Energy-saving self-configuring networked data centers

Nicola Cordeschi *, Mohammad Shojafar, Enzo Baccarelli

Dpt. of Information Engineering, Electronic and Telecommunication, “Sapienza” University of Rome, via Eudossiana 18, 00184 Rome, Italy

Abstract

In this paper, we develop the optimal minimum-energy scheduler for the dynamic online joint allocation of the task sizes, computing rates, communication rates and communication powers in virtualized Networked Data Centers (NetDCs) that operates under hard per-job delay-constraints. The referred NetDC’s infrastructure is composed by multiple frequency-scalable Virtual Machines (VMs), that are interconnected by a bandwidth and power-limited switched Local Area Network (LAN). Due to the nonlinear power-vs.-communication rate relationship, the resulting Computing-Communication Optimization Problem (CCOP) is inherently nonconvex. In order to analytically compute the exact solution of the CCOP, we develop a solving approach that relies on the following two main steps: (i) we prove that the CCOP retains a loosely coupled structure, that allows us to perform the lossless decomposition of the CCOP into the cascade of two simpler sub-problems; and, (ii) we prove that the coupling between the aforementioned sub-problems is provided by a (scalar) constraint, that is linear in the offered workload. The resulting optimal scheduler is amenable of scalable and distributed online implementation and its analytical characterization is in closed-form. After numerically testing its actual performance under randomly time-varying synthetically generated and real-world measured workload traces, we compare the obtained performance with the corresponding ones of some state-of-the-art static and sequential schedulers.

© 2013 Published by Elsevier B.V.

1. Introduction

Green computing over Networked Data Centers (NetDCs) is an emerging paradigm that aims at performing the dynamic energy-saving management of data center infrastructures. The goal is to provide QoS Internet services for large populations of clients, while minimizing the overall computing-plus-communication energy consumption [1–3, Chapter 6]. As recently pointed out in [3], the energy cost of communication gear for data centers may represent a large fraction of the overall system cost and it is induced primarily by switches, LAN infrastructures, routers and load balancers. In order to attain energy-saving, virtualization techniques are typically used for attaining resource isolation and online resource balancing [3]. However, current virtualized data centers are not designed for supporting communication–computing intensive real-time applications, such as, info-mobility applications, real-time video co-decoding, target recognition and tracking [28,33]. In fact, imposing hard-limits on the overall per-job delay does not allow the data center to perform data buffering over large time intervals, but forces the overall networked computing infrastructure to adapt quickly its resource utilization to the (possibly, unpredictable and abrupt) time fluctuations of the offered workload [4,9,10].

1.1. Goal of this paper

In order to attain the green paradigm, the joint balanced provisioning and scaling of the communication-plus-computing resources are demanded. This is, indeed, the
focus of this paper, whose main contributions can be summarized as follows:

i. the contrasting objectives of low consumption of both communication and computing energies in delay and bandwidth-constrained NetDCs affected by reconfiguration costs are cast in the form of a suitable constrained optimization problem, namely, the Computing and Communication Optimization Problem (CCOP);

ii. due to the nonlinear behavior of the rate-vs.-power-vs.-delay relationship, the CCOP is not a convex optimization problem and neither guaranteed-convergence iterative algorithms nor closed-form formulas are, to date, available for its solution. Hence, in order to solve the CCOP in exact and closed-form, we prove that it admits a loss-free (e.g., optimality preserving) decomposition into two simpler sub-problems, namely, the Communication Optimization Problem (CMOP) and the Computing Optimization Problem (CPOP). Although the CMOP and CPOP are not guaranteed to be convex problems, their loosely coupled structure (in the sense of [25]) allows us to solve the (nonconvex) CCOP and develop analytical conditions for its feasibility;

iii. we develop a fully autonomic version of the proposed resource scheduler, that is capable to quickly adapt to the a priori unknown time-variations of the offered workload, without requiring workload forecasting;

iv. finally, we derive analytical conditions for the (possible) hibernation of the instantiated VMs, that highlight the tight inter-play between computing and communication resources.

A remarkable feature of the developed adaptive scheduler is its scalable and distributed structure, that makes the complexity of its online implementation independent from the size of the considered NetDC.

1.2. Related work

Updated surveys of the current technologies and open communication challenges related to the green cloud paradigm have been recently presented in [3,15]. Specifically, power management schemes that exploit Dynamic Voltage and Frequency Scaling (DVFS) techniques [15, Section 5] for performing resource provisioning are the focus of [26,30,31]. Although these contributions consider hard-deadline constraints, they do not consider, indeed, the performance penalty and the energy-vs.-delay tradeoff stemming from the finite capacity of the utilized LANs. Furthermore, the complexity of the online implementation of the therein proposed schedulers scales as $O(M \log(M))$, where $M$ is the minimum between the number of tasks to be executed in parallel and the number of available computing machines.

Energy-saving dynamic provisioning of the computing resources in virtualized green data centers is the topic of [10,16–20,27]. Specifically, [16] formulates the optimization problem as a feedback control problem that must converge to a priori known target performance level. While this approach is suitable for tracking problems, it cannot be employed for energy-minimization problems, where the target values are priori unknown. Roughly speaking, the common approach pursued by [10,17,18,20] is to formulate the afforded minimum-cost problems as sequential optimization problems and, then, solves them by using limited look-ahead control. Hence, the effectiveness of this approach relies on the ability to predict accurately future workload and degrades when the workload exhibits almost unpredictable time fluctuations [15].

Furthermore, the joint provisioning of communication and computing resources is not considered by these contributions, that mainly focus on the computing aspects. In order to avoid the prediction of future workload, [19] resorts to a Lyapunov-based technique, that dynamically optimizes the provisioning of the computing resources by exploiting the available queue information. Although the pursued approach is of interest, it relies on an inherent delay-vs.-utility tradeoff, that does not allow to account for hard-deadline constraints. The combined exploitation of DVFS and virtualization techniques is the focus of [27]. Although the parallel computing platform considered in [27] is, indeed, multi-core and managed by a Virtual Machine Manager (VMM), the framework developed in [27] does not consider load balancing and neglects both the energy and time overheads induced by the underlying LAN.

The joint analysis of the computing-plus-communication energy consumption in NetDCs is, indeed, the focus of [2,4,6], where delay-tolerant Internet-based applications are considered. Interestingly, the main lesson stemming from these contributions is that the energy consumption due to data communication may represent a large part of the overall energy demand, especially when the utilized network is power and/or bandwidth-limited. Overall, these works numerically analyze and test the energy performance of some state-of-the-art schedulers for NetDCs, but do not attempt to optimize it through the dynamic joint scaling of the available communication-plus-computing resources. As also recently pointed out in [3] and [15, Section 9], this is still an open research topic, especially when hard-delay requirements are also present.

The rest of this paper is organized as follows. After modeling in Section 2 the considered NetDC infrastructure, in Section 3 we formally state the afforded CCOP and, then, in Section 4, we solve it and provide the analytical conditions for its feasibility. In Section 5, we present the main structural properties of the resulting optimal scheduler, and analytically characterize the (possible) occurrence of hibernation phenomena of the instantiated VMs. In Section 6, we numerically test the average performance of the proposed scheduler at various values of the Peak-to-Mean Ratio (PMR) of the (randomly time-variant) offered workload, and, then, we compare the obtained performance against the corresponding ones of some state-of-the-art static and sequential schedulers. The conclusive Section 7 recaps the main results and outlines some hints for future research. The final Appendix reports the main analytical proofs.
About the adopted notation, \(\chi_{A}^{b}\) indicates \(\min\{\max(x; a) ; b\}\), \(|x|\), means \(\max(x; 0)\), while \(1_{[A]}\) is the (binary-valued) indicator function of the \(A\) event (that is, \(1_{[A]}\) is unit when the \(A\) event happens, while it vanishes when the \(A\) event does not occur).

2. The considered NetDC infrastructure

In principle, a networked clustered platform for parallel computing is composed by multiple (possibly, virtualized) processing units and a central resource controller [1]. Each processing unit executes the currently assigned task as an independent processor by self-managing own local storage/computing resources. Intra-cluster communication is supported through message passing. When a request for a new job is submitted to the NetDC, the central resource controller dynamically performs both admission control and resource allocation. [14,15].

