Optimization of Data and Energy Migrations in Mini Data Centers for Carbon-Neutral Computing

Marcos de Melo da Silva, Abdoulaye Gamatié, Gilles Sassatelli, Michael Poss, Michel Robert

Optimization of Data and Energy Migrations in Mini Data Centers for Carbon-Neutral Computing

Marcos De Melo da Silva, Abdoulaye Gamatié, Gilles Sassatelli, Michael Poss and Michel Robert

Abstract—Due to large-scale applications and services, cloud computing infrastructures are experiencing an ever-increasing demand for computing resources. At the same time, the overall power consumption of data centers has been rising beyond 1% of worldwide electricity consumption. The usage of renewable energy in data centers contributes to decreasing their carbon footprint and overall electricity costs. Several green-energy-aware resource allocation approaches have been studied recently. None of them takes advantage of the joint migration of *jobs* and *energy* in green data centers to increase energy efficiency.
This paper presents an optimization approach for energy-efficient resource allocation in mini data centers. The observed momentum around edge computing makes the design of geographically distributed mini data centers highly desirable. Our solution exploits both virtual machines (VMs) and energy migrations between green compute nodes in mini data centers. These nodes have energy harvesting, storage, and transport capabilities. They enable the migration of VMs and energy across different nodes. Compared to VM allocation alone, joint-optimization of VM and energy allocation reduces utility electricity consumption by up to 22%. This reduction can reach up to 28.5% for the same system when integrating less energy-efficient servers. The gains are demonstrated using simulation and a Mixed Integer Linear Programming formulation for the resource allocation problem. Furthermore, we show how our solution contributes to sustaining the energy consumption of old-generation and less efficient servers in mini data centers.

Index Terms—Mini data center, distributed computing, carbon neutrality, renewable energy, resource allocation, optimization, energy-aware systems, workload allocation and scheduling

1 Introduction

Cloud computing and other large-scale applications and services have caused an increase in energy needs for infrastructures such as data centers over the past decade. According to [1], the annual energy consumption of data centers is estimated to be 200 terawatt-hours (TWh). This corresponds to around 1% of the worldwide electricity consumption [2] and 0.3% of global CO2 emissions. Given the rising energy demand in data centers, innovative technologies (e.g., hyperscale infrastructures) and renewable energies will become crucial. Major industrial actors such as Google, Amazon, and Facebook claim to operate carbon-neutral data centers thanks to Renewable Energy Credits [3], which are non-physical assets linked to renewable energy projects. Although this strategic incentive does contribute to developing renewables, it does not imply that data centers themselves are powered by renewables. Recently, however, Google announced its intention to match its global data center energy consumption to renewable energy production. Its ultimate objective is to make its data centers operate on decarbonized energy 24/7, 365 days a year [4]. Facebook declared in its report on sustainability that its global operations will be 100% supported by renewable energy in a few years [5]. Amazon has set the same goal for 2025; it plans to achieve net-zero carbon emissions by 2040 [6].

The above trend of incorporating renewable energies into the power supply mix of data centers will keep on developing. It does not only reduce the total power consumption, but also the carbon emissions. To successfully achieve this goal, the design of conventional grid-connected data centers must be revisited. The new designs should be robust to the intermittent nature of renewable energies while minimizing the use of utility electricity. They should also be scalable with respect to energy harvesting and workload execution capabilities. Finally, they should guarantee low response times to requests from client devices and users, as expected in the edge computing context.

1.1 Limitations of current approaches

A notable part of state-of-the-art approaches [7] [8] considers data center designs consuming power from both the utility grid and renewable sources. Each of the sources is connected to the data center infrastructure via its centralized entry point. Renewable energy is either directly used or stored in large-capacity batteries for later usage [9]. The key challenge consists in maximizing the use of renewable energy, while minimizing that from the utility grid. It is usually solved through various energy-aware scheduling techniques applied to tasks, workloads or virtual machines (VMs) [10]. In this paper, we claim that acting only upon mapping and scheduling of software objects (tasks, workloads or VMs) has limited optimization potential in terms of energy consumption.

- *Marcos De Melo da Silva, Abdoulaye Gamatié, Gilles Sassatelli, Michael Poss and Michel Robert are with LIRMM, Univ. Montpellier and CNRS, Montpellier, France.*
E-mail: marcos.de-melo-da-silva@lirmm.fr, abdoulaye.gamatie@lirmm.fr, sassatelli@lirmm.fr, michael.poss@lirmm.fr, michel.robert@umontpellier.fr

Manuscript received September 2021; revised May, 2022.

Indeed, data migrations required between computing nodes are often I/O intensive as they usually involve several operations on tasks and VMs: context saving before migration, transfer towards remote nodes, and context restoration before resuming execution on destination nodes. Beyond additional latencies, these operations incur a significant energy consumption overhead of the server network [11]. As shown in our study, combining energy migration with VM migration between distributed servers, equipped each with local energy harvesting and storage facilities, helps lowering the required brown or non-green energy consumption.

1.2 Our solution : distributed mini data centers

Mini data centers (1–10 racks within 1–25 square meters compute space [12]) are very promising solutions to meet the aforementioned requirements, i.e., energy-efficiency, scalability and low latency. They can execute up to 100 VMs each thanks to their efficient servers. Application domains typically include industrial automation, environmental monitoring, oil and gas exploration, and, in general, urban applications requiring local and real-time data processing [13].

Tens of such mini data centers can be deployed and interconnected at the city district level to form a powerful data center. They can operate through a dedicated electrical grid, and promote the use of renewable energy [14]. In the present work, we extend this concept by exploiting the opportunity to migrate both the workload and energy across different computing nodes.

We consider a novel design approach for green data centers, which is composed of distributed green nodes. The green nodes are interconnected by two kinds of links, as shown in Figure 1: i) a data link (in blue color), typically Gigabit Ethernet, used for task or VM migrations between the nodes; and ii) a power link (in red color), for energy transfer between green nodes connected by a power crossbar [15], [16]. The power crossbar is a central component in the design that makes it possible to control the electrical topology of the energy network via software.

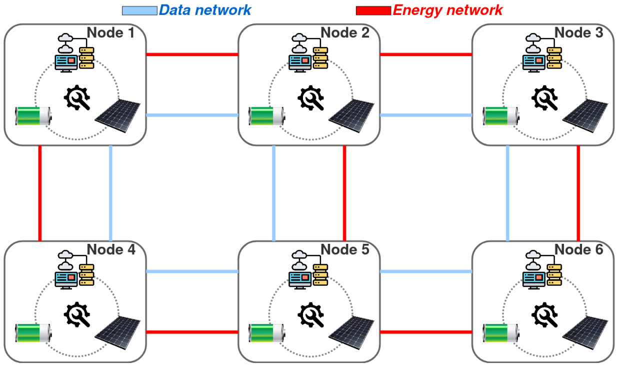


Fig. 1. Mesh network of green nodes in a mini data center.

Each green node includes a compute system, an energy harvester, a local energy storage, and a custom programmable power electronics system handling the energy transfer between the different nodes.

To demonstrate the relevance of our solution, we address the following questions:

- **Q1)**: *given a workload profile, how to dimension the main energy harvesting and storage components of our proposed system design to ensure its energy neutrality over a whole year?* Here, by energy neutrality, we mean how the non-green energy used by a system to execute a workload in scarce green energy periods can be compensated for by the surplus of green energy harvested in more favorable periods. This surplus can be typically re-injected into the grid when the energy storage is already full.
- **Q2)**: *how much non-green energy can be saved when executing typical data center workloads in our proposed system, i.e., supporting both data and energy migration between execution nodes?* The resulting non-green energy savings have a beneficial impact on the electricity bill of data centers. The related expense can reach several million dollars every year, representing a non negligible percentage of the data center exploitation costs. For this reason, electricity cost reduction for data centers is still a major challenge [17].
- **Q3)**: *how does these non-green energy savings vary according to i) different solar irradiation conditions (low versus high irradiation), and ii) according to less energy-efficient servers, typically relying on old-generation power management technologies?*

1.3 Our contribution

The contribution of the current work consists of an optimization approach to maximize the use of renewable energy by exploiting the unprecedented energy and data migration capability of the proposed design. We answer the aforementioned questions by characterizing the energy gains across different design scenarios.

- First, questions Q1) and Q2) are addressed by considering a battery and photovoltaic panel sizing model, detailed in Appendix A. This model is used to solve the energy optimization problem for four representative workload execution policies in the system: i) VM execution without any migration, ii) VM execution with data migration only, iii) VM execution with energy migration only, and iv) VM execution with both data and energy migration. These policies are compared according to their efficiency in terms of non-green energy usage, i.e., from the utility grid.
- To answer question Q3), we evaluate the above migration policies while considering low and high solar irradiation conditions used in the south of France. The irradiation data are obtained from a well-known database [18]. We also assess the same scenarios while modeling old-generation servers dissipating more static power due to inefficient power management mechanisms. Finally, we explore the impact of energy harvesting and storage resource reduction on the considered system. In particular, we reduced the solar panels and battery capacity by 25% in the initial energy neutral dimensioning.
- Given the outcomes of the above evaluations, we show that execution policies with energy migration can reduce by more than 50% the non-green energy usage over a year in favorable and realistic solar

irradiation conditions. This tendency is preserved with either server wear or a reduction by 25% of energy harvesting and storage resources w.r.t. a reference energy-neutral dimensioning. These results offer interesting insights into the trade-off between the cost and sustainability of data centers against their expected energy-efficiency after deployment.

1.4 Outline of this paper

The remainder of this paper is organized as follows. Section 2 presents some related work on green data centers and optimization techniques applied to the resource allocation issues. Section 3 provides more details on the design principles of our solution. It also deals with the modeling of the target distributed computing nodes for the optimization problem. Section 4 describes an application of Mixed Integer Linear Programming to formulate the energy-efficient resource allocation problem in our proposed system design. Section 5 evaluates the optimization solution through different use cases. Finally, Section 6 gives some concluding remarks and future works perspectives. Appendices A and B provide further technical details and complementary results.

2 Related work

We first discuss the management of green data centers. The covered literature focuses on maximizing the use of renewable energy over utility electricity, which is assumed to be partly produced by conventional fossil fuels. Then, a special focus is put on relevant optimisation techniques for efficient resource allocation in data centers.

2.1 Green data center management

A variety of power management techniques has been investigated for reducing the energy consumption of computing systems, from embedded multiprocessor systems to high-performance computing domains [10]. At the software level, these techniques cover workload allocation and scheduling policies as well as dynamic load balancing technologies. At the hardware level, well-established techniques include dynamic voltage and frequency scaling (DVFS), dynamic power management, etc. In the specific case of data centers, the reduction of the non-negligible cooling-related power has been also addressed [19], [20], [21].

The survey in [10] provides a comprehensive presentation of so-called green-energy-aware approaches. It distinguishes workload scheduling from VM management. The former approach focuses on job scheduling to find favorable conditions (electricity price, carbon emission level, renewable power availability) in a data center. Meanwhile, the latter approach leverages a virtualized environment through VM migration during execution. Our solution also applies to VM migration w.r.t. renewable power availability. It also integrates a supplementary dimension, i.e. energy migration, which contributes to reducing the overall energy cost.

