An Energy-aware Scheduling Algorithm in DVFS-enabled Networked Data Centers

Mohammad Shojafar[1], Claudia Canali[1], Riccardo Lancellotti[1] and Saeid Abolfazli[2]
[1]*Department of Engineering "Enzo Ferrari", University of Modena and Reggio Emilia, Modena, Italy*
[2]*YTL Communications - Xchanging Malaysia, Malaysia*
{*mohammad.shojafar, claudia.canali, riccardo.lancellotti*}*@unimore.it, Abolfazli@ieee.org*

Keywords: Virtualized Networked Data Centers, Optimization, Dynamic Voltage Frequency Scaling, Resource provisioning, Energy-efficiency.

Abstract: In this paper, we propose an adaptive online energy-aware scheduling algorithm by exploiting the reconfiguration capability of a Virtualized Networked Data Centers (VNetDCs) processing large amount of data in parallel. To achieve energy efficiency in such intensive computing scenarios, a joint balanced provisioning and scaling of the networking-plus-computing resources is required. We propose a scheduler that manages both the incoming workload and the VNetDC infrastructure to minimize the communication-plus-computing energy dissipated by processing incoming traffic under hard real-time constraints on the per-job computing-plus-communication delays. Specifically, our scheduler can distribute the workload among multiple virtual machines (VMs) and can tune the processor frequencies and the network bandwidth. The energy model used in our scheduler is rather sophisticated and takes into account also the internal/external frequency switching energy costs. Our experiments demonstrate that the proposed scheduler guarantees high quality of service to the users respecting the service level agreements. Furthermore, it attains minimum energy consumptions under two real-world operating conditions: a discrete and finite number of CPU frequencies and not negligible VMs reconfiguration costs. Our results confirm that the overall energy savings of data center can be significantly higher with respect to the existing solutions.

1 Introduction

Energy-saving computing through Virtualized Networked Data Centers (VNetDCs) is an emerging paradigm that aims at performing the adaptive energy management of virtualized computing platforms (Baliga et al., 2011; Canali and Lancellotti, 2016). The goal is to provide high quality Internet services to large populations of clients, while minimizing the overall computing-plus-networking energy consumption (Cugola and Margara, 2012; Mishra et al., 2012; Baliga et al., 2011). Nowadays, the energy cost of the communication infrastructure for current data centers may represent a significant fraction of the overall system due to the presence of switches, routers, load balancers and other network devices (Azodolmolky et al., 2013; Warneke and Kao, 2011). In our scenario we consider a VNetDC that receives as input a set of computationally-intensive jobs and operates on such input data. An example of this type of applications is cloud based data processing based on a map-reduce paradigm. In this scenario, we define the quality of service (QoS) requirements as a threshold on the per-job execution time. To support this type of Service Level Agreement (SLA) the VNetDC must be able to quickly adapt its resource allocation to the current (a priori unpredictable) size of the incoming traffic. A final requirement for our data center is to minimize energy consumption, an aspect that is receiving a growing amount of interest in the scientific literature (Chase et al., 2001; Herbert and Marculescu, 2007; Canali and Lancellotti, 2014). New energy-aware CPU technologies, such as the Dynamic Voltage and Frequency Scaling (DVFS) (Herbert and Marculescu, 2007), are rapidly being adopted for data center energy provisioning. The paradigm of cloud computing relying on virtualization is another characteristic we should take into account in our system. Several approaches (i.e., (Urgaonkar et al., 2010; Mathew et al., 2012; Wang et al., 2014; Cordeschi et al., 2013) highlighted these concepts within the energy management. Specifically, authors in (Urgaonkar et al., 2010) considered optimal resource allocation and power management in VNetDCs with

