications Magazine*, vol. 53, no. 5, pp. 232–240, 2015.

[14] L. Chiaraviglio, N. Blefari-Melazzi, C. Canali, F. Cuomo, R. Lancellotti, and M. Shojafar, "A measurement-based analysis of temperature variations introduced by power management on commodity hardware," in *International Conference on Transparent Optical Networks (ICTON 2017)*, 2017.

[15] D. Frear, D. Grivas, and J. Morris, "Thermal fatigue in solder joints," *JOM*, vol. 40, no. 6, pp. 18–22, 1988.

[16] W. Lee, L. Nguyen, and G. S. Selvaduray, "Solder joint fatigue models: review and applicability to chip scale packages," *Microelectronics reliability*, vol. 40, no. 2, pp. 231–244, 2000.

[17] R. P. G. Muller, "An experimental and analytical investigation on the fatigue behaviour of fuselage riveted lap joints. the significance of the rivet squeeze force, and a comparison of 2024-t3 and glare 3," *Delft University of Technology*, 1995.

[18] J. Mi, Y.-F. Li, Y.-J. Yang, W. Peng, and H.-Z. Huang, "Thermal cycling life prediction of sn-3.0 ag-0.5 cu solder joint using type-i censored data," *The Scientific World Journal*, vol. 2014, 2014.

[19] M. Dayarathna, Y. Wen, and R. Fan, "Data center energy consumption modeling: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 732–794, 2016.

[20] W. Voorsluys, J. Broberg, S. Venugopal, and R. Buyya, "Cost of virtual machine live migration in clouds: A performance evaluation," in *IEEE International Conference on Cloud Computing*, pp. 254–265, Springer, 2009.

[21] J. Stoess, C. Lang, and F. Bellosa, "Energy management for hypervisor-based virtual machines.," in *USENIX annual technical conference*, pp. 1–14, 2007.

[22] H. Liu, H. Jin, C.-Z. Xu, and X. Liao, "Performance and energy modeling for live migration of virtual machines," *Cluster computing*, vol. 16, no. 2, pp. 249–264, 2013.

[23] G. Soni and M. Kalra, "A novel approach for load balancing in cloud data center," in *Advance Computing Conference (IACC), 2014 IEEE International*, pp. 807–812, IEEE, 2014.

[24] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," in *Proceedings of the 2010 10th IEEE/ACM international conference on cluster, cloud and grid computing*, pp. 826–831, IEEE Computer Society, 2010.

[25] J. Bi, Z. Zhu, R. Tian, and Q. Wang, "Dynamic provisioning modeling for virtualized multi-tier applications in cloud data center," in *Cloud Computing (CLOUD), 2010 IEEE 3rd international conference on*, pp. 370–377, IEEE, 2010.

[26] R. Han, M. M. Ghanem, L. Guo, Y. Guo, and M. Osmond, "Enabling cost-aware and adaptive elasticity of multi-tier cloud applications," *Future Generation Computer Systems*, vol. 32, pp. 82–98, 2014.

[27] J. L. Berral, R. Gavalda, and J. Torres, "Power-aware multi-data center management using machine learning," in *Parallel Processing (ICPP), 2013 42nd International Conference on*, pp. 858–867, IEEE, 2013.

[28] R. Ge, X. Feng, and K. W. Cameron, "Modeling and evaluating energy-performance efficiency of parallel processing on multicore based power aware systems," in *Parallel & Distributed Processing, 2009. IPDPS 2009. IEEE International Symposium on*, pp. 1–8, IEEE, 2009.

[29] S. L. Song, K. Barker, and D. Kerbyson, "Unified performance and power modeling of scientific workloads," in *Proceedings of the 1st International Workshop on Energy Efficient Supercomputing*, p. 4, ACM, 2013.

[30] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," *ACM SIGCOMM computer communication review*, vol. 39, no. 1, pp. 68–73, 2008.

[31] C. Guo, H. Wu, K. Tan, L. Shi, Y. Zhang, and S. Lu, "Dcell: a scalable and fault-tolerant network structure for data centers," in *ACM SIGCOMM Computer Communication Review*, vol. 38, pp. 75–86, ACM, 2008.

[32] A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, "Vl2: a scalable and flexible data center network," in *ACM SIGCOMM computer communication review*, vol. 39, pp. 51–62, ACM, 2009.