Typical green-energy-aware approaches exploit task scheduling, as illustrated in [22] and [23]. In [22], a multi-objective co-evolutionary algorithm is advocated during the scheduling. It enables configuring the execution nodes with adequate voltage/frequency levels to match the renewable energy supply. In this way, authors try to maximize both the renewable energy utilization and the workloads quality of service (QoS), i.e., higher task execution throughput and lower energy consumption. In [23], a larger task scheduling problem is considered for data centers. An evolutionary multi-objective approach is applied to solve the problem when both computing and cooling infrastructures are controlled. The solution addresses three optimization dimensions: power consumption, temperature, and QoS.

The problem of VM allocation to servers has also been addressed by focusing on the server network activity [11]. The aim of this study is to reduce the number of active switches in the network and balance the data traffic flows. For this purpose, the relation between power consumption and data routing is exploited. In [24], another approach deals with energy-proportionality in large scale data centers by lowering the power consumption of data center networks while they are not being used. Different allocation policies have been evaluated and analyzed through simulations. An interesting insight from this study is that the size of the networks plays a central role in achieving energy-proportionality: the larger the data center networks, the greater the energy-proportionality.

In [8], the authors propose a methodology for operation planning to maximize the usage of locally harvested energy in data centers. They define a mechanism to shift the energy demand from low renewable energy production time slots to higher energy production ones. This reduces the power consumption from the utility grid. The shifting mechanism relies on energy production prediction according to the weather variation. The authors show their approach enables an increase in renewable energy usage by 12%. A similar study [7] recently shows that this usage can be increased by 10%, while utility electricity energy consumption can be reduced by 21%. It adopts a self-adaptive approach for resource management in cloud computing. In the above studies, the experimental results are obtained via simulation. More generally, access to suitable tools for studying data centers integrating renewable energy sources has been a real challenge. The most popular solutions include the research platforms proposed by Goiri et al. [25], [26], [27]. They mainly focus on solar energy.

While the aforementioned studies account for both grid power supply and renewable energy sources, other studies only consider the latter. For instance, in [28] the authors deal with independent task scheduling in computing facilities powered only by renewable energy sources. Using a Python-based simulation environment, they evaluate different scheduling heuristics within a predicted power envelope to minimize the makespan in multicore systems. In [29], a similar problem is addressed for data centers. A specific task scheduling module is defined, which aims to maximize QoS demands. It considers a so-called power envelope estimated from weather forecasts and states of charge of energy storage components. An interesting insight gained from this study is more power does not necessarily lead to better QoS, but knowing when the power is delivered is more relevant for better outcomes.

The zero-carbon cloud (ZCCloud) project [30] deals with the exploitation of the so-called ”stranded power”. This power is generated by renewable energy sources (e.g. wind, solar) when the harvested power exceeds power demand and cannot be stored by the grid due to limited storage capacity. Instead of discarding this power at the source, the project proposes an approach for using the stranded power, hence reducing the carbon footprint of cloud computing. Examples of issues investigated in the framework of ZCCloud are: compute load shifting to better leverage carbon-free energy, execution of applications under real-time requirements (e.g. virtual reality, distributed video analytics) on serverless cloud computing, and extension of computing hardware lifetime.

A noteworthy approach is presented in the Datazero project [9]. It aims at the zero-emission and robust management of data centers using renewable energy sources. Unlike the majority of existing approaches, Datazero advocates a separate optimization of design objectives: objectives of the IT services versus electrical management. A negotiation module is defined between both to find a satisfactory compromise with respect to their respective objectives and constraints, e.g., high availability of IT services under the erratic behavior of renewable energy sources. By doing so, the authors avoid a challenging global optimization problem.

2.2 Optimization for data center resource allocation

An important optimization problem in cloud data centers is the *consolidation* of VMs to physical servers. It consists in placing VMs in as few servers as possible and putting in sleep mode or shutting off idle servers. This reduces the global power consumption without sacrificing QoS. The objective is to enable high performance while ensuring that *Service Level Agreement* (SLA) levels are met and operational costs are minimized. Approaches for the VM placement need to effectively answer the following questions: i) which node(s) should host new VMs as well as VMs that are being migrated? ii) when should a VM be migrated? iii) which VMs should be migrated? The last two questions are addressed by techniques that detect underutilized and overloaded servers [31]. Finally, tackling the first question requires the solution to an optimization problem involving the allocation of limited server resources to the VMs.

The VM placement problem is usually formulated as a *multi-dimensional bin-packing problem*, where servers are modeled as bins and VMs as items. Multiple server resources, e.g., CPU, memory, disk space, network bandwidth, are allocated to the VMs. Despite their limited practical applicability, some authors have proposed exact approaches based on Integer Linear Programming (ILP) to solve the problem [32], [33]. However, given the dynamic nature of the problem, most approaches are heuristic algorithms. Among those, [34], [35], [36] developed algorithms based on classical bin-packing heuristics, e.g., *first-fit decreasing* and *best-fit decreasing*. Meanwhile, as mentioned in the previous section, [22], [23] investigated evolutionary algorithms. We refer the interested reader to recent surveys [31], [37], [38] for a more exhaustive literature coverage.

When comparing the existing ILP-based resource allocation solutions with ours, we observe that the adopted formulations are often presented with the purpose of describing the problem. The formulations are actually not solved in practice. Instead, the authors prefer to use heuristic procedures. The main reason is the higher execution times required by ILP solvers. In addition, such formulations only tackle the problem of assigning VMs to servers and allocating the necessary resources at a given moment. They do not take into account how the resource utilization change over time. This last observation can be applied to the heuristic approaches as well. In our case, we need to properly model and simulate the whole computing infrastructure so that we are able to capture its dynamic behaviour, i.e., how the resource utilization and server states evolve over time, which implies solving the problem for an extended period of time. For that, we employ a time-indexed formulation that not only performs the VMs placement and tracks resources, but also models energy consumption and its flow.

2.3 Summary

Table 1 summarizes some relevant features of the discussed studies in comparison with the present work. While our approach aims at leveraging renewable energy sources in mini data centers, it fundamentally differs from the above studies in its additional optimization dimension brought by energy transfer. The ability to trigger on-demand energy transfers between distributed nodes is an important lever (beyond data/workload migrations) for achieving the best possible energy-efficient trade-offs depending on the node requirements. This enables us to propose an optimization model capable of finding the most favorable execution of the system. By applying suitable data and energy migrations between the nodes, we seek to minimize utility power consumption, up to solely using renewable energy for system execution. This matches expectations in mini data centers [14]. In a seminal work [39], we leveraged the energy migration principle to address the formal modeling and analysis of a safety-critical application on a multicore embedded system, under energy neutrality conditions. The present work rather focuses on the optimization problem of resource allocation for a different kind of system.

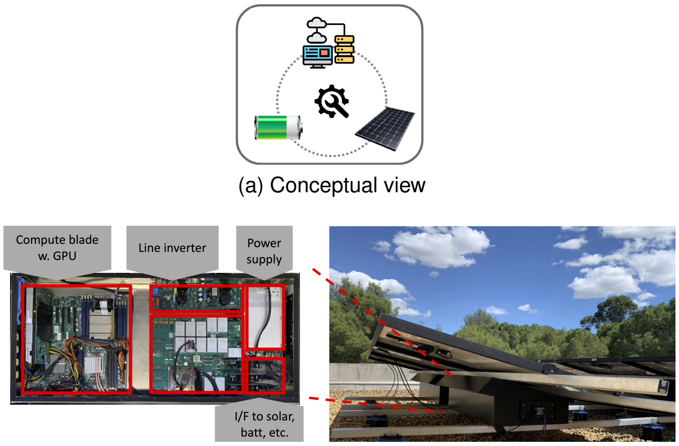
3 Design principles of our proposal

Our energy-neutral system design consists of n interconnected green nodes, $N = \{1, \ldots, n\}$. Each green node (see Figure 2) includes a *computing system* such as a server blade, an energy harvester consisting of photo-voltaic (PV) *solar panels*, a *battery* for local energy storage, and a logic board for managing the energy generation and storage, as well as the transfer of energy between nodes.

A node operates primarily on the harvested energy. In periods of low solar irradiation (e.g., night time, cloudy and rainy days) in which the average energy demand is higher than PV production, nodes consume the energy accumulated in their batteries. In the event that a node has a near-empty battery, which prevents continuing operation, it can either transfer its workload to other green nodes or fetch energy from remote green nodes (see Figure 3). Nodes can therefore wire power ports together (or conversely isolate), thereby connecting electrically remote components, e.g. a given node’s battery with a distant node’s compute system. This, in essence, means that energy can be migrated, i.e.

TABLE 1
Summary of related work on green data center management

	Renewable energy	Task–Job–VM scheduling	Power scheduling	Cooling power reduction	Simulation approach	Hardware prototype	ILP optim.	Heuristics / Machine learn.	Distributed energy transfer
Xu et al. (2020) [7]	+	+			+	+	+		
Cioara et al. (2015) [8]	+	+		+	+			+	
Pierson et al. (2019) [9]	+	+	+		+		+	+	
Kong et al. (2012) [10]	+	+		+	+	+	+	+	
Wang et al. (2014) [11]		+			+		+		
Abbasi et al. (2012) [19]		+		+	+			+	
Ganesh et al. (2013) [20]			+	+	+				
Li et al. (2020) [21]				+	+			+	
Lei et al. (2016) [22]	+	+			+			+	
Nesmachnow et al. (2015) [23]	+	+			+			+	
Goiri et al. (2013) [25] Goiri et al. (2014) [26] Goiri et al. (2015) [27]	+	+	+		+	+	+		
Kassab et al. (2017) [28]	+	+			+			+	
Caux et al. (2018) [29]	+	+			+			+	
Chien et al. (2015) [30]	+	+			+	+			
Ismaeel et al. (2018) [31]		+			+		+		
Ruiu et al. (2017) [24]		+			+			+	
Hwang et al. (2013) [32]		+			+		+	+	
Tseng et al. (2015) [33]		+			+		+		
Beloglazov et al. (2011) [34]		+			+			+	
Jangiti et al. (2020) [35]		+			+			+	
Liu et al. (2020) [36]		+			+			+	
This work	+	+	+	+	+	+	+	+	+



(a) Conceptual view

(b) On-roof prototype view

Fig. 2. Green node: server + batteries + solar panels + control logic.