heterogeneous applications; however, they do not take into account re-configuration and network costs. Authors in (Mathew et al., 2012) proposed a new method to reduce the energy consumption of large Internet-scale distributed systems by modeling the problem into offline algorithm and an online algorithm to extract energy savings both at the level of local load balancing within a data center and global load balancing across data centers. The drawback of this method is that it cannot manage the spikes and valleys of incoming workloads, and does not to control the internal/external switch of the server frequencies. The authors of (Wang et al., 2014) introduce an algorithm to minimize the number of switches that will be used and to balance network traffics and handle the data center energy. This mathematical approach can be applied in large-scale data centers, but it is unable to manage the tear-and-wear of the server, the workload fluctuations and does not consider inter-costs for reconfiguration among various discrete ranges of frequencies. Finally, the works in (Cordeschi et al., 2013; Cordeschi et al., 2014; Shojafar et al., 2015) concentrate on the computing-plus-communication energy consumed for the several components of the VNetDCs and try to manage the entire energy of data centers respecting the considered SLAs, but did not emphasize the internal switching costs occurring at the VMs level. In particular, the work in (Shojafar et al., 2015) is based on a simplified approach for the computation of the communication costs, that does not consider the Shannon-Hartley model; moreover, the results are compared with a limited number of state-of-the-art alternatives.

In this paper, we propose a new approach to minimize energy consumption in computing, communication and reconfiguration costs in a scenario of parallel data processing based on cloud computing, while satisfying SLAs that are expressed as the maximum time to process a job (including computation and communication times). A qualifying contribution of our research is that we consider an energy objective model that is a non-convex function. Hence, we propose a mathematical approach to change non-convexity into convexity. Additional features of our approach are its scalability, easy implementation, and independence of workload scheduling from the reconfiguration costs. The remainder of this paper is organized as follows. After presenting the system model in Section 2, the approach and the mathematical proofs which cover computation-plus-communication objective functions and the optimization problem constraints are introduced in Section 3. Numerical results are presented in Section 4. Finally, Section 5 summarizes the main results and outlines future research directions.

2 System Model

The considered VNetDC is modeled as multiple virtualized processing units interconnected by a single-hop virtual network and managed by a central controller. Each processing unit executes the currently assigned task using its own local virtualized storage and computing resources. When a request for a new job is submitted to the VNetDC, the central resource controller dynamically performs both admission control and allocation of the available virtual resources (Almeida et al., 2010) as in Fig. 1.

We recall that the model for a VNetDC adopted in this paper follows the emerging trends for communication-plus-computing system architecture. A VNetDC is composed by multiple reconfigurable VMs, that are interconnected by a throughput-limited switched Virtual Local Area Network (VLAN). We assume a star-topology VLAN where, in order to guarantee inter-VM communication, the Network Switch of Fig. 1 acts as a gather/scatter central node. The operations of both VMs and VLAN are jointly managed by a Virtual Machine Manager (VMM), which performs task scheduling by dynamically allocating the available virtual computing-plus-communication resources to the VMs and Virtual Links of Fig. 1. A new job is initiated by the arrival of a data of size L_{tot} $[bit]$. Due to the SLA formulation, full processing of each input job must be completed within assigned and deterministic processing time which spans $\overline{T}$ seconds. $M \geq 1$ is the maximum number of VMs in the data center. In this paper, we assume that the VMs deployed over a server can change their share of server resources according to the model described in (Gmach et al., 2012), that is widely adopted in private cloud environments. This model tends to face conditions of high computational demand by means of few large VMs instead of many small VMs. For the sake of this research, we adopt a simplified model where a single VM is deployed over each server and uses all the available resources for that server. At any given time, the physical server CPU operates at a frequency f_i (chosen within a predefined set of frequencies available from the DVFS technology). A VM on server i is capable to process $F(i)$ bits per second as in (Cordeschi et al., 2013), where the processing rate $F(i)$ of VM i is linearly proportional to the CPU frequency $f(i)$. An extension of the model to consider more VMs on each physical server is left as an open issue to be addressed in future works.