Hence, from an infrastructure perspective, emerging NetDCs are composed by three main components [14,15], e.g., the data storage, the VMM and the switched LAN (see Fig.1). A new job is initiated by an event, that is constituted by the arrival at the instant \(t_{a}\) of a file of size \(L_{b}\) (bit). Due to the real-time nature of the considered application scenario, full processing of the input file must be carried out within a given (e.g., a priori assigned) deterministic working time \(T_{b}\) (s). Hence, in our framework, a real-time job is characterized by [13]: (i) the size \(L_{b}\) (bit) of the file to be processed; (ii) the maximum tolerated processing delay \(T_{b}\) (s); and, (iii) the job granularity, that is, the (integer-valued) maximum number \(M_{V}\) of independent parallel tasks embedded into the submitted job [13, Section 2.4].

Let \(M_{V} \geq 1\) be the (integer-valued) maximum number of VMs that may be instantiated onto the NetDC of Fig. 1. In principle, each VM may be modeled as a (virtual) server, that is capable to process \(f_{c}\) bits per second [24]. Depending on the size \(L_{b}\) (bit) of the task \(T\) to be currently processed by the VM, the processing rate \(f_{c}\) may be adaptively scaled at run-time, and it may assume values over the interval \([0, f_{c}^{\max}]\), where \(f_{c}^{\max}\) (bit/s) is the maximum allowed processing rate.

Furthermore, due to the real-time nature of the considered application scenario, the time allowed the VM to fully process each submitted task is fixed in advance at \(A(s)\), regardless of the actual size \(L_{b}\) of the task currently submitted to the VM. In addition to the currently submitted task (of size \(L_{b}\)), the VM may also process a background workload of size \(L_{b}\) (bit), that accounts for the Operating System (OS) programs. As in [13,24], this background workload is assumed stored by the local memory equipping the VM, so that the execution of the background task entails computing costs but does not induce communication costs.

Hence, by definition, the utilization factor \(\eta\) of the VM equates [24]: \(\eta \triangleq f_{c}/f_{c}^{\max} \in [0, 1]\). The analysis reported in the (quite recent) contributions [7, Section 3.4], [14,23, Section 3.1] points out that the reduction of the dynamic component of the computing energy plays a major role in the reduction of the overall computing energy, especially when the supported services are of real-time type. Then, in agreement with these contributions, let \(E_{c} \equiv E_{c}(f_{c})\) (Joule) be the overall energy consumed by the VM to process a single task of duration \(A\) (s) at the processing rate \(f_{c}\), and let \(E_{c}^{\max} \equiv E_{c}(f_{c}^{\max})\) (Joule) be the corresponding maximum energy when the VM operates at the maximum processing rate \(f_{c}^{\max}\). Then, by definition, the dimensionless ratio:

\[
\Phi(\eta) \triangleq \frac{E_{c}(f_{c})}{E_{c}^{\max}} \equiv \frac{f_{c}}{f_{c}^{\max}},
\]

is the so-called Normalized Energy Consumption of the considered VM [5]. From an analytical point of view, \(\Phi(\eta) : [0, 1] \rightarrow [0, 1]\) is a function of the actual value \(\eta\) of the VM utilization factor. Its analytical behavior depends on the specific features of the resource provisioning policy.

**Fig. 1.** The considered NetDC architecture. Continuous lines (–) indicate bidirectional data flows. Dotted lines (—) are bidirectional controlling paths. The LAN is composed of the switch unit and the associated point-to-point links.
actually implemented by the VMM of Fig. 1 [7,11]. However, at least for CMOS-based physical CPUs, the following three (mild) assumptions on $\Phi(\eta)$ are typically met [5,23]: (i) $\Phi(\eta = 0) = 0$ and $\Phi(\eta = 1) = 1$; (ii) $\Phi(\eta)$ is strictly increasing and continuous in $\eta$; and, (iii) $\Phi(\eta)$ is strictly convex in $\eta$, e.g., $d^2\Phi(\eta)/d\eta^2 > 0, \forall \eta \in [0, 1]$. Just as a practical example, the analytical form assumed by $\Phi(\eta)$ for DVFS-enabled CMOS CPUs is recognized to be well described by the following quadratic one [5,11,23]:

$$\Phi(\eta) = \eta^2, \eta \in [0, 1].$$  \hspace{1cm} (2)

Furthermore, as in [13], we may utilize the (dimensionless) attribute $\omega$ for measuring the relative energy cost incurred by the actuation of the considered VM for the execution of the planned task. Specifically, larger values of $\omega$ make more energy expensive the execution of the task on the considered VM, and $\omega = \infty$ forbids at all the execution of the planned task by VM. Therefore, a suitable setting of the attributes $\omega(i), i = 1, \ldots, M$, may capture task preferences, heterogeneity in the functionality of the available VMs, as well as underlying task-placement constraints [13].

2.1. The offered workload

Let $M \triangleq \min\{M_0, M_1\}$ be the concurrency degree of the submitted workload (that is, the number of not overlapping tasks that can be executed in parallel for carrying out the job; [13, Section 2.4]), and let $L_i$ (bit) be the size of the task currently submitted to the VM (i). Hence, since the per-task processing time allowed the VM (i) does not depend on the task size $L_i$ and equates $A$, the resulting processing rate $f_i(t)$ (bit/s) equates $f_i(t) = L_i/A$ (bits/s). This means, in turn, that the maximum size $L_{i}^{\text{max}}$ (bit) allowed a single task is: $L_{i}^{\text{max}} = A f_i^{\text{max}}(t)$ (bit), so that: $\eta \triangleq \frac{L_i}{L_{i}^{\text{max}}} = \frac{1}{\eta}$. Afterwards, by referring to Fig. 1, let $L_i$ (bit) the overall size of the job currently submitted to the NetDC, and let $L_i \geq 0, i = 1, \ldots, M$, be the size of the task that the scheduler of Fig. 1 assigns to the VM (i). Hence, the following constraint: $\sum_{i=1}^{M} L_i = L$, guarantees that the overall job $L$ is partitioned into (at the most) $M$ parallel tasks. We anticipate that, in Section 6, we explicitly take account for the random nature (possibly) retained by the size $L_i$ of the per-job offered workload and we consider dynamic stochastic settings where $L_i$ exhibits (possibly, abrupt and unpredictable) time-fluctuations.

2.2. The bandwidth and power-limited communication infrastructure

In order to keep to a minimum the transmission delays from (to) the scheduler to (from) the connected VMs of Fig. 1, as in [2,14], we assume that each VM (i) communicates with the scheduler through a dedicated (i.e., contention-free) reliable link, that operates at the transmission rate of $R_i$ (bit/s), $i = 1, \ldots, M$ (see Fig. 1). The one-way transmission-plus-switching operation over the ith link drains a (variable) power of $P_{i}^{\text{net}}$ (Watt), where $P_{i}^{\text{net}}$ is the summation: $P_{i}^{\text{net}} \triangleq P_{i}^{\text{net}}(i) + P_{i}^{\text{sw}}(i)$ of the power $P_{i}^{\text{sw}}(i)$ required by the (one-way) transmission and switching and the corresponding power $P_{i}^{\text{net}}(i)$ demanded by the receive circuit.

In general, the actual value assumed by $P_{i}^{\text{net}}$ depends on the corresponding transmission rate $R_i$, noise spectral power density $N_0(i) (W)/(Hz)$, bandwidth $W(i) (Hz)$ and (non-negative) gain $g_i$ of the ith link [8]. The Shannon-Hartley exponential formula:

$$P_{i}^{\text{net}} = P_{i}^{\text{net}}(R_i) = \zeta_i \left( \frac{R_i}{W(i)} - 1 \right),$$  \hspace{1cm} (3)

with $\zeta_i \triangleq \frac{N_0(i) W(i)}{g_i}$, $i = 1, \ldots, M$, and the $x$-powered formula:

$$P_{i}^{\text{net}} = P_{i}^{\text{net}}(R_i) = \Omega_i \left( \frac{R_i}{W(i)} \right)^{1/x},$$  \hspace{1cm} (4)

with $x \in [0, 1]$, and $\Omega_i \triangleq \frac{N_0(i) W(i)}{(\ln(2)^2 g_i)}$. $i = 1, \ldots, M$, are examples of power-rate functions of practical interest [8, Chapter 3]. Hence, the corresponding one-way transmission delay $D(i)$ (s) equates: $D(i) = L/R_i$, so that the corresponding one-way communication energy $E_{i}^{\text{net}}(i)$ (Joule) is: $E_{i}^{\text{net}}(i) = P_{i}^{\text{net}}(R_i)(L_i/R_i)$.