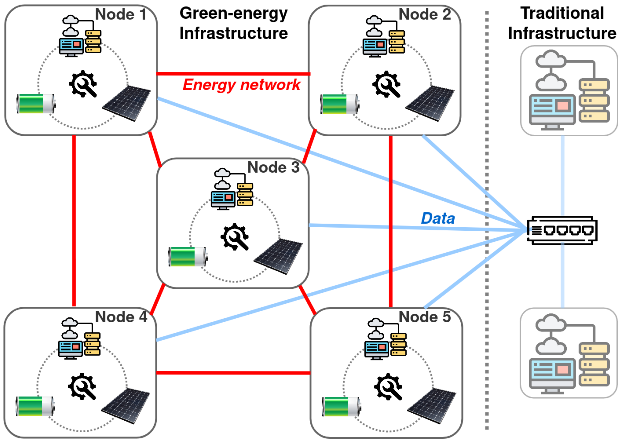


Fig. 3. Simulated infrastructure for energy-neutral distributed computing.

the power can be either supplied remotely, or transferred and stored before use. This is a main difference with energy packet networks [40], which support the latter option only.

As last resort, in case none of the previous actions are possible, the nodes will be forced to purchase energy from the utility grid to which they are connected. We assume that the nodes are connected through Ethernet wires to a switch (see Figure 3). This allows inter-node communication and ensures the connectivity to the existing computing and storage infrastructure, e.g., database servers, file servers, cloud managing servers.

The above energy-neutral system operates outdoor, for instance placed on a rooftop for maximum solar irradiation. The outdoor installation alongside the rather low compute density (required for matching harvesting and compute power consumption) makes for a cooling-friendly design, in contrast to conventional data centers which require heavy climate control equipment (HVAC). Experiments show that even under high outside temperature a clever node thermal design (using the enclosure as heatsink and having proper positioning of vents) alongside few temperature-controlled fans enables the system to maintain operation at temperatures below 70°C under heavy stress.

In addition, the advocated design inherently favors a modular system extension. Any new green node is inserted locally, thereby reducing drastically the necessary system-wide modifications. Finally, a failure of any green node could be easily bypassed through data or energy re-routing within the networked system, according to its topological connections. This naturally increases the resilience of the whole system [16].

3.1 Modeling the system behavior

Beyond the physical infrastructure and its various components, i.e., the static elements of our design, we also need to model their behaviour and how they evolve over time, i.e., the dynamics of the system.

Indeed, each component mentioned above presents an interdependent dynamic behaviour, for instance,

- the amount of energy generated by the PV panels varies with the solar irradiation levels, which depends on weather conditions, time of day, and season;
- the amount of energy in the batteries varies as it is consumed by the computational system or is charged by the PV panels; and finally,
- the amount of energy drained by the computing elements changes in response to the workload.

To represent this inherent dynamic behaviour of the system, we define a planning time horizon H, which we discretize into T time steps. Each time step corresponds to an interval of τ seconds.

In the following sections we provide further details on the elements that compose the proposed green computing infrastructure, as well as introduce some of the notation used in the remainder of the paper.

3.2 Modeling of solar panels and batteries

The amount of energy generated by the solar panels is influenced both by the weather conditions (which affect the local solar irradiation levels) and the physical characteristics of PV panels, e.g., power conversion efficiency, dimensions, etc. We apply the following equation to compute the energy, in Watts-hour (Wh), produced in node $i \in N$ by ρ_n solar panels, each one having an area of ρ_s m^2 and conversion efficiency of ρ_e, during time step $t \in H$, with a solar irradiation of ι^t W/m^2:

$$G_i^t = \iota^t \times (\rho_e \times \rho_n \times \rho_s) \times (\tau/3600) \qquad (1)$$

The energy generated by the energy harvester is used primarily to power the computational system. The production surplus is stored in the batteries, whose capacity is equal to U_i for each node $i \in N$. Furthermore, in order to avoid excessive battery wear, we ensure that batteries cannot be discharged below a safety level of $L_i = 0.15 \times U_i$, i.e., we always keep the batteries charged at least 15% of their maximum capacity.

3.3 Modeling of energy migration efficiency

The migration of energy between nodes involves a chain of different power electronics components (e.g., several DC/DC converters, wires) with variable efficiencies (DC/DC converters) and losses (wires resistances, solid-state MOSFET switches, etc.). In our physical system prototype, all unit efficiencies have been measured so as to be able to accurately model energy transfer losses for any arbitrary path in the system.

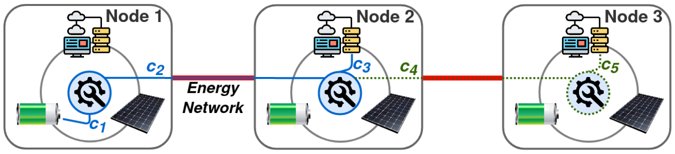


Fig. 4. Energy migration paths among green nodes.

Figure 4 depicts two possible paths when transferring energy between two nodes. In our physical system implementation, the *direct connection path* ($c_1 \rightarrow c_2 \rightarrow c_3$), in which the energy stored in the battery of *node 1* is transferred directly to supply the computational module of *node 2*, has an efficiency of 85.8%, which is rather high in a source-to-sink configuration as typical industrial grade AC/DC PSUs achieve similar efficiency at node-level only. The *indirect connection path* ($c_1 \rightarrow c_2 \rightarrow c_4 \rightarrow c_5$), in which the energy stored in the battery of *node 1* is transferred to supply the computational module of *node 3* using *node 2* as intermediary, has efficiency of 84.3%. Furthermore, for each additional intermediary node the efficiency drops by 1.8%. Overall, the actual energy migration capability incurs little additional losses compared to node-local functioning, which itself has efficiency similar to that of conventional compute nodes.

3.4 Modeling of computing resources and workloads

The computing system installed in a green node is characterized by its available processing, storage and network resources and an idle power consumption and a dynamic power consumption that vary with the computational load being executed. Each node $i \in N$ has R_i^M MB of RAM, R_i^D GB of disk storage, a network bandwidth of R_i^B Mb/s, and CPU load capacity R_i^C. Please note that instead of representing the CPU resource capacity in MIPS (million instructions per second), as in [34], [36], we define an utilization ratio in the interval $[0.0, 1.0]$, where 0.0 means that the machine is idle and 1.0 means that the CPU is 100% utilized. Furthermore, for the sake of simplicity, this utilization measure is not applied in a per-core basis, but for the whole processing unit. As a consequence, a VM can utilize 100% of all the cores available in a computing node. With respect to the energy consumption of a node, for a given idle and full load power profile and a time interval of τ seconds, the idle energy consumption is equal to ε_I Wh and, the dynamic consumption is equal to ε_P Wh. We note that the green nodes considered in a mini data center can be either homogeneous or heterogeneous in terms of compute resources, solar panel and battery capacities.

The computational workload to be executed on the green nodes consists of m VMs, $J = \{1, \ldots, m\}$. Each VM $j \in J$ has a requirement in terms of processing capacity, as well as memory, disk, and network bandwidth resources. We denote V_j^r, for $r \in \{M, D, B\}$, the memory, disk, and bandwidth resources, respectively, required by VM $j \in J$.

As for the CPU resources, the amount of computational work carried by each VM changes over time; hence, we have that C_j^t is the average CPU load imposed by VM $j \in J$ during time step $t \in H$.

4 MILP formulation for the Energy-Efficient Resource Allocation Problem

The energy-neutral resource allocation optimization problem we need to solve concerns the distributed computing infrastructure designed according to the principles described in the previous section. It consists in allocating m VMs to n green-energy nodes, such that the processing, memory, disk, and bandwidth resource demands of each VM are met without overloading the machines. Nodes are allowed to share energy among themselves or buy it from the utility in order to process their workloads. Acceptable levels of QoS are ensured by allowing VMs to be migrated between nodes. The objective is to minimize the amount of non-green energy bought from the utility grid and avoid energy waste by performing unnecessary VMs and energy migrations between nodes.

In this section we propose a Mixed Integer Linear Programming (MILP) [41], [42] formulation for the resource allocation problem. We first present our working assumptions. Next, we summarize the model parameters, define the decision variables and introduce the mathematical formulation of the problem. Then, we describe the energy cost estimation related to VM migrations used within the model. Finally, we explain how different variants of the problem can be obtained by incorporating or eliminating certain families of constraints from the formulation. In addition, we provide implementation details and solution approaches.

4.1 Working assumptions

Considering the dynamic nature of the modeled system and the need to represent how its component's states change over time, we formulate the resources allocation problem as a time-indexed MILP. To complement the problem description, and for ease of presentation, we list below our working assumptions on the system's components.

1) We model the changing aspects of the system by considering the optimization problem over a planning *horizon* $H = \{1, \dots, T\}$.
2) The VMs exist during the whole planning horizon. Those that are not performing any work are assumed to be in an idle state in which they do not consume resources.
3) The CPU utilization C_j^t of each VM $j \in J$ is known $\forall t \in H$.
4) The dynamic energy consumption of each node is based on CPU utilization level, which is directly affected by the computational load of the VMs assigned to the node.
5) The *initial energy* I_i stored in the battery of each node $i \in N$ is known.
6) The *safe discharge limit* L_i and the *maximum storage capacity* U_i of the battery installed in the node $i \in N$ are known.
7) The *energy gain* G_i^t of each node $i \in N$ is known $\forall t \in H$.
8) The *efficiency when transferring energy* among nodes E_{ik}, $\forall i, k \in N$ is know.
9) VMs can be *migrated* among nodes, and the energy cost for the source and target servers are μ_s and μ_d, respectively. In other words, the costs of VM context saving and restoring during VM migrations are captured. We do not take into account the energy consumed by switches and other network equipment, as it would complicate the problem even further. ILP optimization provides high-quality results at the expense of high computation time. It is a good fit for moderate complexity problems, which correspond to the mini data centers and workloads targeted in this paper (5 nodes and 25 VMs). Larger scale systems are more tractable with heuristics as found in some of related works.

4.2 Problem formulation

We summarize in Table 2 the parameters tied to the system's components described in the previous sections, and introduce the sets and new parameters that are specific to our formulation.

TABLE 2
Sets and parameters used in the problem formulation

Sets	Description
H	Planning horizon
T	Number of time step in planning horizon H
N	Set of nodes
J	Set of VMs
R_i	Resources set of node $i \in N$
V_j	Resource requirements set of VM $j \in J$
Parameters	Description
C_j^t	CPU utilization of VM $j \in J$ at time step $t \in H$
E_{ik}	Energy transfer efficiency between nodes $i, k \in N$
G_i^t	Energy generated in node $i \in N$ at time step $t \in H$
I_i	Initial amount of energy stored in node $i \in N$
L_i	Safety discharge level of battery in node $i \in N$
U_i	Maximum capacity of battery in node $i \in N$
ε_I	Node's idle energy consumption
ε_P	Additional energy consumption when node's at 100%
λ	Energy loss for transferring 1 Wh between nodes
μ	Total energy cost for migrating a VM
μ_d	Target server energy cost for migrating a VM
μ_s	Source server energy cost for migrating a VM
ν	Penalization for server CPU overloading
ϕ	Energy loss for injecting 1 Wh in the utility grid

Decision variables. We define the following decision variables:

- $x_{ij}^t = \begin{cases} 1, & \text{if VM } j \in J \text{ is running on node } i \in N \\ & \text{during time step } t \in H. \\ 0, & \text{otherwise} \end{cases}$
- $z_{ikj}^t = \begin{cases} 1, & \text{if VM } j \in J \text{ is transferred from node} \\ & i \in N \text{ to node } k \in N \text{ at the beginning} \\ & \text{of time step } t \in H. \\ 0, & \text{otherwise} \end{cases}$
- $f_{ik}^t \geq 0$, indicates the amount of energy transferred from node $i \in N$ to node $k \in N$ during time step $t \in H$.