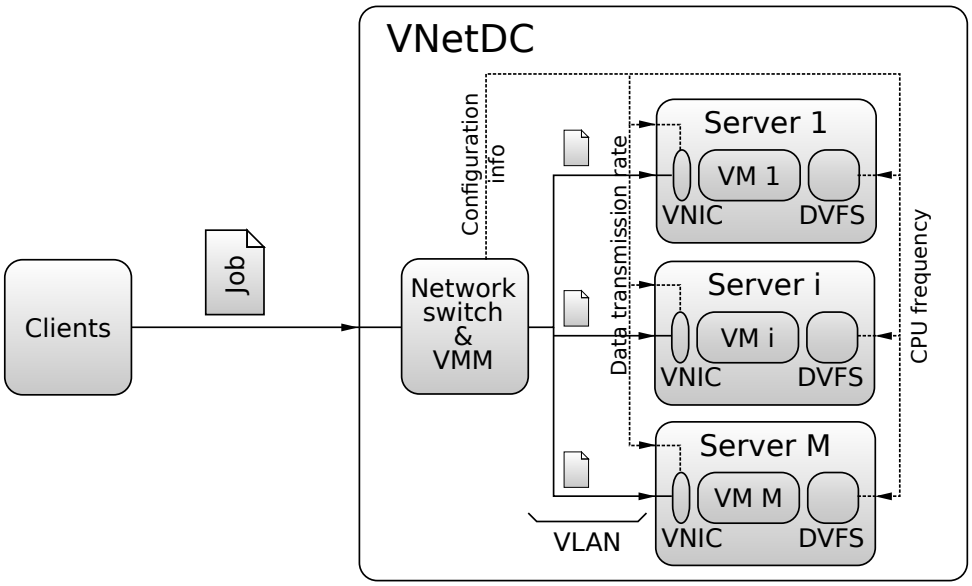


Figure 1: The considered VNetDC architecture.

2.1 Computational Cost

The adopted model for the computing energy is based on the CPU energy curve and VM states (we recall that each physical server hosts only one VM). DVFS is applied by the hosting physical servers to stretch the processing times of the tasks and reduce the energy consumptions by decreasing the CPU frequencies of the active VMs. For this purpose, each server can be operated at multiple voltages which operate different CPU frequencies (Azodolmolky et al., 2013). Q is the number of CPU frequencies allowed for each VM (plus an idle state). The set of the allowed frequencies is $f(i) \in \{f_j(i)\}$, with $i \in \{1,\ldots,M\}, j \in \{0,1,\ldots,Q\}$, where $f_j(i)$ is the $j-th$ discrete frequency of VM i, with $j=0$ representing the idle state. Furthermore, we define $t_j(i)$ as the time where i-th VM operates at frequency $f_j(i)$. Fig. 2 illustrates an example for $Q=5$.

According to (Qian et al., 2013), the dynamic power consumption P of the hosting CPU grows with the third power of the CPU frequency. So we can define the energy consumption of the generic VM i as:

$$\varepsilon_{CPU}(i) \triangleq \sum_{j=0}^{Q} AC_{eff} f_j(i)^3 t_j(i), [Joule], \forall i = \{1,\ldots,M\}, \quad (1)$$

where A, C_{eff} and f represents the active percentage of gates, effective load capacitance, and the processor frequency of the considered CPU, respectively.

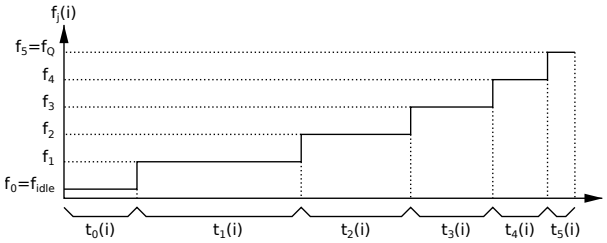


Figure 2: The discrete range of frequencies considered for $VM(i)$

2.2 Frequency Reconfiguration Cost

For the CPU frequency reconfiguration (switching) cost, we need to consider two costs: *internal switching cost* and *external switching cost*. The first one is the cost of changing the internal-switching among discrete frequencies of $VM(i)$ from $f_j(i)$ to $f_{j+k}(i)$ (i.e., k steps movement to reach the next active discrete frequency). The *second one* is the cost for external-switching from the final active discrete frequency of VM i at the end of a job to the first *active discrete frequency* for the next incoming job of size L_{tot}. Note that the *active discrete frequencies* are a subset of the available operating frequencies found based on their related times-quota variables, which means that frequency $f_j(i)$ belongs to the set of active discrete frequencies if and only if $t_j(i) > 0$. Given the list of *active discrete frequencies* for each VM for each job coming into the system, switching from the current active discrete frequency to another one affects the reconfiguration cost. We start from the first active discrete frequency ($f_k(i)$), move to the second