[33] C. Guo, G. Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C. Tian, Y. Zhang, and S. Lu, "Bcube: a high performance, server-centric network architecture for modular data centers," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 63–74, 2009.

[34] D. Kliazovich, P. Bouvry, Y. Audzevich, and S. U. Khan, "Greencloud: a packet-level simulator of energy-aware cloud computing data centers," in *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, pp. 1–5, IEEE, 2010.

[35] P. Gill, N. Jain, and N. Nagappan, "Understanding network failures in data centers: measurement, analysis, and implications," in *ACM SIGCOMM Computer Communication Review*, vol. 41, pp. 350–361, ACM, 2011.

[36] K. V. Vishwanath and N. Nagappan, "Characterizing cloud computing hardware reliability," in *Proceedings of the 1st ACM symposium on Cloud computing*, pp. 193–204, ACM, 2010.

[37] Q. Zhang, M. F. Zhani, M. Jabri, and R. Boutaba, "Venice: Reliable virtual data center embedding in clouds," in *INFOCOM, 2014 Proceedings IEEE*, pp. 289–297, IEEE, 2014.

[38] R. Jhawar and V. Piuri, "Fault tolerance management in iaas clouds," in *Satellite Telecommunications (ESTEL), 2012 IEEE First AESS European Conference on*, pp. 1–6, IEEE, 2012.

[39] J. Lau, W. Dankshen, and P. Vianco, "Acceleration models, constitutive equations, and reliability of lead-free solders and joints," in *Electronic Components and Technology Conference, 2003. Proceedings. 53rd*, pp. 229–236, IEEE, 2003.

[40] L. Chiaraviglio, D. Ciullo, M. Mellia, and M. Meo, "Modeling sleep mode gains in energy-aware networks," *Computer Networks*, vol. 57, no. 15, pp. 3051–3066, 2013.

[41] V. Eramo, E. Miucci, and M. Ammar, "Study of reconfiguration cost and energy aware vne policies in cycle-stationary traffic scenarios," *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, pp. 1281–1297, 2016.

[42] N. Bobroff, A. Kochut, and K. Beaty, "Dynamic placement of virtual machines for managing sla violations," in *Integrated Network Management, 2007. IM'07. 10th IFIP/IEEE International Symposium on*, pp. 119–128, IEEE, 2007.

[43] C. Canali, R. Lancellotti, and M. Shojafar, "A computation-and network-aware energy optimization model for virtual machines allocation," in *Cloud Computing and Services Science (CLOSER 2017), The 6th International Conference on*, pp. 1–11, 2017.

[44] P. Roubaud, G. Ng, G. Henshall, R. Bulwith, R. Herber, S. Prasad, F. Carson, S. Kamath, and A. Garcia, "Impact of intermetallic growth on the mechanical strength of pb-free bga assemblies," *Proceedings of APEX*, 2001.

[45] *EnergyStar*. https://www.energystar.gov/index.cfm?c=archives.en\-ter\-prise_servers, last accessed on 22th May 2017.

[46] *Cisco Data Sheet Switching Infrastructure*. http://www.cisco.com/c/en/us/products/collateral/switches/catalyst-2960-x-series-switches/data_sheet_c78-728232.html, last accessed on 22th May 2017.

[47] *Cisco Power Data Sheet*. http://blogs.cisco.com/enterprise/reduce-switch-power-consumption-by-up-to-80, last accessed on 22th May 2017.

[48] W. L. Winston, M. Venkataramanan, and J. B. Goldberg, *Introduction to mathematical programming*, vol. 1. Thomson/Brooks/Cole Duxbury; Pacific Grove, CA, 2003.

[49] S. Arrhenius, *Über die Reaktionsgeschwindigkeit bei der Inversion von Rohrzucker durch Säuren*. Wilhelm Engelmann, 1889.

[50] J.-P. Clech, D. M. Noctor, J. C. Manock, G. W. Lynott, and F. Baders, "Surface mount assembly failure statistics and failure free time," in *Electronic Components and Technology Conference, 1994. Proceedings., 44th*, pp. 487–497, IEEE, 1994.

[51] A. Marotta and S. Avallone, "A simulated annealing based approach for power efficient virtual machines consolidation," in *Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on*, pp. 445–452, IEEE, 2015.