- $L_i \leq w_i^t \leq U_i$, indicates the level of energy on the battery of node $i \in N$ at the end of time step $t \in H \cup \{0\}$.
- $b_i^t \geq 0$, indicates the amount of energy bought by node $i \in N$ during time step $i \in H$.
- $q_i^t \geq 0$, indicates the amount of energy injected into the utility grid by node $i \in N$ during time step $i \in H$.
- $v_i^t \geq 0$, measures the amount of CPU capacity overload in node $i \in N$ during time step $i \in H$.

Objective function and constraints. Our mathematical formulation consists of the objective function defined below:

$$\begin{aligned}\text{Minimize } \lambda \sum_{t\in H}\sum_{i,k\in N} f_{ik}^t + \mu \sum_{t\in H}\sum_{i,k\in N}\sum_{j\in J} z_{ikj}^t \\ \sum_{t\in H}\sum_{i\in N} b_i^t + \phi \sum_{t\in H}\sum_{i\in N} q_i^t + \nu \sum_{t\in H}\sum_{i\in N} v_i^t\end{aligned} \quad \text{(Obj1)}$$

which is subject to the following constraints:

$$\begin{aligned} w_i^t = {} & w_i^{t-1} + b_i^t + G_i^t + \sum_{k\in N} E_{ki} f_{ki}^t \\ & - \sum_{k\neq i\in N} f_{ik}^t - \mu_s \sum_{k\in N}\sum_{j\in J} z_{ikj}^t - \mu_d \sum_{k\in N}\sum_{j\in J} z_{kij}^t \\ & - (\varepsilon_I + \varepsilon_P \sum_{j\in J} C_j^t x_{ij}^t) - q_i^t \qquad \forall t\in H,\ i\in N \end{aligned} \quad (2)$$

$$\begin{aligned} w_i^0 &= I_i && \forall i \in N && (3)\\ \sum_{j\in J} C_j^t x_{ij}^t &\leq R_i^C + v_i^t && \forall t\in H,\ i\in N && (4)\\ \sum_{i\in N} x_{ij}^t &= 1 && \forall t\in H,\ j\in J && (5)\\ z_{ikj}^t &\geq x_{ij}^{t-1} + x_{kj}^t - 1 && \forall t\geq 2,\ j\in J,\ i\neq k\in N && (6)\\ L_i \leq w_i^t &\leq U_i && \forall t\in H,\ i\in N && (7)\\ x_{ij}^t,\ z_{ikj}^t &\in \{0,1\} && \forall t\in H,\ j\in J,\ i\neq k\in N && (8)\\ b_i^t,\ q_i^t,\ v_i^t,\ f_{ik}^t &\geq 0 && \forall t\in H,\ i\neq k\in N && (9) \end{aligned}$$

The objective function (Obj1) seeks to minimize the energy losses incurred when performing energy and VM migrations among nodes, the total amount of energy bought from the utility grid, the energy losses associated with the surplus energy generated that is injected into the utility grid, and the penalties for over-utilization of processing resources, respectively.

Constraints (2)-(7) model the characteristics of the problem related to resources allocation/VM scheduling, battery charge levels, energy generation/consumption, and energy flow conservation. More specifically, constraints (3) set up the initial state of each node's battery, i.e., the batteries charge levels at the beginning of the simulation period. Similarly, constraints (2) define how the charge of the batteries vary between two consecutive time steps ($\forall t \in H,\ t \geq 1$) by taking into account the batteries state in the previous time step ($t-1$) and the flow of energy from other sources during the current time step (t), i.e., the energy generated by the solar panels, the energy transferred among nodes, the energy bought from and inject in the grid, and the energy utilized by the computational workload.

CPU resource allocation is modeled by constraints (4). Note that we are not explicitly enforcing the allocation of other computational resources as memory, disk and network bandwidth. The reason is that we are interested mainly in correctly modeling the additional energy consumption due to increased CPU utilization. Nevertheless, these additional resources can be easy incorporated by adding the following constraints:

$$\sum_{j\in J} V_j^r x_{ij}^t \leq R_i^r \qquad \forall t\in H,\ i\in N,\ r\in\{M,D,B\} \quad (10)$$

The scheduling of the VMs to nodes is ensured by constraints (5), i.e., they ensure that during all time steps $t \in H$, each VM $j \in J$ should be assigned to one of the computational nodes $i \in N$. The number of VM migrations among nodes is accounted for in constraints (6), which checks whether a VM changes from computational node between time steps $t-1$ and t. The batteries' maximum capacity and discharge safety levels are enforced by constraints (7). Finally, constraints (8)-(9) define the domains of the variables.

In the energy flow conservation constraint (2), a node's computational energy consumption ($\varepsilon_I + \varepsilon_P \sum_{j\in J} C_j^t x_{ij}^t$) is described using the model proposed by [43]. This model assumes that the server power consumption and CPU utilization have a linear relationship.

4.3 Estimating the energy cost of VM migration

Several authors have proposed models for estimating the energy cost of migrating VMs in cloud environments [44], [45], [46], [47]. Such models present varying levels of precision and modeling complexity, with the more descriptive and complex ones achieving better precision at the expense of extra parameter estimation and model tuning efforts [47].

Due to its simplicity and reasonable precision, we chose to implement the model by [45]. As pointed out by the authors, VM migration is an I/O intensive task and also the most energy expensive one when transmitting and receiving data over the network. Indeed, their model is based on the assumption that the energy cost of performing VM migrations can be determined by the amount of data that is transferred during the migration process. The energy consumption by the source and destination hosts increases linearly with the network traffic, as described in the following equation:

$$E_{mig} = E_{sour} + E_{dest} = (\gamma_s + \gamma_d)V_{mig} + (\kappa_s + \kappa_d) \quad (11)$$

where E_{sour} is the energy consumed by the source host and E_{dest} is the energy (in joules) spent by the destination host for transferring V_{mig} megabytes of data. γ_s, γ_d, κ_s, and κ_d are model parameters to be trained. Equation (11) can be further simplified if both source and destination hosts are homogeneous:

$$E_{mig} = E_{sour} + E_{dest} = \gamma V_{mig} + \kappa \quad (12)$$

Then, the MILP formulation (Obj1), (2)-(9) parameters related to VM migration are defined as $\mu = E_{mig}$, $\mu_s = E_{sour}$, and $\mu_d = E_{dest}$. In addition, in our simulations we use $\gamma = 0.512$, $\kappa = 20.165$, as in [45].

4.4 Resource problem variants and implementation

The formulation (Obj1), (2)-(9) can be adjusted to tackle different variants of the resource allocation problem in our proposed distributed infrastructure. In the following, we describe how these variants are obtained and how we solve them.

Resource allocation problem variants. By adding or removing variables and constraints from the model (Obj1), (2)-(9) such that energy/data migrations are permitted or not generates four variants of the problem:

- *Energy+Data Migrations*: this is the default formulation and includes all variables, objective function (Obj1) and constraints (2)-(9).
- *Data Migration Only*: this variant is obtained by removing variables $f_{ik}^t, \forall t \in H,\ i, k \in N$. This model is a suitable baseline for the majority of state-of-the-art solutions.
- *Energy Migration Only*: in this variant no data migration is allowed. This is defined by removing the variables $z_{ikj}^t, \forall t \in H,\ i, k \in N,\ j \in J$ and constraint (6), while adding the binary variables $y_{ij}, \forall i \in N,\ j \in J$, and the following constraints:

$$\sum_{i \in N} y_{ij} = 1 \qquad \forall j \in J \tag{13}$$

$$\sum_{t \in H} x_{ij}^t = T y_{ij} \qquad \forall i \in N,\ j \in J \tag{14}$$

  which together ensure that each VM is assigned to a node for the whole planning horizon. Each variable y_{ij} is equal to 1 if VM $j \in J$ is assigned in to node $i \in N$, 0 otherwise.
- *No Migration*: this is the simplest one. It includes the two previous modifications, i.e., removing variables f_{ik}^t, and $z_{ikj}^t, \forall t \in H,\ i, k \in N,\ j \in J$, and constraints (6), while adding variables $y_{ij}, \forall i \in N,\ j \in J$ and constraints (13)-(14).

Rolling-Horizon heuristic. While the scenarios with no migrations or energy migration only can be solved in a couple of hours when simulating an infrastructure with 5 nodes and 25 VMs for a planning horizon of one week, the solution times become prohibitively high for the other two scenarios which involves data migration. The reason is the large number of variables $z_{ikj}^t, \forall t \in H,\ i, k \in N,\ j \in J$ and constraints (6) being generated. To cope with such large models, we decided to apply a rolling-horizon heuristic approach [48], [49]. This heuristic consists in splitting the planning horizon in smaller pieces, and solving them sequentially. We call each planning horizon fragment a *frame*.

Implementation details. The formulations, heuristic and other algorithms were coded in Julia[1] (version 1.5.4) using the embedded modeling language for mathematical optimization JuMP[2] [50] and executed on an Intel® Xeon® 2.2 GHz CPU, with 64.0 GB of RAM running under GNU/Linux Ubuntu 14.04 (*kernel* 4.4.0). Gurobi[3] 9.0.3 was used as the LP and MILP solver. Four computation threads were used when simulating each scenario. In our rolling-horizon heuristic, each *frame* has 24 time steps.

We present in Figure 5 the average computational times obtained when solving the proposed formulations, either by using the rolling-horizon heuristic or applying the MILP solver directly, for all the case studies analysed in the next sections. As expected, the scenarios without any migrations are the fastest, and the optimal solutions are obtained in less than 10 minutes. Next, using the rolling-horizon heuristic, on average one hour is needed to compute the solutions for the scenarios with energy and data migrations. In these two scenarios, no significant variation can be observed for different periods of the year. The most time consuming scenarios are those with data migration only. On average, five hours of computation using the rolling-horizon heuristic are necessary for a complete solution. The scenarios with energy migration only take on average two hours to prove the optimality of solutions. Those with both energy and data migrations require four hours. We note that the solutions in the former scenarios are optimal, while those in the latter are heuristic, i.e., approximate and not necessarily optimal.

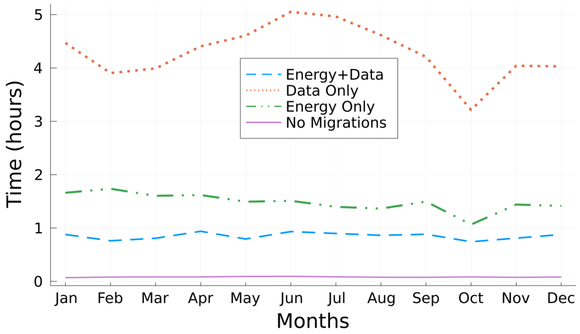


Fig. 5. Average solution times of the four formulations considering all study cases.