one ($f_{k+1}(i)$) and so on. We define the differences as $\Delta f_k(i) \triangleq f_{k+1}(i) - f_k(i)$ and the cost is $k_e \Delta f_k(i)^2$, where k_e $[Joule/(Hz)^2]$ is the reconfiguration cost induced by a unit-size frequency switching. Typical values of k_e for current DVFS-based virtualized computing platforms are limited up to few hundreds of μJ per $[MHz]^2$ (Cordeschi et al., 2014). If we consider homogeneous VMs, the total cost of internal-switching for all VMs is: $k_e \sum_{i=1}^{M} \sum_{k=0}^{K} (\Delta F_k(i))^2$, where $k \in \{0,1,\ldots,K\}$. $K \leq Q$ is the number of active discrete frequencies for VM i. The external-switching cost is calculated as multiplication of k_e with the quadratic differences between the last active discrete frequency of i-th VM for the current job and the first active discrete frequency of i-th VM in the next incoming job, which is named Ext_Cost. In a nutshell, the total reconfiguration energy can be written as (2):

$$\sum_{i=1}^{M} \varepsilon_{Reconf}(i) \triangleq k_e \sum_{i=1}^{M} \sum_{k=0}^{K} (\Delta f_k(i))^2 + k_e \sum_{i=1}^{M} Ext\_Cost \tag{2}$$

In the worst case, $K = Q+1$ and for external-switching we need to move Q steps to f_0 (Idle state). In this case the internal-switching cost is $k_e M \sum_{k=0}^{Q} (\Delta F_k)^2$ and the external-switching cost is $k_e M (f_Q^t - f_0^{t-1})^2$.

2.3 Communication Cost

We assume that each VM i communicates to the scheduler through a dedicated (i.e., contention-free) reliable virtual link, that operates at the transmission rate of $R(i)$ $[bit/s]$, $i = 1,\ldots,M$ and it is equipped with suitable Virtual Network Interface Cards (VNICs) (as in Fig. 1). The one-way transmission-plus-switching operation over the i-th virtual link drains a (variable) power of $P^{net}(i) = P^T_{net}(i) + P^R_{net}(i)$ $[Watt]$, where $P^T_{net}(i)$ is the power consumed by the transmit VNIC and Switch and $P^R_{net}(i)$ is related to VNIC receiving operations. We assume that the channel power of transmitting and receiving are the same and can be calculated according to the Shannon-Hartley exponential formula as

$$P^{net}(i) = \zeta_i \left(2^{R(i)/W_i} - 1\right) + P^{idle}(i), \, [Watt], \tag{3}$$

with $\zeta_i \triangleq \frac{\mathcal{N}_0(i) W_i}{g_i}$, $i = 1,\ldots,M$, where $\mathcal{N}_0(i)$, $[W/Hz]$, W_i $[Hz]$ and g_i are noise spectral power density, transmission bandwidth and (nonnegative) gain of the i-th link, respectively. Hence, the corresponding one-way transmission delay equates: $D(i) = \sum_{j=1}^{Q} F_j(i) t_j(i) / R(i)$, so that the corresponding one-way communication energy $\varepsilon^{net}(i)$ is:

$$\varepsilon^{net}(i) \triangleq P^{net}(i) \left(\sum_{j=1}^{Q} \frac{F_j(i) t_j(i)}{R(i)} \right) \, [Joule]. \tag{4}$$

Table 1: Main taxonomy of the paper.