2.3. Reconfiguration cost and machine management

The Manager Module of Fig. 1 must carry out two main operations at run-time, namely, the Virtual Machine management and the load balancing. Specifically, goal of the Virtual Machine management is to dynamically control the Virtualization Layer of Fig. 1, in order to attain an optimal mapping of the available physical resources onto multiple (possibly, heterogeneous) VMs. In particular, the (aforementioned) set of VMs’ attributes:

$$\{A, f_i^{\text{net}}(i), \Phi(\eta), \omega(i), L_i \}, i = 1, \ldots, M \} ,$$

are dictated by the Virtualization Layer and, then, they are passed to the VMM of Fig. 1. It is in charge of the VMM to implement a suitable frequency-scaling policy, so as to allow the VMs to scale up/down in real-time their processing rates $f_i$’s at the minimum cost [1,5]. At this regard, we note that switching from the processing frequency $f_i$ to the processing frequency $f_j$ entails an energy cost of $\varepsilon(f_i; f_j)$ (Joule) [27,30]. Although the actual behavior of the function $\varepsilon(f; f_j) = \varepsilon(f_1; f_2)$ may depends on the adopted DVFS technique and the underlying physical CPUs [24], any practical $\varepsilon(f; f_j)$ function typically retains the following (mild) properties [27,30]: (i) $\varepsilon(f_1; f_j)$ depends on the absolute frequency gap $|f_j - f_i|$; (ii) it vanishes at $f_1 = f_2$ and is not decreasing in $|f_1 - f_i|$; and, (iii) it is jointly convex in $f_1, f_2$. A quite common practical model that retains the aforementioned formal properties is the following one:

$$\varepsilon(f_1; f_2) = k_e(f_1 - f_2)^2 \text{(Joule)},$$

where $k_e \text{(Joule)/(Hz)}^2$ dictates the reconfiguration cost induced by an unit-size frequency switching. Typical values of $k_e$ for current DVFS-based virtualized computing platforms are limited up to few hundreds of $\mu$J’s per $(MHz)^2$ [30]. For the sake of concreteness, in the analytical developments of the following Section 3, we directly resort to the quadratic model in (6). The generalization to the case of $\varepsilon(\cdot, \cdot)^2$ functions that meet the aforementioned analytical properties is, indeed, direct.
Remark 1. Parallel tasks in clustered NetDCs

We point out that, in our framework, the size $L_i$ of each submitted job remains constant over the corresponding working time $T_i$ and no workload fluctuations take place during the execution of each job. Hence, formally speaking, neither migrations of VMs nor actions for turning the underlying physical CPUs ON/OFF are to be forecast at runtime, so that the scheduling policies considered here are of clairvoyant-type [13]. Finally, our model subserves that the job inter-arrival are delays larger than the allowed per-job processing time $T_i$, so that no queuing phenomena occur. As detailed in the following Section 3, load balancing and task parallelization are effective means to meet this assumption. This is in agreement with the assumed hard real-time behavior of the overall NetDC of Fig. 1, that, in principle, cannot be guaranteed when randomly time-variant unpredictable queuing/blocking delays occur [30].

Remark 2. Discrete DVFS

In general, the aforementioned assumption of continuous-valued utilization factor $\eta$ may be questionable, because, due to the one-to-one mapping in (1), it is equivalent to require continuous-valued computing rates $f_i$’s. Actual VMs are instantiated on top of physical CPUs, that offer, indeed, only a finite set:

$$A = \{ 0, f_1, \ldots, f_{Q-1}, f_{Q} = f_{\text{max}} \},$$

(7)

of $Q$ discrete computing rates [28]. Hence, in order to deal with both continuous and discrete DVFS-enabled computing platforms without introducing loss of optimality, we borrow the approach formerly developed, for example, in [21,31]. Specifically, after indicating as

$$B = \{ 0, \eta_1, \ldots, \eta_{Q-1}, 1 \},$$

(8)

discrete values of $\eta$ that correspond to the frequency set $A$ in (7), we build up a Virtual Energy Consumption curve $\Phi(\eta)$ by resorting to a piecewise linear interpolation of the allowed operating points: $\{(\eta_j, \Phi(\eta_j))\}, j = 0, \ldots, Q - 1$. Obviously, such virtual curve retains the (aforementioned) continuity property and, then, we may use it as the true energy consumption curve for resource provisioning [21]. Unfortunately, being the virtual curve of continuous type, it is no longer guaranteed that the resulting optimally scheduled computing rates are still discrete valued. However, as also explicitly noted in [21,31], any point $(\eta^j, \Phi(\eta^j))$, with $\eta^j < \eta^j < \eta^{j+1}$, on the virtual curve may be actually attained by time-averaging over $A$ secs (i.e., on a per-job basis) the corresponding surrounding vertex points: $(\eta^j, \Phi(\eta^j))$ and $(\eta^{j+1}, \Phi(\eta^{j+1}))$. Due to the piecewise linear behavior of the virtual curve, as in [21,31], it is guaranteed that the average energy cost of the discrete DVFS system equates that of the corresponding virtual one over each time interval of duration $A$ (s) (e.g., on a per-job basis). Furthermore, from a technological point of view, the time-overheads induced by frequency scaling are limited up to few tens of $\mu$s in state-of-the-art DVFS-enabled multi-core computing platforms [15,28], so that current technology makes feasible the execution of time-sharing operations at run-time [31]. Hence, in the sequel, we directly focus on the case of continuous (possibly, piecewise linear) DVFS-enabled computing platforms.

3. Optimal communication–computing resource allocation

In this section, we deal with the second service offered by the Manager Module of Fig. 1, namely, the dynamic load balancing and provisioning of the communication-plus-computing resources. Specifically, this service aims at properly tuning the task sizes $\{L_i, i = 1, \ldots, M\}$, the communication rates $\{R_i, i = 1, \ldots, M\}$ and the computing rates $\{f_i, i = 1, \ldots, M\}$ of the DVFS-enabled VMs of Fig. 1. The goal is to minimize (on a per-slot basis) the overall resulting communication-plus-computing energy:

$$E_{\text{tot}} = \sum_{i=1}^{M} E_{\text{com}}(i) + \sum_{i=1}^{M} E_{\text{comp}}(i) \text{ (Joule)},$$

(9)

under the (aforementioned) hard constraint $T_i (s)$ on the allowed per-job execution time. This last depends, in turn, on the (one-way) delays $(D(i), i = 1, \ldots, M)$ introduced by the LAN and the per-task processing time $A$ required by the VMs of Fig. 1. Specifically, since the $M$ connections of Fig. 1 are typically activated by the Switch Unit in a parallel fashion [8, Chapter 5], the overall two-way communication-plus-computing delay induced by the $i$th connection equates: $2D(i) + A$, so that the aforementioned hard constraint on the overall per-job execution time reads as in

$$\max_{1 \leq i \leq M} \{2D(i)\} + A \leq T_i.$$  

(10)

Thus, the overall CCOP assumes the following form:

$$\min_{\{\{R_i, L_i\}\}} \sum_{i=1}^{M} \Phi_i(\frac{L_i}{R_i}) \alpha_\ell(i) \sigma_{\text{max}} + k_c(f_i - f_{\text{max}})^2 + 2PP_{\text{com}}(R_i) \left(\frac{L_i}{R_i}\right),$$