5 Case Study

In this section, we evaluate our resource allocation solution through different use cases[4]. We first describe the experimental setup. Then, we evaluate the gains in non-green energy enabled by four execution policies.

5.1 Experimental setup

For our case studies, we simulated a mini data center consisting of 5 nodes, whose maximum power consumption is 500 W and the idle power consumption is either 50 W or 100 W. The nodes are connected as depicted in Figure 3. We selected 5 nodes in our experiments for simulation complexity reasons. This is due to the costly constraints resolution induced by the applied global ILP optimization problem. The other infrastructure and formulation parameter values are described in Table 3.

Beyond the above 5 nodes, a centralized computer is used to perform the ILP solver. The cost of this computer is considered as constant and is ignored in the remainder of this case study.

Irradiation and VMs CPU utilization data. Historical irradiation data was retrieved with the aid of the European

[1] https://julialang.org
[2] https://jump.dev/
[3] https://www.gurobi.com/

[4] Some complementary evaluation scenarios of the use cases are presented in Appendix B.

TABLE 3
Parameter values used in our simulations

Model parameters	Values
n	5 nodes
m	25 VMs
τ	5 minutes
T	2016 time steps (7 days)
V_{mig}	8192 MB (*Equation 12*)
λ	$1 - (\sum_{i=1}^{n} \sum_{k=1,k\neq i}^{n} E_{ik})/n(n-1)$
μ	E_{mig} (*Equation 12*)
μ_d	E_{dest} (*Equation 12*)
μ_s	E_{sour} (*Equation 12*)
ν	1000
ϕ	0.5
Nodes parameters	**Values**
ε_I	4.17 Wh (50 W)
ε_P	37.50 Wh (500 W)
I_i	$0.5 \times U_i, \forall i \in N$
PV panels[1] parameters	**Values**
ρ_s	$1.59\ m^2$
ρ_e	0.19

([1]) Corresponds to panels NeON 2 by LG, model LG325N1C-A5.

Photo-voltaic Geographical Information System (PVGIS) [18]. The hourly irradiation datasets for Montpellier, France (Lat. 43.611, Long. 3.876) for the period 2005-2016 (Database PVGIS-SARAH, all available years) were used to compute the energy generated by each node. We also performed additional studies for a location in Africa: Mali (Lat. 17.159, Long. -3.340). Please note that the irradiation data points in these datasets have a one hour interval to make it compatible with our time step interval τ of 5 minutes we used piece-wise linear interpolation.

The workload of the VMs are simulated using real VMs workload traces from the CoMon project, a monitoring infrastructure for PlanetLab [51]. We used the same traces dataset[5] as [34], [36]. This traces dataset consists of the CPU utilization by a few thousand VMs from servers located at more than 500 places around the world that were recorded in the period of Mars and April of 2011. Each data point in a VM trace corresponds to a 5 minutes interval of utilization measurement.

From the thousand of traces available in this dataset, we selected 25 for which the associated VMs are active for at least the simulated time horizon, i.e., 7 days (2016 time steps). It is worth noting that our time step duration τ is the same as the measurement interval used when creating those traces, i.e., 5 minutes. The 25 VMs we chose for our simulation have a mean of 2.23, a standard deviation of 0.50 and a variance of 0.25.

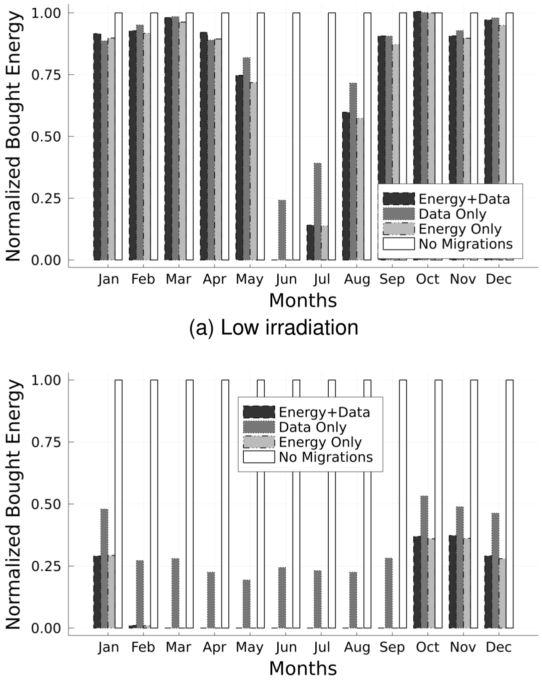
Batteries and PV panels sizing. Using the MILP sizing formulation (Obj2)-(25) presented in Appendix A combined with the CPU and irradiation database described above, we computed the optimal sizing of batteries capacities and amount of PV panels to be installed so that the proposed computational infrastructure would be neutral in terms of non-green energy.

For the system consisting of 5 nodes, each with an idle power consumption of 50 W and maximum power consumption of 500 W, planning horizon of one week (2016 time steps with a 5 minutes granularity) and the average irradiation for Montpellier (Lat. 43.611, Long. 3.876), using all the 624 weeks (2005-2016) of available data, the optimal sizing for the whole system consists of 20 PV panels and a combined battery capacity of 25 kWh.

In the next, we evaluate the resource allocation policies introduced in Section 4.4. The simulated workloads are executed in a best-effort manner: the makespan of the execution is identical for all policies, while the computational load of the computing nodes may vary slightly depending on the VM migrations applied by the optimizer. The energy transfer between the five nodes has an impact on the amount of non-green energy used from the utility grid, when batteries are empty. Thus, we compare the four resource allocation policies based on the amount of non-green energy needed to fulfill the makespan.

5.2 Use case 1: energy-neutral heterogeneous system

We discuss the results obtained with the optimally sized heterogeneous system consisting of 5 nodes, each with an idle power consumption of 50 W and maximum power consumption of 500 W, the PV panels and batteries are distributed as follows: 2 *big* nodes with 7 PV panels and battery capacity of 8 kWh in each one, and 3 *little* nodes with 2 PV panels and battery capacity of 3 kWh in each one.



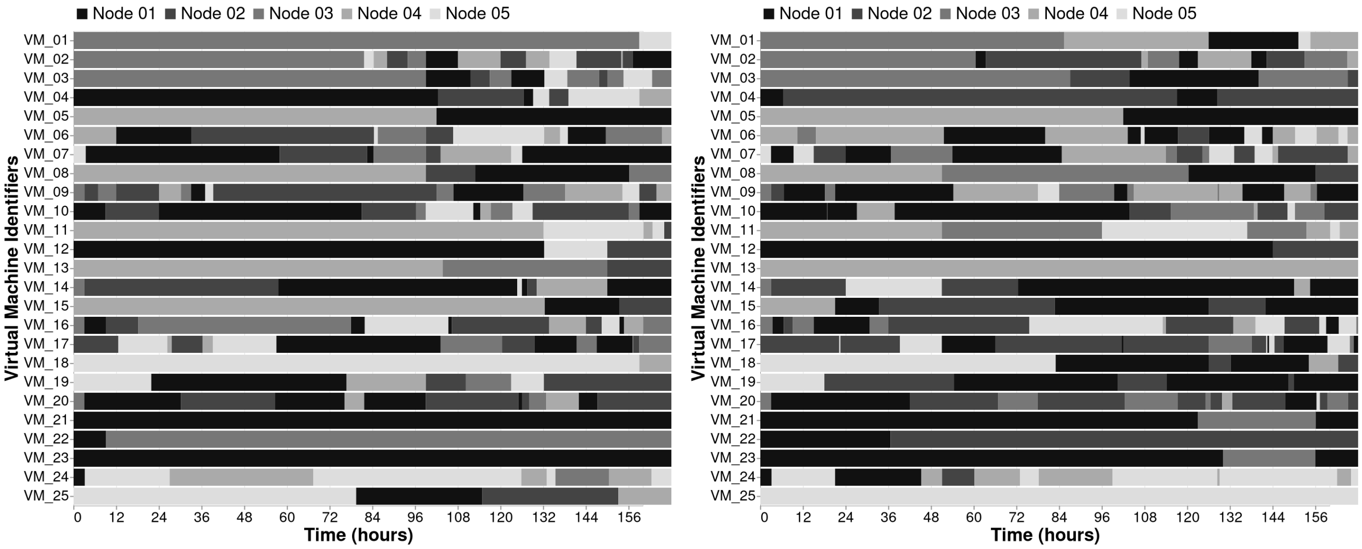
(a) Low irradiation

(b) High irradiation

Fig. 6. Use case 1: normalized results for low and high irradiation periods.

The results for a planning horizon of one week and periods of low and high irradiation over a whole year are presented in the plots depicted in Figures 6a-6b. In both irradiation conditions, the execution policies integrating energy migration provide the best outcome in terms of non-green energy reduction, compared to the policy without any migration. In particular, the high irradiation scenario

[5] https://github.com/beloglazov/planetlab-workload-traces



(a) May

(b) November

Fig. 7. Use case 1: scheduling of VMs to nodes with high irradiation, during two different months.

enables up to 82.5% reduction on average over the year thanks to the larger amount of harvested green energy. In the same scenario, the execution policy relying merely on data migration provides only up to 59.8% reduction on average over the year. This corresponds to 22% less savings than the energy migration based execution policy.

We should point out that the marginal savings observed when using only data migration, for the period of low irradiation in the months of January and April, are due to errors introduced when applying the rolling-horizon heuristic approach. If solved to optimality, the model which employs both energy and data migrations would have at most the same cost as the model using data migration only, as the former is more general than the latter in terms of migration options. Table 4 shows the annual energy bought from and injected into the utility grid by each one of the four tested policies, considering both periods of low and high irradiation.

TABLE 4
Annual average energy bought from and energy injected into utility grid for periods of low and high irradiation.