Symbol	Meaning/Role
$F_j(i)$ $[bit/s]$	j-th processing rate of $VM(i)$
L_{tot} $[bit]$	Job size
$R(i)$ $[bit/s]$	Communication rate of the i-th end-to-end connection
R_t $[bit/s]$	Aggregate communication rate of the Virtual LAN
$\overline{T}$ $[s]$	Per-job maximum allowed computing time
$t_j(i)$ $[s]$	Computing time of $VM(i)$ working at $F_j(i)$
$\overline{T}$ $[s]$	Per-job maximum allowed computing-plus-communication time
$P^{net}(i)$ $[Watt]$	Power consumed by the i-th end-to-end connection
$P^{idle}(i)$ $[Watt]$	Power consumed by the i-th end-to-end link connection in the idle mode
ε_{tot} $[Joule]$	Total consumed energy
ε_{CPU} $[Joule]$	Computing energy
ε_{Reconf} $[Joule]$	Reconfiguration energy
ε^{net} $[Joule]$	Communication(Network) energy
M	Maximum number of available VMs
$Q+1$	Number of discrete CPU frequencies allowed for each VM
PMR	Peak-to-Mean Ratio of the offered workload

2.4 Optimization Problem

The goal is to minimize the overall resulting communication-plus-computing energy, formally defined as:

$$\varepsilon_{tot} \triangleq \sum_{i=1}^{M} \varepsilon_{CPU}(i) + \sum_{i=1}^{M} \varepsilon_{Reconf}(i) + \sum_{i=1}^{M} \varepsilon^{net}(i) \; [Joule], \tag{5}$$

where $\varepsilon_{CPU}(i)$, $\varepsilon_{Reconf}(i)$, $\varepsilon^{net}(i)$ are the computational cost, the reconfiguration cost, and the communication cost of $VM(i)$, respectively. Furthermore, we recall the our problem is subject to the *hard* constraint $\overline{T}$ on the allowed per-job execution time Table 1 summarizes the main notations of in this paper.

3 Proposed Solution for the Optimization Problem

The proposed methodology aims to minimize the total energy consumption of incoming workload by selecting the best computing resource for job execution based on current load level and by selecting the optimal bandwidth to minimize the communication energy consumption, while considering the content-based reconfiguration frequencies for each VM. Computing resources are the collection of Physical Machines (PMs), each comprised of one or more cores,

memory, network interface and local I/O. Specifically, this functionality aims to tune properly the task sizes, the communication rates and the processing rates of the networked VMs. The goal is to minimize (on a per-job basis) the overall resulting communication-plus-computing energy, which includes the summation of $\varepsilon_{CPU}(i)$, $\varepsilon_{Reconf}(i)$ and $\varepsilon^{net}(i)$ for all M VMs in ε_{tot} which are calculated for each VM. Furthermore, the total duration takes to process each incoming workload is bounded to a hard constraint $\overline{T}$ $[s]$ which conveys the SLA considered for that workload.

For the solution of the optimization problem, we find it useful to add an additional parameter, namely T, that is a threshold for the computation operation in job processing (without considering network delays). Hence, we split the SLA into two different constraints, that are computation time less or equal than T and network-related time less or equal than $\overline{T}-T$.

The resulting Optimization Problem can thus be expressed as follows:

$$\min \sum_{i=1}^{M} \varepsilon_{CPU}(i) + \sum_{i=1}^{M} \varepsilon_{Reconf}(i) + \varepsilon^{net}(i) \tag{6.1}$$

s.t.:

$$\sum_{i=1}^{M}\sum_{j=0}^{Q} F_j(i)t_j(i) = L_{tot}, \tag{6.2}$$

$$\sum_{i=1}^{M} R(i) \leq R_t, \tag{6.3}$$

$$\sum_{j=0}^{Q} t_j(i) \leq T, \quad i=1,\ldots,M, \tag{6.4}$$

$$\sum_{j=0}^{Q} \frac{2F_j(i)t_j(i)}{R(i)} \leq \overline{T}-T, \ i=1,\ldots,M, \tag{6.5}$$