(11.1)

s.t. : $(L_i + L_s(i)) \leq f_i A, \quad i = 1, \ldots, M,$

$$\sum_{i=1}^{M} L_i = L_t,$$

$$0 \leq f_i \leq f_{\text{max}}, \quad i = 1, \ldots, M,$$

$$L_i \geq 0, \quad i = 1, \ldots, M,$$

$$2L_i \frac{R_i}{R_s} + A \leq T_i, \quad i = 1, \ldots, M,$$

$$\sum_{i=1}^{M} R_i = R_s,$$

$$R_i \geq 0, \quad i = 1, \ldots, M.$$
Furthermore, $f_i^0$ and $f_i$ in (11.1) represent the current (i.e., already computed and consolidated) computing rate and the target one, respectively. Formally speaking, $f_i$ is the variable to be optimized, while $f_i^0$ describes the current state of the VM (i), and, then, it plays the role of a known constant. The constraint in (11.2) guarantees that the VM (i) executes the assigned task within $\Delta$ secs, while the (global) constraint in (11.3) assures that the overall job is partitioned into $M$ parallel tasks. According to (10), the set of constraints in (11.6) forces the NetDC to process the overall job within the assigned deadline $T_d$ and, then, it guarantees that the overall communication–computing platform of Fig. 1 operates in hard real-time. Finally, the global constraint in (11.7) limits up to $R_t$ (bits/s) the aggregate transmission rate sustainable by the underlying LAN of Fig. 1, so that $R_t$ is directly dictated by the actually implemented LAN technology (such as, for example, Myrinet, InfiniBand, Fast/Gigabit Ethernet, and so on; see, for example, [8]).

3.1. Generalizations of the CCOP’s formulation

Depending on the actually considered NetDC infrastructure, several generalizations of the reported CCOP’s formulation are possible. Specifically, as in [30], we assume constant the frequency-switching time-overhead induced by the adopted DVFS technique. However, it is direct to generalize the considered optimization problem in (11.1)(11.2)(11.3)(11.4)(11.5)(11.6)(11.7)(11.8) to the case when the frequency-switching time-overhead suffered from the VM (i) is described by a (nonnegative) function $\Psi_i(f_i, f_i^0, s)$, that retains the following two properties: (i) $\Psi_i(\cdot, \cdot, \cdot)$ is not decreasing in $[f_i - f_i^0]$; and, (ii) the product: $f_i \cdot \Psi_i(\cdot, \cdot, \cdot)$ is convex in the $f_i$ variable. In fact, it is direct to check that the addition of a such kind of $\Psi(f_i, \cdot)$ function at the l.h.s. of the constraint in (11.2) has no impact on the convexity properties retained by the resulting optimization problem. Furthermore, we also point out that, formally speaking, the summation: $\sum_{i=1}^M \Phi_i \left( f_i^0 / \bar{m} \right) \Omega(i) \delta_{i}^{\text{max}}$ of the computing energies in (11.1), and the summation: $\sum_{i=1}^M k_i \cdot (f_i - f_i^0)^2$ of the corresponding reconfiguration energies may be replaced by any two energy functions: $H(f_1, \ldots, f_M)$ and $V(f_1, \ldots, f_M)$, that are jointly convex in $[f_i, i = 1, \ldots, M]$.[4] Likewise, $\Delta$ in (11.2), (11.6) may be replaced by an $i$-depending $\Delta(i)$ without compromising the linear structure of the resulting constraints. Overall, we anticipate that the solving approach of the following Section 4 still applies verbatim to these (more) general formulations of the CCOP.

4. Optimal dynamic resource provisioning

The CCOP in (11) is not a convex optimization problem. This due to the fact that, in general, each function: $P_{i}^{\text{net}}(R_i)(L_i/R_i)$ in (11.1) is not jointly convex in $L_i$, $R_i$, even

\[ \text{in the simplest case when the power function } P_{i}^{\text{net}}(\cdot) \text{ reduces to an assigned constant. Therefore, neither guaranteed-convergence iterative algorithms nor closed form expressions are, to date, available to compute the optimal solution } \{ L_i, R_i, f_i, i = 1, \ldots, M \} \text{ of the CCOP. However, a direct examination of Eqs. (11.1), (11.2), (11.3), (11.4), (11.5), (11.6), (11.7), (11.8) unveils that the CCOP is a loosely coupled optimization problem (in the sense of [25]), where the variables } L_i, i = 1, \ldots, M \text{ couple the communication and computing sub-problems. In the sequel, we formally develop a solving approach that is based on the (lossless) decomposition of the CCOP into the (aforementioned) CMOP and CPOP.

Formally speaking, for any assigned nonnegative vector $\overline{L}$ of the tasks sizes, the CMOP is the (generally nonconvex) optimization problem in the communication rate variables $\{R_i, i = 1, \ldots, M\}$ so formulated:

\[ \min_{\{R_i\}} \sum_{i=1}^M 2P_{i}^{\text{net}}(R_i) / R_i, \quad (12.1) \]

s.t.: CCOP’s constraints in (11.6), (11.7), (11.8).

Let $\{R_i(\overline{L}), i = 1, \ldots, M\}$ be the optimal solution of the CMOP in (12), and let

\[ S = \left\{ \overline{L} \in (\mathbb{R}_+)^M : \left( L_i / R_i(\overline{L}) \right) \leq (T_t - \Delta) / 2, i = 1, \ldots, M; \sum_{i=1}^M R_i(\overline{L}) \leq R_t \right\}. \quad (13) \]

be the region of the (nonnegative) $M$-dimensional Euclidean space constituted by all $\overline{L}$ vectors meeting the constraints in (11.6)(11.7)(11.8). Thus, after introducing the dummy function:

\[ X(L_1, \ldots, L_M) \triangleq \sum_{i=1}^M 2P_{i}^{\text{net}}(R_i) / R_i, \quad (14) \]

the CPOP reads as:

\[ \min_{\{L_i \in L \}} \sum_{i=1}^M \left( \phi_i \left( f_i / f_i^0 \right) \delta_{i}^{\text{max}} + k_i (f_i - f_i^0)^2 \right) + X(L_1, \ldots, L_M), \quad (15.1) \]

s.t.: CCOP’s constraints in (11.2)(11.5) and $\overline{L} \in S$.

Let $\{L_i', R_i', f_i'; i = 1, \ldots, M\}$ indicate the (possibly, empty) set of the (possibly, not unique) solutions of the cascade of the CMOP and CPOP. The following Proposition 1 states that the cascade of these sub-problems is equivalent to the CCOP.\(^3\)

**Proposition 1.** The CCOP in (11.1)(11.8) admits the same feasibility region and the same solution of the cascade of the CMOP and CPOP, that is, $\{L_i, R_i, f_i, i = 1, \ldots, M\} \equiv \{L_i', R_i', f_i'; i = 1, \ldots, M\}$.

4.1. Resolution of the CMOP and closed-form characterization of the $S$ region

About the feasibility and solution of the CMOP, the following result holds.

\(^2\) From an application point of view, such kind of jointly convex energy functions may model the coupling effects that are typically present when the employed Virtualization Layer of Fig. 1 fail to provide “perfect” isolation among the instantiated VMs [24].

\(^3\) In order to simplify the notation, we understand the dependence of $R_i'$ on $L$ when it is not strictly mandatory.
Proposition 2. Let each function \( P_{\text{net}}^{\text{opt}}(R_i)/R_i \) in (12) be continuous and increasing for \( R_i \geq 0 \). Hence, for any assigned vector \( \overline{L} \), the following two properties hold:

(i) the CMOP in (12) is feasible if and only if the vector \( \overline{L} \) meets the following condition:
\[
\sum_{i=1}^{M} L_i \leq (R_i(T_i - \Delta))/2; \tag{16}
\]

(ii) the solution of the CMOP is given by the following closed-form expression:
\[
R_i^*(\overline{L}) \equiv R_i^*(L_i) \equiv (2L_i/(T_i - \Delta)), \quad i = 1, \ldots, M. \tag{17}
\]

Proof. For any assigned \( \overline{L} \), the objective function in (12) is the summation of \( M \) nonnegative terms, where the \( i \)th term depends only on \( R_i \). Thus, being the objective function in (12) separable, its minimization may be carried out component-wise. Since the \( i \)th term in (12) is increasing in \( R_i \) and the constraint in (11.6) must be met, the minimum is attained when the constraint in (11.6) is binding, and this proves the validity of (17). Finally, the set of rates in (17) is feasible for the CMOP if and only if the constraint in (11.7) is met, and this proves the validity of the feasibility condition in (16). □

4.2. Resolution of the CPOP and feasibility issues

Before proceeding, we remark that the validity of the Proposition 2 does not rely on any convexity assumption about the functions \( P_{\text{net}}^{\text{opt}}(R_i)/R_i \), \( i = 1, \ldots, M \), the assumption of increasing \( P_{\text{net}}^{\text{opt}}(R_i)/R_i \), \( R_i \geq 0 \), is compliant with the observation that larger communication rates demand for larger communication energies, as also confirmed by the power-rate functions in (3), (4). Furthermore, being the condition in (16) necessary and sufficient for the feasibility of the CMOP, it fully characterizes the admissible region \( S \) in (13).