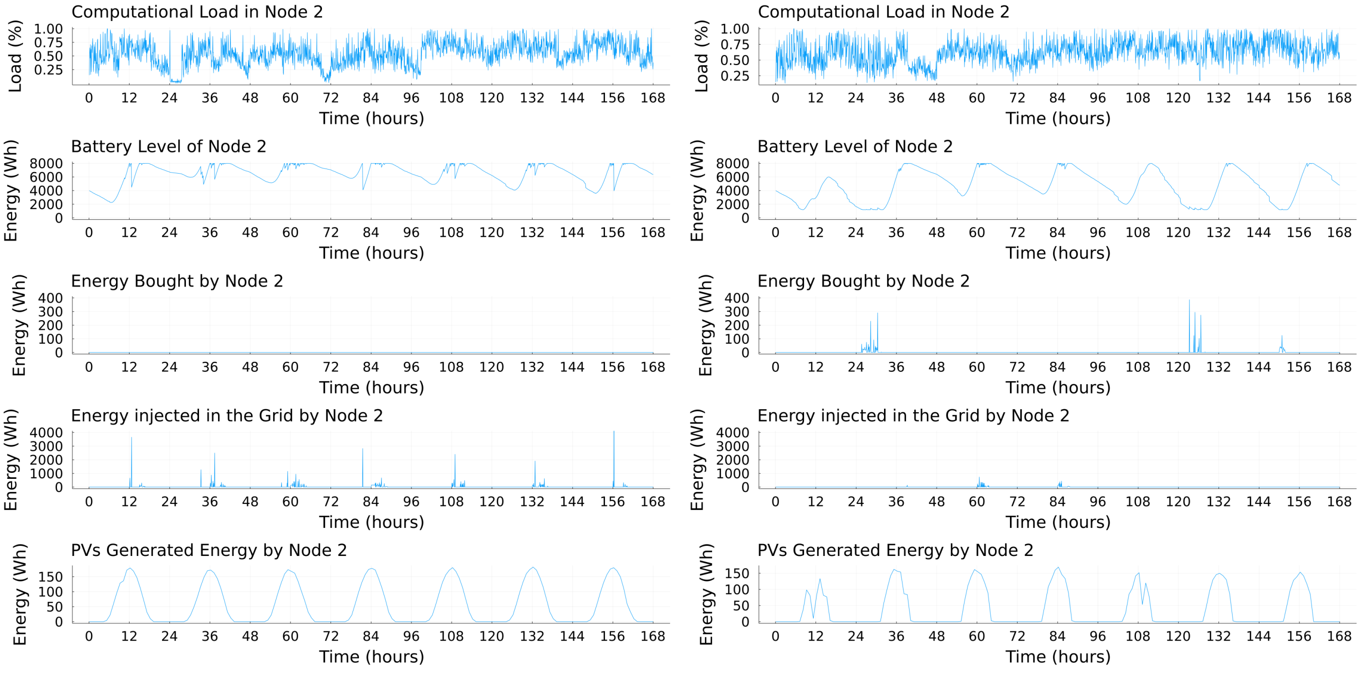
Energy (kWh)	E.+D.	D. Only	E. Only	None
Bought - Low	810.23	838.18	796.02	933.27
Bought - High	56.52	130.03	56.50	323.57
Total	866.75	968.22	852.52	1256.84
Injected - Low	206.90	269.86	206.08	359.60
Injected - High	690.50	830.02	691.09	1031.81
Total	897.40	1099.89	897.17	1391.40
Energy Migrated – Low	259.03	-	178.39	-
Energy Migrated – High	370.07	-	370.91	-
Total	629.10	-	549.30	-
VM Migrations – Low	2018	5570	-	-
VM Migrations – High	2036	5773	-	-
Total	4054	11343	-	-

More generally, when the amount of harvested green energy is lower, the generated scheduling solution exploits VMs migration as much as possible to meet the system execution requirements. For illustration, let us consider again the high irradiation scenario depicted in Figure 6b, where both energy and VMs migrations are enabled.

Figures 7a-7b detail the scheduling of the 25 VMs on the five nodes for the months of May and November, under their best energy harvesting conditions. Note that May and November are two typical months during which the solar irradiation is respectively high and low in Montpellier. As a result, we can observe, through the figures, the system behavior in the presence of potential surplus and deficit of energy production.

Here, each row describes the temporal execution of a VM on the five nodes. For instance, in Figure 7b VM_13 is executed on Node 04 without any migration, while in Figure 7a it is migrated three times during its execution (on Node 04, Node 03 and Node 02).

Globally, we observe that VMs migrations tend to be more frequent in the right-hand half of the execution timeline for both months. This can be explained by the fact that the overall VMs average CPU utilization for the first 84 hours, increases by 26% (from 1.98 to 2.49) when compared to second half of the simulated period. For instance, let us focus on the activity on Node 02, one of the two biggest nodes in terms of energy harvesting and storage capacity. Figures 8a and 8b show the CPU load and energy evolution profiles for this node in the scenario with energy and VMs migrations, and high irradiation period for the months of May and November. We observe an increase in its associated average load after the 84^{th} hour, by 27% and 23%, respectively, in these two months. This is mainly due to the increase in the VMs average CPU utilization, which forces frequent migrations of VMs to avoid CPU over-assignment in some nodes. In addition, to compensate for the extra energy production and storage capacity, CPU intensive VMs may be migrated to Node 02 from the other nodes with less energy storage to successfully achieve VM execution.



(a) May

(b) November

Fig. 8. Use case 1: node 2 load and energy evolution profiles for May and November under high irradiation.

Figure 8a shows that due to the higher and stable irradiation in May there is no need to buy energy from the utility grid. On the other hand, for November, as illustrated in Figure 8b, we obtain a mixed profile in which energy is both bought and injected back into the utility grid.

5.3 Use case 2: accounting for old-generation servers

Most data center operators, such as those mentioned in the introduction section, regularly update the IT hardware, notably for benefiting from higher energy efficiency of last-generation silicon. This indeed results in higher mid-term benefits, i.e. lesser power consumption for similar sold compute service. Nevertheless, this may not be a problem if the ”free” harvested renewable energy enables to sustain the full utilization of these old generation servers.

In the current use case, we therefore explore the outcomes of the previous energy-neutral system dimensioning when considering old generation servers. The rest of the system is kept identical as in the use case 1 (see Section 5.2). However, we degrade the static power consumption of each server by increasing its idle power consumption to 100 W, while keeping its maximum power consumption of 500 W.

The results for a planning horizon of one week and periods of high irradiation over a whole year are presented in Figure 9. We observe that despite the degradation of the static power consumption of the servers, the overall energy-efficiency of the system is almost preserved thanks to the energy migration scheduling policies. The current design enables up to 77.5% reduction of the bought energy on average over the year compared to the no-migration policy. On the other hand, the policy based on data migration reduces this energy by only 49% on average over the year, compared to the no-migration policy.

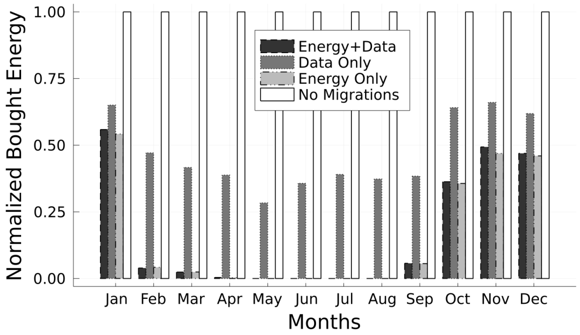


Fig. 9. Use case 2: results for high irradiation period.

5.4 Use case 3: cost-effective heterogeneous system

To devise an energy-neutral system over a whole year, we consider the same sizing of batteries and PV panels as in Section 5.1. In this new use case, we are interested in the cost reduction of the considered energy infrastructure. For this purpose, we explore an alternative system dimensioning by reducing the battery and PV panel components compared to use case 1 (see Section 5.2). Therefore, the total amount of PV panels and battery capacities installed in use case 1 are now reduced by 25%.

These energy resources are now distributed in the following way: 2 *big* nodes with 5 and 4 PV panels, respectively, and battery capacity of 6 kWh in each one, and 3 *little* nodes with 1 PV panel and battery capacity of 1 kWh in each one.

The results for a planning horizon of one week during a period of high irradiation over a whole year is presented in Figure 10. The lighter system dimensioning considered here reduces by 66% the bought energy reduction on average over the year, compared to the no-migration policy. This is only 17% less than the energy reduction obtained with the same policy in use case 1. On the other hand, the policy leveraging data migration only in use case 3 reduces the bought energy by 47% on average over the year, compared to the no-migration policy.

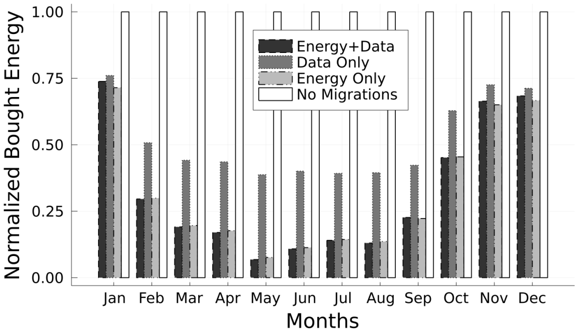


Fig. 10. Use case 3: results for high irradiation period.

6 Conclusion and perspectives

In this paper, we presented an optimization approach for the energy-efficient resource allocation of data centers integrating renewable energy. We promoted a novel distributed system design where both data (or VMs) and energy migrations are permitted. We formulated and solved the resource allocation problem by adopting Mixed Integer Linear Programming combined with a rolling horizon heuristic. We validated our proposal on a representative case study, by analyzing real VMs workload traces and accounting for old generation and less energy-efficient servers. We showed the relevance of our solution for reducing non-green energy consumption and sustaining computing equipment.

In particular, compared to usual resource allocation policies relying on data migration, our solution provides up to 22% reduction of the non-green energy consumption thanks to its energy migration capability. When replacing the servers of the baseline system with old-generation and less energy-efficient servers, this reduction can reach up to 28.5%. This favors the sustainability of the computing equipment at a reasonable exploitation cost in data centers. Further gains could be foreseen with system deployment in geographical areas with higher solar irradiation conditions, such as the Saharan zone. Appendix B reports the evaluation in Mali (West-Africa), of the same system design as for use case 1. Figure 12 shows that even under low irradiation, the reduction of the non-green energy is notable.

Future work will focus on the reduction of the MILP resolution complexity used in our approach. In particular, we plan to extend our resource allocation framework with further heuristics. On the other hand, investigating self-adaptive management approaches such as [7], capable of leveraging energy migration and online prediction of solar irradiation, is a compelling research direction. More generally, the solution presented in the current study at a mini data center level could be extended between multiple mini data centers. Indeed in a realistic urban setup several such mini clusters could be within a limited geographical area, with therefore negligible overheads when exchanging workloads. A straightforward abstraction consists in modeling each entire mini data center as a single green compute node with an aggregated energy profile. Solving could either be handled as per the MILP formulation proposed in this paper or alternatively using heuristics in case the intended number of participating mini data centers is large.

References

[1] N. Jones, "How to stop data centres from gobbling up the world's electricity," *Nature*, vol. 561, no. 7722, pp. 163–166, 2018.

[2] E. Masanet, A. Shehabi, N. Lei, S. Smith, and J. Koomey, "Recalibrating global data center energy-use estimates," *Science*, vol. 367, no. 6481, pp. 984–986, 2020.

[3] E. Sage, "Renewable energy credits," https://www.energysage.com/other-clean-options/renewable-energy-credits-recs/. accessed 23 Dec 2020, 2020.

[4] Google, "Google environmental report," https://www.gstatic.com/gumdrop/sustainability/google-2019-environmental-report.pdf. accessed 27 Feb 2021, 2019.

[5] Facebook, "Facebook sustainability," https://sustainability.fb.com. accessed 27 Feb 2021, 2021.

[6] Amazon, "Sustainable operations: Renewable energy," https://sustainability.aboutamazon.co.uk/environment/sustainable-operations. accessed 27 Feb 2021, 2021.

[7] M. Xu, A. N. Toosi, and R. Buyya, "A self-adaptive approach for managing applications and harnessing renewable energy for sustainable cloud computing," *IEEE Transactions on Sustainable Computing*, pp. 1–1, 2020.