Proposition 3. The CCOP in (11) is feasible if and only if the CPOP in (18) is feasible. Furthermore, the following set of \( (M + 2) \) conditions:
\[
L_0(i) \leq A f_i^{\text{max}}, \quad i = 1, \ldots, M, \tag{20.1}
\]
\[
L_i \leq \sum_{i=1}^{M} (f_i^{\text{max}} A - L_0(i)), \tag{20.2}
\]
\[
L_i = R_i/(2(T_i - \Delta)), \tag{20.3}
\]
is necessary and sufficient for the feasibility of the CPOP.

About the resolution of the CPOP, after denoting by \( \pi_i(\cdot) \) the following dummy function:
\[
\pi_i(f_i) = \left( \begin{array}{c} \omega(i) e_i^{\text{max}} \\ \xi f_i \\ \eta f_i \end{array} \right), \quad i = 1, \ldots, M, \tag{21}
\]
and by \( \pi_i^{-1}(\cdot) \) its inverse, let \( TH(i) \) be the nonnegative threshold so defined:
\[
TH(i) = 2(\partial P_{\text{net}}^{\text{opt}}(R_i)/\partial R_i)|_{R_i=0}, \quad i = 1, \ldots, M. \tag{22}
\]

Hence, after indicating by \( (\partial P_{\text{net}}^{\text{opt}}(R_i)/\partial R_i)^{-1}(y) \) the (nonnegative) inverse function of \( \partial P_{\text{net}}^{\text{opt}}(R_i)/\partial R_i \), the following Proposition 4 analytically characterizes the optimal scheduler.

Proposition 4. Let the feasibility conditions in (20) be met. Let the \( \lambda(\cdot) \) function in (19) be strictly jointly convex and let each function: \( P_{\text{net}}^{\text{opt}}(R_i)/R_i \), \( i = 1, \ldots, M \), be continuous and increasing for \( R_i \geq 0 \). Hence, the global optimal solution of the CPOP is unique and it is analytically computable as in:
\[
f_i^* = [\pi_i^{-1}(2k_i f_i^0 + v_i^*)]^{\text{max}}, \tag{23.1}
\]
\[
L_i^* = 1_{\{f_i^* - 0\}} f_i^* - L_0(i) + 1_{\{f_i^* = 0\}} \left[ \frac{(T_i - \Delta)}{2}(\partial P_{\text{net}}^{\text{opt}}(R_i)/\partial R_i)^{-1}(v_i^*) \right]. \tag{23.2}
\]

where \( f_i^{\text{min}} \leq f_i^0 \leq A f_i^{\text{max}} \), and the nonnegative scalar \( v_i^* \) is defined as in
\[
v_i^* = [\mu^* - 2(\partial P_{\text{net}}^{\text{opt}}(R_i)/\partial R_i)]_{+} \tag{24}
\]

Finally, \( \mu^* \in \mathbb{R}_0^+ \) is the unique nonnegative root of the following algebraic equation:
\[
\sum_{i=1}^{M} L_i^*(\mu) = L_i, \tag{25}
\]

where \( L_i^*(\cdot) \) is given by the r.h.s. of Eq. (23.2), with \( \mu^* \) replaced by the dummy variable \( \mu \).

The proof of the above relationships is reported in the final Appendix A. Before proceeding, we point out that the final solution of the overall CCOP is directly given by Eqs. (23.1), (23.2), with the optimal link flows given by (see Eq. (17)): \( R_i^* = (2L_i^*/(T_i - \Delta)), \quad i = 1, \ldots, M \).

5. Structural properties and adaptive online implementation of the optimal scheduler

About the main structural properties and implementation aspects of the optimal scheduler, the following remarks may be of interest.

5.1. Hibernation effects

Formally speaking, the \( i \)th VM is hibernated when \( L_i^* = 0 \) (i.e., no exogenous workload is assigned to the VM
(i) and the corresponding processing rate $f_i^*$ is strictly larger than the minimum one: $f_i^{\min} \leq L_s(i)/I$, requested for processing the background workload $L_s(i)$ (see the constraint in (11.2)). In principle, we expect that the hibernation of the VM $(i)$ may lead to energy savings when $k_e f_0$'s and the ratios $\{f_i^{\min}/R_i\}$'s are large, while the offered workload $L_i$ is small. As proved in the final Appendix A, this is, indeed, the behavior exhibited by the optimal scheduler, that hibernates the VM $(i)$ at the processing frequency $f_i^{*}$ in (23.1) when the following hibernation condition is met:

$$\mu^* \leq TH(i).$$

(26)

Interestingly, the ith hibernation threshold $TH(i)$ in (22) is fully dictated by the power-rate behavior of the ith transmission link of the utilized LAN, while the corresponding hibernated frequency in (23.1) depends on computing-related parameters (see Eq. (21)). This provides additional evidence of the tight computing-communication coupling induced by the green paradigm.

5.2. Online implementation issues and parameter tracking

As also pointed out in [13, Table 6.1], the minimum execution-time scheduling of independent tasks on parallel computing platforms is, in general, an NP-hard BinPacknig optimization problem, that resists closed-form solutions, even when the communication costs are not considered. From this point of view, remarkable features of the optimal scheduler of Proposition 4 are that: (i) it leads to distributed and parallel computation (with respect to the i-index) of the $3M$ variables $\{f_i^*, L_i, R_i\}$, $i = 1, \ldots, M$; and, (ii) its implementation complexity is fully independent from the size $L_i$ of the offered workload and the aggregate capacity $R_i$ of the utilized LAN.

Moreover, in time-varying environments characterized by (possibly, abrupt and random) time-fluctuations of the offered workload $L_i$ (see Section 6), the per-job evaluation $D_{\min}$ that assumes the following form [12]:

and online tracking of the Lagrange multiplier in (25) may be performed by resorting to a gradient-based updating, that assumes the following form [12]:

$$\mu^{(n)} = \left[\mu^{(n-1)} - \frac{\beta}{\gamma} \left(\sum_{i=1}^{M} l_i^{(n-1)} - L_i\right)\right],$$

$$\mu^{(0)} = L_i^{(0)} = 0,$$ (27)

where $n \geq 1$ is an integer value iteration index, $\{\mu^{(n-1)}\}$ is a (suitable) n-variant nonnegative step-size sequence, and the following dummy iterates in the n-index also hold (see Eqs. (23.1), (23.2) and (24)):

$$f_i^{(n)} = f_i^{(n-1)} - 2\beta f_i^{(n-1)} \left(\frac{\partial E_i^{(n-1)}}{\partial R_i} \right) \left(\frac{1}{T_i - A}\right),$$ (28.1)

$$l_i^{(n)} = f_i^{(n)} - L_i(i) = 1, \ldots, M.$$ (28.2)

Furthermore, an effective means for coping with the unpredictable time-variations of the offered workload is provided by the gradient-descendent algorithm of [12] for the adaptive updating of the step-size in (27). In our framework, this updating reads as in [12, Eq. (2.4)]:

$$\varphi^{(n)} = \max\left\{0; \min\left\{\beta; \frac{1}{\gamma} \left(\sum_{i=1}^{M} l_i^{(n-1)} - L_i\right)\right\}\right\},$$

(29)

where $\beta$ and $\gamma$ are positive constants, while $V^{(n-1)}$ is updated as in [12, Eq. (2.5)]:

$$V^{(n)} = (1 - \gamma^{(n-1)}) V^{(n-1)} - \left(\sum_{i=1}^{M} l_i^{(n-1)} - L_i\right) V^{(0)} = 0.$$ (30)

The convexity of the objective function in (18) and the existence and uniqueness of the solution of the CPOP allow the above iterations to converge to the optimum set $\{\mu^*; l_i^*; v_i^*; f_i^*; i = 1, \ldots, M\}$ of Proposition 4 [12].

From an application point of view, a (first) appealing feature of the step-size updating algorithm in (29) is its robustness against the actual tuning of the involved parameters $\beta$ and $\gamma$. As already pointed out in [12], we anticipate that, also in our framework, the final energy performance of the overall resource provisioning algorithm remains virtually unchanged for $\beta$ and $\gamma$ ranging over the intervals $[10^{-4}, 10^{-2}]$ and $[0.1, 0.6]$, respectively.

An additional appealing feature of (29) is that, some times, if the offered workload changes, it would be equivalent to re-start the overall iterates (27)(28)(29)(30) with the current workload as input [12]. Hence, it is expected that the presented gradient-based updating algorithm exhibits adaptive capabilities and the numerical results confirm, indeed, this expectation.

5.3. Adaptive setting of the computing time $A$

In principle, adapting the value of the allowed computing time $A$ in (11.2), (11.6) to the (possibly, abrupt) time-variations of the offered workload $L_i$ could provide an effective means for attaining additional energy savings. Unfortunately, due to the product form of the constraints in (11.2) and the combined presence of $A$ and $L_i$, $i = 1, \ldots, M$ in the $C(i)$ function in (19), the CPOP in (18) is no longer a jointly convex optimization problem when $A$ is included into the set of variables to be optimized. This implies, in turn, that the corresponding Karmarkar–Kuhn–Tucker (KKT) conditions are no longer sufficient for analytically characterizing the global optimum of the CPOP. However, as also remarked in [14, Section 5.1], the Fixed Point Iteration Method (FPM) (also referred to as Gauss–Seidel Component Solution method [32, Section 3.1]) provides an effective (albeit suboptimal) form for dealing with nonconvex optimization problems composed by the coupling of convex sub-problems. Specifically, by referring to the CPOP in (18), the FPM iteratively computes the optimum values of a subset of variables (e.g., $f_i, L_i, i = 1, \ldots, M$ or $A$), while the values of the other ones (e.g., alternatively, $A$ or $f_i, L_i, i = 1, \ldots, M$) are held fixed. The iteration stops when the improvement in the value of the objective function in (18) is less than an assigned (small) threshold over two consecutive iterations [32, Section 3.1.2]. In detail, let the feasibility conditions of
Proposition 3 be met. Then, when \( A \) is fixed to a specific previously computed value \( A^* \), the CPOP in (18) is solved with respect to \( \{ f_i, L_i, i = 1, \ldots, M \} \) according to Proposition 4. Viceversa, when \( \{ f_i, L_i, i = 1, \ldots, M \} \) are held fixed to specific previously computed values \( \{ f_i^*, L_i^*, i = 1, \ldots, M \} \), the CPOP is solved with respect to \( A \). Under the (additional) assumption that the \( \lambda(\cdot) \) function in (19) is convex in \( A \) for any assigned \( \{ f_i, L_i, i = 1, \ldots, M \} \), the constrained minimization of the objective function in (18) over \( A \) simply reduces to [22, Chapter 4]: (i) equate to zero the derivative \( \partial \lambda(\cdot) / \partial A \), and, then, solve the resulting algebraic equation with respect to \( A \); and, (ii) project the so obtained solution onto the following closed interval:

\[
\max_{i=1,...,M} \left\{ \left( L_i^* + L_0(i) \right) / f_i^* \right\}, T_i - (2L_i/R_i) \]

that formally accounts for the box constraints in (11.2), (11.6) and (11.7) on the feasible \( A^* \)'s values.

The proposed FPIM allows an adaptive setting of \( A \) at run-time. Formally speaking, since, in our framework, the analytical conditions of [32, Proposition 1.7] are not assured to hold, we cannot guarantee that the proposed FPI procedure converges to a global optimum. However, we can state that the FPI procedure always converges to (at least) a local optimum. In fact, since, at each iteration, the optimal solution of each component sub-problem is computed, the current \( A^* \) and \( \{ f_i^*, L_i^*, i = 1, \ldots, M \} \) assignments are improved. This formally guarantees that the overall FPI algorithm always approaches a solution that is (at least) locally optimal.

5.4. Cases of study and application examples

From an application point of view, it should be remarked that the Shannon-Hartley- and \( \alpha \)-powered relationships in (3), (4) as well as the quadratic computing energy function in (2) are examples of functions of practical interest that meet all the assumptions of Proposition 4. Specifically, for the case of the Shannon-Hartley’s formula in (3), the derivative in (24) and its inverse specialize to

\[
\frac{\partial \psi_{\text{per}}^i(R_i)}{\partial R_i} = (2n) \left( \frac{X_0}{E_i} \right)^{\frac{1}{2n}}, \quad (\partial \psi_{\text{per}}^i / \partial R_i)^{-1}(y) = W_i \log_2 \left( \frac{2y}{TH(i)} \right).
\]

with (see Eq. (22)): \( TH(i) \equiv (2n) \lambda(\cdot) / \gamma(i) \).

Finally, for the case of the quadratic computing energy function in (2), the expression in (23.1) for the ith optimal processing frequency is directly computable as in

\[
f_i^* = \left[ \frac{2K_0 f_i^{00} + \gamma(i) A_{\text{max}}}{2K_0 + 2 \alpha(i) e_{\text{max}}^i / (f_i^{00})^2} \right]_{i=1}^{\text{max}}.
\]

The above examples support the practical effectiveness of the proposed solving approach.

6. Numerical results and performance comparison

We numerically evaluate the per-job average communication-plus-computing (e.g., total) energy \( \epsilon_{\text{tot}} \) consumed by the proposed optimal scheduler under both synthetically generated and measured realistic workload traces.

6.1. Simulated stochastic setting

Specifically, in order to stress the effects of the reconfiguration costs and the time-fluctuations of the offered workload on the energy performance of the simulated schedulers, as in [19], we model the workload size as an independent identically distributed (i.i.d.) random sequence \( \{L_i(mT_i), m = 0, 1, \ldots \} \), whose samples are \( T_i \)-spaced apart r.v.s that are uniformly distributed over the interval \([T_i - a, T_i + a]\), with \( T_i \equiv 8 \) (Mbit). By setting the spread parameter \( a \) to 2 (Mbit), 4 (Mbit), 6 (Mbit) and 8 (Mbit), we obtain PMRs of 1.25, 1.5, 1.75 and 2.0, respectively. These PMRs are quite common in large-to-medium size data centers, that perform in real-time Internet-based content delivery [10]. About the dynamic setting of \( f_i^0 \) in (11.1), at the first round of each batch of the carried out simulations, all the frequencies \( f_i^0 \)'s are reset. Afterwards, at the mth round, each \( f_i^0 \) is set to the corresponding optimal value \( f_i^* \) computed at the previous \((m - 1)\)th round. Furthermore, unless otherwise stated, the presented numerical results refer to the Shannon-Hartley’s power-rate function in (3), together with the quadratic computing energy function in (2). Each simulated point has been numerically evaluated by averaging over 1000 independent runs.

6.2. Impact of the LAN setup, hibernation phenomena and reconfiguration costs

Goal of a first set of numerical tests is to evaluate the effects on the per-job average consumed energy \( \epsilon_{\text{tot}} \) of the size \( M \) of the NetDC and the setting of the bandwidths \( \{ W_i \} \), noise levels \( \{ X_0^0 \} \) and link gains \( \{ g_i \} \) of the employed LAN. For this purpose, we set [28]: \( T_i = 5 \) (s), \( R_i = 100 \) (Mbit/s), \( PMR = 1.25 \), \( K_0 = 0.05 \) (/MHz), \( f_{\text{max}} \equiv 105 \) (Mbit/s), \( \alpha(i) = 1 \), \( e_{\text{max}} = 60 \) (Joule), \( A = 0.1 \) (s), \( W_i = 1 \) (MHz) and \( L_{\text{th}}(i) = 0 \).