[8] T. Cioara, I. Anghel, M. Antal, S. Crisan, and I. Salomie, "Data center optimization methodology to maximize the usage of locally produced renewable energy," in *2015 Sustainable Internet and ICT for Sustainability (SustainIT)*. Institute of Electrical and Electronics Engineers (IEEE), 2015, pp. 1–8.

[9] J.-M. Pierson, G. Baudic, S. Caux, B. Celik, G. D. Costa, L. Grange, M. Haddad, J. Lecuivre, J.-M. Nicod, L. Philippe, V. Rehn-Sonigo, R. Roche, G. Rostirolla, A. Sayah, P. Stolf, M.-T. Thi, and C. Varnier, "DATAZERO: Datacenter with zero emission and robust management using renewable energy," *IEEE Access*, vol. 7, pp. 103 209–103 230, 2019.

[10] F. Kong and X. Liu, "A survey on green-energy-aware power management for datacenters," *ACM Computing Surveys*, vol. 47, no. 2, pp. 1–38, 2015.

[11] L. Wang, F. Zhang, J. A. Aroca, A. V. Vasilakos, K. Zheng, C. Hou, D. Li, and Z. Liu, "GreenDCN: A general framework for achieving energy efficiency in data center networks," *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 1, pp. 4–15, jan 2014. [Online]. Available: https://doi.org/10.1109\%2Fjsac.2014.140102

[12] Y. Sverdlik, "How is a Mega Data Center Different from a Massive One?" https://www.datacenterknowledge.com/archives/2014/10/15/how-is-a-mega-data-center-different-from-a-massive-one. accessed 20 Sep. 2021, 2014.

[13] K. Bilal, O. Khalid, A. Erbad, and S. U. Khan, "Potentials, trends, and prospects in edge technologies: Fog, cloudlet, mobile edge, and micro data centers," *Computer Networks*, vol. 130, pp. 94–120, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1389128617303778

[14] S. Bird, A. Achuthan, O. Ait Maatallah, W. Hu, K. Janoyan, A. Kwasinski, J. Matthews, D. Mayhew, J. Owen, and P. Marzocca, "Distributed (green) data centers: A new concept for energy, computing, and telecommunications," *Energy for Sustainable Development*, vol. 19, pp. 83–91, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0973082613001129

[15] F. Di Gregorio, G. Sassatelli, A. Gamatié, and A. Castelltort, "A Flexible Power Crossbar-based Architecture for Software-Defined Power Domains," in *22nd European Conference on Power Electronics and Applications (EPE'20 ECCE Europe)*, Lyon, France, Sep. 2020.

[16] F. Di Gregorio, "Exploration of Dynamic Reconfiguration Solutions for Improved Reliability in DC Microgrids," Theses, Université de montpellier 2, Nov. 2021. [Online]. Available: https://hal.archives-ouvertes.fr/tel-03558951

[17] Y. Zhang, Y. Wang, and X. Wang, "Electricity bill capping for cloud-scale data centers that impact the power markets," in *2012 41st International Conference on Parallel Processing*, 2012, pp. 440–449.

[18] The European Commission's science and knowledge service, "Photovoltaic geographical information system (pvgis)," https://ec.europa.eu/jrc/en/pvgis. accessed 27 Feb 2021, 2021.

[19] Z. Abbasi, G. Varsamopoulos, and S. K. S. Gupta, "TACOMA: server and workload management in internet data centers considering cooling-computing power trade-off and energy proportionality," *ACM Transactions on Architecture and Code Optimization*, vol. 9, no. 2, pp. 1–37, 2012.

[20] L. Ganesh, H. Weatherspoon, T. Marian, and K. Birman, "Integrated approach to data center power management," *IEEE Transactions on Computers*, vol. 62, no. 6, pp. 1086–1096, 2013.

[21] Y. Li, Y. Wen, D. Tao, and K. Guan, "Transforming cooling optimization for green data center via deep reinforcement learning," *IEEE Transactions on Cybernetics*, vol. 50, no. 5, pp. 2002–2013, 2020.

[22] H. Lei, R. Wang, T. Zhang, Y. Liu, and Y. Zha, "A multi-objective co-evolutionary algorithm for energy-efficient scheduling on a green data center," *Computers & Operations Research*, vol. 75, pp. 103–117, 2016.

[23] S. Nesmachnow, C. Perfumo, and Í. Goiri, "Holistic multiobjective planning of datacenters powered by renewable energy," *Cluster Computing*, vol. 18, no. 4, pp. 1379–1397, 2015.

[24] P. Ruiu, C. Fiandrino, P. Giaccone, A. Bianco, D. Kliazovich, and P. Bouvry, "On the energy-proportionality of data center networks," *IEEE Transactions on Sustainable Computing*, vol. 2, no. 2, pp. 197–210, 2017.

[25] Í. Goiri, W. Katsak, K. Le, T. D. Nguyen, and R. Bianchini, "Parasol and GreenSwitch: Managing datacenters powered by renewable energy," *ACM SIGPLAN Notices*, vol. 48, no. 4, pp. 51–64, 2013.

[26] ——, "Designing and managing data centers powered by renewable energy," *IEEE Micro*, vol. 34, no. 3, pp. 8–16, 2014.

[27] Í. Goiri, M. E. Haque, K. Le, R. Beauchea, T. D. Nguyen, J. Guitart, J. Torres, and R. Bianchini, "Matching renewable energy supply and demand in green datacenters," *Ad Hoc Networks*, vol. 25, pp. 520–534, 2015, new Research Challenges in Mobile, Opportunistic and Delay-Tolerant Networks Energy-Aware Data Centers: Architecture, Infrastructure, and Communication. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1570870514002649

[28] A. Kassab, J.-M. Nicod, L. Philippe, and V. Rehn-Sonigo, "Scheduling independent tasks in parallel under power constraints," in *2017 46th International Conference on Parallel Processing (ICPP)*. Institute of Electrical and Electronics Engineers (IEEE), 2017, pp. 543–552.

[29] S. Caux, P. Renaud-Goud, G. Rostirolla, and P. Stolf, "IT optimization for datacenters under renewable power constraint," in *24th European Conference on Parallel Processing (Euro-Par 2018)*. Springer International Publishing, 2018, pp. 339–351.

[30] A. A. Chien, R. Wolski, and F. Yang, "The zero-carbon cloud: High-value, dispatchable demand for renewable power generators," *The Electricity Journal*, vol. 28, no. 8, pp. 110–118, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1040619015001931

[31] S. Ismaeel, R. Karim, and A. Miri, "Proactive dynamic virtual-machine consolidation for energy conservation in cloud data centres," *Journal of Cloud Computing*, vol. 7, no. 1, 2018.

[32] I. Hwang and M. Pedram, "Hierarchical virtual machine consolidation in a cloud computing system," in *2013 IEEE Sixth International Conference on Cloud Computing*. Institute of Electrical and Electronics Engineers (IEEE), 2013, pp. 196–203.

[33] F.-H. Tseng, C.-Y. Chen, L.-D. Chou, H.-C. Chao, and J.-W. Niu, "Service-oriented virtual machine placement optimization for green data center," *Mobile Networks and Applications*, vol. 20, no. 5, pp. 556–566, 2015.

[34] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency and Computation: Practice and Experience*, vol. 24, no. 13, pp. 1397–1420, 2011.

[35] S. Jangiti and S. S. VS, "EMC2: Energy-efficient and multi-resource-fairness virtual machine consolidation in cloud data centres," *Sustainable Computing: Informatics and Systems*, vol. 27, p. 100414, 2020.

[36] X. Liu, J. Wu, G. Sha, and S. Liu, "Virtual machine consolidation with minimization of migration thrashing for cloud data centers," *Mathematical Problems in Engineering*, vol. 2020, pp. 1–13, 2020.

[37] I. Hamzaoui, B. Duthil, V. Courboulay, and H. Medromi, "A survey on the current challenges of energy-efficient cloud resources management," *SN Computer Science*, vol. 1, no. 2, 2020.

[38] R. Zolfaghari, A. Sahafi, A. M. Rahmani, and R. Rezaei, "Application of virtual machine consolidation in cloud computing systems," *Sustainable Computing: Informatics and Systems*, vol. 30, p. 100524, 2021.

[39] A. Gamatié, G. Sassatelli, and M. Mikučionis, "Modeling and analysis for energy-driven computing using statistical model-checking," in *Design, Automation and Test in Europe Conference (DATE'21)*, 2021.

[40] E. Gelenbe, "Energy packet networks: Adaptive energy management for the cloud," in *Proceedings of the 2nd International Workshop on Cloud Computing Platforms*, ser. CloudCP '12. New York, NY, USA: Association for Computing Machinery, 2012. [Online]. Available: https://doi.org/10.1145/2168697.2168698

[41] A. Schrijver, *Theory of linear and integer programming*, ser. Wiley-Interscience series in discrete mathematics and optimization. Wiley, 1998.

[42] G. Sierksma and Y. Zwols, *Linear and integer optimization : theory and practice*, 3rd ed., ser. Advances in applied mathematics. Chapman & Hall/CRC, 2015.

[43] X. Fan, W.-D. Weber, and L. A. Barroso, "Power provisioning for a warehouse-sized computer," *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2, pp. 13–23, 2007.

[44] Q. Huang, F. Gao, R. Wang, and Z. Qi, "Power consumption of virtual machine live migration in clouds," in *2011 Third International Conference on Communications and Mobile Computing*. Institute of Electrical and Electronics Engineers (IEEE), 2011, pp. 122–125.

[45] 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, 2011.

[46] A. Strunk, "A lightweight model for estimating energy cost of live migration of virtual machines," in *2013 IEEE Sixth International Conference on Cloud Computing*. Institute of Electrical and Electronics Engineers (IEEE), 2013, pp. 510–517.

[47] V. D. Maio, R. Prodan, S. Benedict, and G. Kecskemeti, "Modelling energy consumption of network transfers and virtual machine migration," *Future Generation Computer Systems*, vol. 56, pp. 388–406, 2016.

[48] J. G. Rakke, M. Stålhane, C. R. Moe, M. Christiansen, H. Andersson, K. Fagerholt, and I. Norstad, "A rolling horizon heuristic for creating a liquefied natural gas annual delivery program," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 5, pp. 896–911, 2011.

[49] A. Bischi, L. Taccari, E. Martelli, E. Amaldi, G. Manzolini, P. Silva, S. Campanari, and E. Macchi, "A rolling-horizon optimization algorithm for the long term operational scheduling of cogeneration systems," *Energy*, vol. 184, pp. 73–90, 2019.

[50] I. Dunning, J. Huchette, and M. Lubin, "Jump: A modeling language for mathematical optimization," *SIAM Review*, vol. 59, no. 2, pp. 295–320, 2017.

[51] K. Park and V. S. Pai, "CoMon: a mostly-scalable monitoring system for PlanetLab," *ACM SIGOPS Operating Systems Review*, vol. 40, no. 1, pp. 65–74, 2006.