Afterwards, since the bandwidth/noise/link-gain effects are captured by the corresponding ratios \( \zeta(i) \) (see (3)), we have numerically evaluated the average total energy consumption \( \epsilon_{\text{tot}} \) of the proposed optimal scheduler under the following settings: (i) \( \zeta(i) = 0.2 \) (mW); (ii) \( \zeta(i) = 0.5 \) (mW); (iii) \( \zeta(i) = [0.5 + 0.25(i - 1)] \) (mW); and, (iv) \( \zeta(i) = [0.5 + 0.5(i - 1)] \) (mW), \( i = 1, \ldots, M \). The obtained numerical plots are drawn in Fig. 2. As it could be expected, larger \( \zeta(i)'s \) penalize the overall energy performance of the simulated NetDC. Interestingly, since \( \epsilon_{\text{tot}} \) is, by definition, the minimum energy when up to \( M \) VMs may be instantiated, at fixed positive \( P_{\text{per}} \)'s, \( \epsilon_{\text{tot}} \) decreases for increasing \( M \) and, then, it approaches a minimum value that does not vary when \( M \) is further increased (see the flat segment of the two uppermost plots of Fig. 2).

An instance of hibernation of the instantiated VMs is exemplified by the plots of Fig. 3. They refer to the considered application scenario with \( \zeta(i) = [0.5 + 0.5(i - 1)] \) (mW), \( f_{i}^{0} = 0.2 f_{i}^{\text{max}}, i = 1, \ldots, M \). Specifically, the upper bars of Fig. 3 report the (numerically evaluated) optimal average processing rates \( f_i^* \)'s, while the lower bars refer to the
corresponding optimal average workloads $L_i$’s. An examination of the lower bars of Fig. 3 points out that only the first nine VMs are permanently loaded, while the corresponding upper bars confirm, indeed, that all the available VMs constantly run at positive processing rates. This means that, in the considered application scenario, the last three VMs are hibernated by the optimal scheduler.

About this last aspect, Fig. 4 reports the effects of the reconfiguration costs on $E_{tot}$ at $k_e = 0.005$ (Joule/(MHz)$^2$), $k_e = 0.05$ (Joule/(MHz)$^2$) and $k_e = 0.5$ (Joule/(MHz)$^2$). Interestingly, these plots show that $E_{tot}$ increases for growing $k_e$’s only for small values of $M$, while the optimal number $M^*$ of VMs to be instantiated (e.g., the right size of the Net-DC) decreases for increasing $k_e$’s.

6.3. Performance comparison under synthetic time-uncorrelated workload profiles

These conclusions are also supported by the numerical results of this subsection. They aim at unveiling the impact of the PMR of the offered workload on the average energy consumption of the proposed scheduler and comparing it against the corresponding ones of two state-of-the-art schedulers, namely, the STATIC Scheduler (STAS) and the SEquential Scheduler (SES)[15]. Intuitively, we expect that the energy savings attained by dynamic schedulers increase when DVFS-enabled VMs are used, especially at large PMR values. However, we also expect that not negligible reconfiguration costs (e.g., large $k_e$’s) may reduce the attained energy-saving and that the experienced reductions tend to increases at large PMRs. In order to validate these expectations and attain insight about the underlying tradeoff, we have implemented the communication–computing platform of Section 6.2 at $\zeta = 0.5$ (mW).

About the implemented STAS, we note that current data centers usually rely on static resource provisioning, where, by design, a fixed number $M_s$ of computing machines constantly run at the maximum processing rate $p_{\text{max}}$. The goal is to constantly provide the exact computing capacity needed for satisfying the peak workload $L_1^\text{max} \triangleq (L_1 + \alpha)$ (Mb). Although the resulting static scheduler does not experience reconfiguration costs, it induces overbooking of the computing resources [15]. Hence, the per-job average communication-plus-computing energy consumption $E_{tot}^{\text{STAS}}$ (Joule) of the STAS gives a benchmark for numerically evaluating the actual energy-saving attained by dynamic schedulers [15]. In the application framework considered here, the energy consumption $E_{tot}^{\text{STAS}}$ of the static scheduler is computable in closed-form and equates

$$E_{tot}^{\text{STAS}} = \frac{M_s \left[ L_1^\text{max} \left( C_0 L_1 \sinh \left( aC_0 \right) / \left( aC_0 - 1 \right) \right) + \alpha \right]}{\left[ L_1 + \alpha \right] / \left( aC_0 - 1 \right)} + \frac{C_0 \cdot \left( L_1 + \alpha \right) / \left( aC_0 - 1 \right)}{\left[ L_1 + \alpha \right] / \left( aC_0 - 1 \right)}.$$

where $M_s \triangleq \left[ L_1^\text{max} / \left( (L_1 + \alpha) / \left( aC_0 - 1 \right) \right) \right]$ is the number of constantly running machines that the STAS must utilize for satisfying the peak workload $L_1^\text{max}$. [...] is the ceiling function and the (dimension-less) parameter $C_0$ is defined as: $C_0 \triangleq (2\ln2)/(M_sW(T_1 - \alpha))$.

About the implemented SES, we observe that, by design, it exploits perfect future workload information over a
time-window of size $I \geq 2$ (measured in multiple of the slot period $T_s$), in order to perform off-line resource provisioning at the minimum reconfiguration cost [15]. Formally speaking, the SES implements the solution of the following constrained sequential optimization problem:

$$
\min_{\{R(m)|R(m)|L(m)|}\sum_{m=1}^{W} \sum_{i=1}^{I} \left\{ \phi_i \left( f_i(m) / f_i^{\max} \right) o_{max} e_i \right\} + k_e \left( f_i(m) - f_i(m-1) \right)^2 + 2 R_i \left( L_i(m) / R_i(m) \right) }^{(34)}
$$

under the constraints in (11.1)-(11.8). We have evaluated the solution of the above sequential optimization problem by resorting to numerical computing tools, that we have implemented through Matlab routines. In doing so, we have also numerically ascertained that, at least in the simulated application scenarios, $I = 10$ slots suffice for attaining full energy-saving. Since the SES operates off-line, it cannot be really employed in hard real-time applications. However, by design, it fixes the ultimate performance attainable through dynamic resource provisioning policies. Hence, it allows us to (numerically) evaluate the ultimate performance loss suffered by dynamic schedulers that must work in real-time.

Table 1 and Table 2 report, the average energy savings (in percent) provided by the proposed scheduler and the sequential scheduler over the static one for the cases of medium reconfiguration costs (e.g., $k_e = 0.05$ (Joule/(MHz)^2)) and high reconfiguration costs (e.g., $k_e = 0.5$ (Joule/(MHz)^2)).

An examination of the numerical results reported in Table 1 leads to two main conclusions. First, the average energy-saving of the proposed dynamic scheduler over the static one approaches 94% at $PMR = 2$, even when the VMs are equipped with a limited number $Q = 6$ of discrete processing rates. Second, the performance loss suffered by the proposed (online) scheduler with respect to the sequential (off-line) one tends to increase for growing PMRs, but it remains limited up to 5–6%. The same conclusions may be drawn from the data of Table 2. Specifically, although they refer to a computing platform with reconfiguration costs that are ten times higher than those of Table 1, the average energy reduction of the proposed scheduler over the static one still approaches 92% at $PMR = 2$, while the corresponding energy loss with respect to the sequential scheduler is still limited up to 4–5%.

<table>
<thead>
<tr>
<th>PMR</th>
<th>Proposed scheduler (%)</th>
<th>SES (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.25</td>
<td>87</td>
<td>91</td>
</tr>
<tr>
<td>1.5</td>
<td>91</td>
<td>95</td>
</tr>
<tr>
<td>1.75</td>
<td>93</td>
<td>98</td>
</tr>
<tr>
<td>2</td>
<td>92</td>
<td>96</td>
</tr>
</tbody>
</table>

6.4. Performance comparison under time-correlated workload profiles

In order to test the energy performance of the proposed scheduler when the sequence of the offered workload is time-correlated and the FPI-based dynamic tuning of $A$ of Section 5.3 is also implemented, we have considered the realistic workload trace reported in Fig. 14a of [29], that, as pointed out in [29, Section 6.1], is representative of an 1-h HTTP-type session arrival process actually measured at the Web servers of the 1998 Soccer World Cup site (see [29, Section 6] and references therein). The numerical tests carried out in this sub-section refer to the communication–computing infrastructure of Section 6.3 at $k_e = 0.5$ (Joule/(MHz)^2).

Furthermore, in order to maintain the peak workload still fixed at 16 (Mbit/slot), we assume each arrival of the HTTP sessions in [29, Fig. 14a] carries out a workload of 0.533 (Mbit). Hence, by referring to the described application scenario, a first set of numerical trials has been performed by statically setting $A$ at 1.2 (s). We have numerically evaluated that, in this case, the average energy reduction of the proposed scheduler over the static one of Eq. (33) is around 27%, while the corresponding average energy reduction of the sequential scheduler over the proposed one is of the order of 5.5%. Afterwards, in a second set of numerical trials, we have still initialized $A$ at 1.2 (s), and, then, we have included into the implementation of the proposed scheduler the FPI-based procedure of Section 5.3.

Hence, on the basis of the obtained numerical results, we have experienced that the average energy reduction of the FPI-equipped proposed scheduler over the static one approaches 40–42%, while the corresponding average energy loss with respect to the sequential scheduler scales down to 2.5–3.0%.

7. Conclusion

In this paper, we developed the optimal online scheduler for the joint dynamic load balancing and provisioning of the computing rates, communication rates and communication powers in self-configuring energy-saving NetDCs that operate under hard real-time constraints. Although the resulting optimization problem is inherently nonconvex, we unveil and exploit its loosely coupled structure for attaining the analytical characterization of its solution. The carried out numerical performance comparisons highlight that the average energy savings provided by the
proposed dynamic scheduler over the state-of-the-art static one may approach 90%, even when the offered workload is limited up to 2 and the number of different processing rates equipping each computing machine is limited up to 5–6. Interestingly, the corresponding average energy loss of the proposed scheduler with respect to the corresponding sequential one is limited up to 4–6%, especially when the offered workload exhibits not negligible time-correlation.

Appendix A. Derivations of Eqs. (23.1)–(25)

Being the constraint in (11.7) already accounted for by the feasibility condition (20.3), without loss of optimality, we may directly focus on the resolution of optimization problem in (18) under the constraints in (11.2), (11.3), (11.4), (11.5). Since this problem is strictly convex and all its constraints are linear, the Slater’s qualification conditions hold [22, Chapter 5], so that the KKT conditions [22, Chapter 4] are both necessary and sufficient for analytically characterizing the corresponding unique optimal global solution. Before applying these conditions, we observe that each power-rate function in (19) is increasing for \( L_i \geq 0 \), so that, without loss of optimality, we may replace the equality constraint in (11.3) by the following equivalent one: \( \sum_{i=1}^{M} L_i \geq L_c \). In doing so, the Lagrangian function of the afforded problem reads as

\[
\mathcal{L}(L_i, f_i, \nu, \mu) = \mathcal{Z}(L_i, f_i) + \sum_{i=1}^{M} \nu_i (L_i - f_i A + L_b(i)) + \mu \left( L_i - \sum_{i=1}^{M} L_i \right),
\]

(A.1)

where \( \mathcal{Z}(L_i, f_i) \) indicates the objective function in (18.1), \( \nu_i \)'s and \( \mu \) are nonnegative Lagrange multipliers, and the box constraints in (11.4), (11.5) are managed as implicit ones. The partial derivatives of \( \mathcal{L}(\cdot; \cdot) \) with respect to \( f_i, L_i \) are given by

\[
\frac{\partial \mathcal{L}(\cdot)}{\partial f_i} = \omega(i) e^{\max} \frac{\partial \phi_i(f_i; e^{\max})}{\partial \eta_i} + 2k_b(f_i - f_i^0) - \nu_i, \quad i = 1, \ldots, M,
\]

(A.2)

\[
\frac{\partial \mathcal{L}(\cdot)}{\partial L_i} = 2 \frac{\partial}{\partial R_i} \left( \frac{L_i}{T_i - A} \right) + \nu_i - \mu, \quad i = 1, \ldots, M,
\]

(A.3)

while the complementary conditions [22, Chapter 4] associated to the constraints present in (A.1) read as

\[
\nu_i (L_i - f_i A + L_b(i)) = 0, \quad i = 1, \ldots, M; \mu \left( L_i - \sum_{i=1}^{M} L_i \right) = 0.
\]

(A.4)

Hence, by equating to zero Eq. (A.2) and, then, by solving the resulting algebraic equation with respect to \( f_i \) we directly arrive at Eq. (23.1), that also accounts for the box constraint: \( f_i^{\min} \leq f_i \leq f_i^{\max} \) through the corresponding projector operator. Moreover, a direct exploitation of the last complementary condition in (A.4) allows us to compute the optimal \( \mu^* \) by solving the algebraic equation in Eq. (25). In order to obtain the analytical expressions for \( L_i^* \) and \( \nu_i^* \), we proceed to consider the two cases of \( \nu_i^* > 0 \) and \( \nu_i^* = 0 \). Specifically, when \( \nu_i^* > 0 \), the ith constraint in (11.2) is binding (see Eq. (A.4)), so that we have

\[
L_i^* = f_i^* - L_b(i), \quad \text{at } \nu_i^* > 0.
\]

(A.5)

Hence, after equating to zero Eq. (A.3) and solving the resulting algebraic equation with respect to \( \nu_i \), we obtain the following expression for the corresponding optimal \( \nu_i^* \):

\[
\nu_i^* = 2 \left[ \frac{\partial P_i^{\text{net}}}{\partial R_i} \left( \frac{L_i^*}{T_i - A} \right) \right], \quad \text{at } \nu_i^* > 0.
\]

(A.6)

Since \( L_i^* \) must fall into the closed interval \( [0, A f_i^* - L_b(i)] \) for feasible CPOPs (see Eqs. (11.2), (11.5)), at \( \nu_i^* = 0 \), we must have: \( L_i^* = 0 \) or \( 0 < L_i^* < A f_i^* - L_b(i) \). Specifically, we observe that, by definition, vanishing \( L_i^* \) is optimal when \( \frac{\partial \mathcal{L}(\cdot)}{\partial L_i} \|_{i=0} = 0 \). Therefore, by imposing that the derivative in (A.3) is nonnegative at \( L_i^* = 0 \), we obtain the following condition for the resulting optimal \( \mu^* \):

\[
\mu^* \leq 2 \left[ \frac{\partial P_i^{\text{net}}(R_i)/\partial R_i}{R_i} \right]_{i=0} \triangleq \text{TH}(i),
\]

(A.7)

at \( \nu_i^* = L_i^* = 0 \). \( i = 1, \ldots, M, \)

Eq. (A.8) vanishes at \( \mu^* = \text{TH}(i) \) (see Eq. (A.7)), and this proves that the function: \( L_i^*(\mu^*) \) vanishes and is continuous at \( \mu^* = \text{TH}(i) \). Therefore, since Eq. (A.7) already assures that vanishing \( L_i^* \) is optimal at \( \nu_i^* = 0 \) and \( \mu^* \leq \text{TH}(i) \) and the (aforementioned) KKT optimality condition leading to Eq. (A.8) is unique, we conclude that the expression in (A.8) for the optimal \( L_i^* \) must hold when \( \nu_i^* = 0 \) and \( \mu^* \geq \text{TH}(i) \). This structural property of the optimal scheduler allows us to merge Eqs. (A.7), (A.8) into the following equivalent expression:

\[
L_i^* = (T_i - A) \left[ \left( \frac{\partial P_i^{\text{net}}(R_i)/\partial R_i}{R_i} \right) \frac{\mu^*}{2} \right], \quad \text{for } \nu_i^* = 0.
\]

(A.9)

so that Eq. (23.2) directly arises from Eqs. (A.5), (A.9). Finally, after observing that \( \nu_i^* \) cannot be negative by definition, from Eq. (A.6) we obtain Eq. (24), where the projector operator accounts for the nonnegative value of \( \nu_i^* \). This completes the proof of Proposition 4.

References