Acknowledgments

This work is supported by the IWARE project, funded by Région Occitanie, France.

Dr. Marcos De Melo da Silva is a research engineer at CNRS, currently working as an operations research specialist with the Adaptive Computing team at LIRMM laboratory, University of Montpellier, France. He is a computer scientist with specialization in combinatorial optimization and operations research. His research focuses on the design, analysis and development of efficient, exact and approximate, algorithms for combinatorial problems in the domains of city logistics, transportation and scheduling.

Dr. Abdoulaye Gamatié is currently a Senior CNRS Researcher at LIRMM, a joint laboratory of Univ. Montpellier and CNRS, France. His research activity focuses on the design of energy-efficient multicore and multiprocessor systems for embedded and high-performance computing domains. He has coauthored more than 90 articles in peer-reviewed journals and international conferences. He has been involved in several collaborative international projects with both academic and industrial partners. He has served on the editorial boards of scientific journals including IEEE TCAD and ACM TECS.

Dr. Gilles Sassatelli is a Senior CNRS Scientist at LIRMM, a CNRS-University of Montpellier joint research unit. He conducts research in the area of adaptive energy-efficient systems in the adaptive computing group. He is the author of more than 200 publications in a number of renowned international journals and international conferences. He regularly serves as Track or Topic Chair in major conferences in the field of embedded systems (DATE, ReCoSoC, ISVLSI etc.). Most of his research is conducted in collaboration with international partners; over the past five years he has been involved in several national and European research projects including DREAMCLOUD and MONT-BLANC projects (FP7 and H2020).

Dr. Michael Poss is a senior research fellow at the LIRMM laboratory that depends on both the University of Montpellier and the National Center for Scientific Research (CNRS). His current research focuses mainly on robust combinatorial optimization. He has been involved in several collaborative projects, and has served as PI for some of them.

Prof. Michel Robert (PhD'1987) is Professor at the University of Montpellier (France), where he is teaching microelectronics in the engineering program. His present research interests at the Montpellier Laboratory of Informatics, Robotics, and Micro-electronics (LIRMM) are design and modelisation of system on chip architectures. He is author or co-author of more than 250 publications in the field of CMOS integrated circuits design.

Appendix A
Battery and PV panels sizing model

In the MILP formulation (Obj1), (2)-(9) presented in Section 4.2, the number of photovoltaic panels used for computing the amount of solar energy that is injected into the nodes and the capacity of the batteries installed in each node are input parameters that need to be informed by the user. We will now describe an MILP formulation that can be applied to compute such parameters.

The proposed sizing model can be seen as an extension of the scheduling MILP model (Obj1), (2)-(9). Therefore, in addition to the decision variables (f, q, v, x, w, and z), we also need the following variables:

- $u_i \geq 0$: battery capacity to be used in node i.
- $g_i \geq 0$: amount of PV panels to be used in node i.

When sizing the batteries we ensure that they cannot be discharged below a safety level $\sigma = 0.15$, i.e., 15%.

The sizing formulation objective function is:

$$\begin{aligned}\text{Minimize } & \sum_{i\in N} g_i + \sum_{i\in N} u_i + \lambda \sum_{t\in H} \sum_{i,k\in N} f_{ik}^t \\ & + \mu \sum_{t\in H} \sum_{i,k\in N} \sum_{j\in J} z_{ikj}^t + \phi \sum_{t\in H} \sum_{i\in N} q_i^t \\ & + \nu \sum_{t\in H} \sum_{i\in N} v_i^t \end{aligned} \tag{Obj2}$$

and is subject to the following constraints:

$$\begin{aligned} w_i^t = {} & w_i^{t-1} + G_i^t g_i + \sum_{k\in N} E_{ki} f_{ki}^t \\ & - \sum_{k\neq i\in N} f_{ik}^t - \mu_s \sum_{k\in N} \sum_{j\in J} z_{ikj}^t - \mu_d \sum_{k\in N} \sum_{j\in J} z_{kij}^t \\ & - (\varepsilon_I + \varepsilon_P \sum_{j\in J} C_j^t x_{ij}^t) \qquad \forall t\in H,\ i\in N \end{aligned} \tag{15}$$

$$w_i^0 \geq \sigma u_i \qquad \forall i \in N \tag{16}$$

$$w_i^t \geq \sigma u_i \qquad \forall t\in H,\ i\in N \tag{17}$$

$$w_i^t \leq u_i \qquad \forall t\in H,\ i\in N \tag{18}$$

$$w_i^T \geq w_i^0 - \varphi \sum_{t\in H} q_i^t \qquad \forall i\in N \tag{19}$$

$$\sum_{j\in J} C_j^t x_{ij}^t \leq R_i^C + v_i^t \qquad \forall t\in H,\ i\in N \tag{20}$$

$$\sum_{i\in N} x_{ij}^t = 1 \qquad \forall t\in H,\ j\in J \tag{21}$$

$$z_{ikj}^t \geq x_{ij}^{t-1} + x_{kj}^t - 1 \qquad \forall t\geq 2,\ j\in J,\ i\neq k\in N \tag{22}$$

$$x_{ij}^t,\ z_{ikj}^t \in \{0,1\} \qquad \forall t\in H,\ j\in J,\ i\neq k\in N \tag{23}$$

$$g_i,\ u_i \geq 0 \qquad \forall i\in N \tag{24}$$

$$b_i^t,\ q_i^t,\ v_i^t,\ w_i^t,\ f_{ik}^t \geq 0 \qquad \forall t\in H,\ i\neq k\in N \tag{25}$$

The objective function (Obj2) seeks to minimize the sum of battery capacities, the number of installed solar panels, and similar to the formulation (Obj1), (2)-(9), minimizes: i) the energy losses incurred when performing energy or VM migrations between nodes, ii) the energy losses associated with the surplus energy generated that is injected into the utility grid, and iii) the penalties for over-utilization of processing resources, respectively.

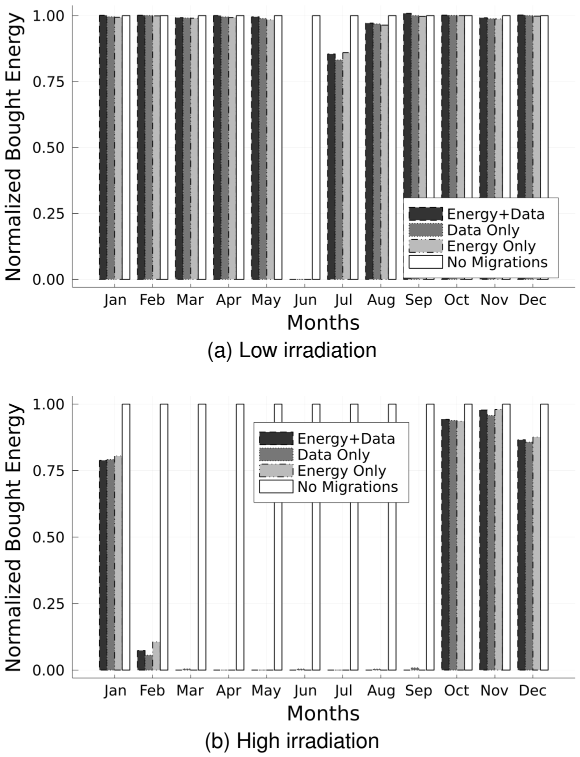
Constraint (15) defines how the state of the batteries is updated each time step $t \in H$. The batteries initial charge, safe discharge levels, maximum capacity and remaining charge levels are enforced by constraints (16)-(19), respectively. The constraints related to CPU resource allocation (20), and the scheduling of the VMs to nodes (21) and (22) are the same as in model (Obj1), (2)-(9). Finally, constraints (23)-(25) define the domains of the variables.

Appendix B
Alternative use case evaluations

In the sequel, we briefly illustrate two additional evaluations of our proposal, under different setups: an homogeneous system resource dimensioning and the deployment of the system on a different geographical zone.

B.1 Homogeneous system under energy-neutrality

We discuss the results obtained with an optimally sized homogeneous system consisting of 5 green nodes, each with an idle power consumption of 50 W and maximum power consumption of 500 W. The PV panels and batteries are equally distributed among the 5 green nodes: 4 PV panels and battery capacity of 5 kWh per node. The results for a planning horizon of one week and periods of low and high irradiation over a whole year are presented in the plots depicted in Figures 11a-11b.



(a) Low irradiation

(b) High irradiation

Fig. 11. Homogeneous system design under energy neutrality in the South of France.

Given the identical resource dimensioning across the different green nodes, the energy and computing demand is also identical over the time. Therefore, neither VM nor energy migration is helpful here. As a consequence, the four execution scenarios become equivalent in terms of non-green energy reduction.

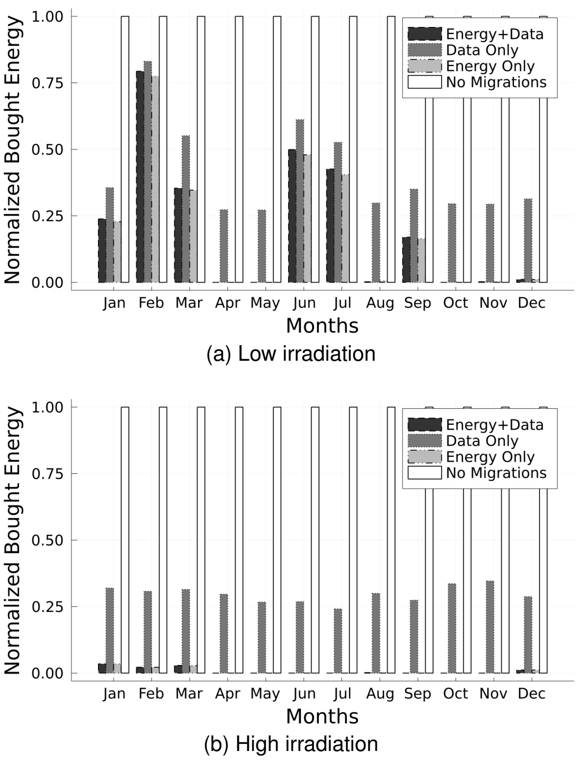
This homogeneous system design shows that both data and energy migrations are mainly relevant in the situations where the resource availability evolves differently in the considered nodes, over the time. Therefore, VMs and energy migrations can help in re-equilibrating the resource utilization.

B.2 Use case 1-bis: evaluation for Mali (West-Africa)

We analyse the behaviour of the proposed system for Mali, a Saharan country in West-Africa, where we expect higher levels of solar irradiation during the whole year provided that this country is closer to the equator.

We consider the same heterogeneous system and resources sizing of the use case 1 (see Section 5.2). The results for a planning horizon of one week and periods of low and high irradiation over a whole year for Mali are presented in the plots depicted in Figures 12a-12b.

They show that for the regions of the World with a very favorable solar irradiation condition, the overall gains in terms of non-green energy reduction are very significant over a year.



(a) Low irradiation

(b) High irradiation

Fig. 12. Use case 1 normalized results for low and high irradiation periods in Mali (West-Africa).