$$0 \leq t_j(i) \leq T, \ i=1,\ldots,M, \ j=0,\ldots,Q, \tag{6.6}$$

$$0 \leq R(i) \leq R_t, \ \forall i=1,\ldots,M. \tag{6.7}$$

Specifically, equation (6.1) is the objective function which consists of the sum of three terms which accounts for the computing energy, the reconfiguration energy cost is the networking energy. The decision variables of the optimization problem are the $t_j(i)$ (the time spent by each VM operating at frequency f_j and $R(i)$ is the transmission rate for VM i. Eq. (6.2) is the (global) constraint which guarantees that the overall job is decomposed into M parallel tasks. Here, L_{tot} is the size of the incoming job that needs to be distributed over M VMs for computation. Also, the product $F_j(i)t_j(i)$ is the workload processed for each discrete frequency f_j which is processed by VM i during the interval $t_j(i)$. The bandwidth inequality in (6.3) ensures that the bandwidth summation of each VM must be less than the maximum available bandwidth of the global network. Eq. (6.4) is the constraint on computation time, while Eq. (6.5) is the constraint on data exchange time (the two constraints combined express the SLA) Eq. (6.6) guarantees that the duration of each computing interval is no negative and less than T. Finally, the last constraint in (6.7) ensures that our control parameter R (communicate rate of the channel) is positive and lower than the maximum network capacity.

The third term of the optimization problem is non-convex but the rest of the constraints are affine or convex in their considered range and in closed-form. However the global problem can still be turned into an equivalent (possibly, feasible) convex problem, as pointed out by the following **Proposition 1**.

Proposition 1. ε^{net} can be put in the following form

$$\begin{aligned}\sum_{i=1}^{M}\sum_{j=0}^{Q} 2P^{net}(i)\left(\frac{F_j(i)t_j(i)}{R(i)}\right) = \\ (\overline{T}-T)\sum_{i=1}^{M}\sum_{j=0}^{Q} P^{net}(i)\left(\frac{2F_j(i)t_j(i)}{\overline{T}-T}\right).\end{aligned} \tag{7}$$

Proof: Let $R(i)^*$ be the optimal solution of the eq. (6.1), and let

$$\begin{aligned}\mathcal{C} \triangleq \Bigg\{ \left(\overrightarrow{F_j(i)t_j(i)}\right) \in (\mathbb{R}_0^+)^M : \\ \left(\sum_{j=0}^{Q} F_j(i)t_j(i)/R(i)^* \left(\overrightarrow{F_j(i)t_j(i)}\right)\right) \leq \\ (\overline{T}-T)/2, i=\{1,\ldots,M\}, j=\{0,\ldots,Q\}; \\ \sum_{i=1}^{M}\sum_{j=0}^{Q} R(i)^* \left(\overrightarrow{F_j(i)t_j(i)}\right) \leq R_t \Bigg\},\end{aligned} \tag{8}$$

be the region of nonnegative M-dimensional Euclidean space constituted by all $\overrightarrow{F_j(i)t_j(i)}$ vectors meeting the constraints in (6.4) and (6.5). For feasibility and solution of (6.1) we have

i) The communication term in (6.1) is feasible *if and only if* the vector $\overrightarrow{F_j(i)t_j(i)}$ meets the following condition:

$$\sum_{i=1}^{M}\sum_{j=0}^{Q} F_j(i)t_j(i) \leq R_t(\overline{T}-T)/2 \tag{9}$$

ii) The solution of the communication term in eq. (6.1) is given by the following closed-form ex-

pression:

$$R(i)^* \left(\overrightarrow{F_j(i)t_j(i)} \right) \equiv R(i)^* \left(\sum_{j=0}^{Q} F_j(i)t_j(i) \right) \equiv \left(\sum_{j=0}^{Q} 2F_j(i)t_j(i)/(\overline{T}-T) \right), i = 1,\ldots,M. \tag{10}$$

For any assigned $\overrightarrow{F_j(i)t_j(i)}$, the objective function in (6.1) is the summation of $M(Q+1)$ nonnegative terms, where the ij-th term depends only on $R(i)$ for all j. Thus, being the objective function in (6.1) separable and its minimization may be carried out component-wise. Since the ij-th term in (6.1) is increasing in $R(i)$ and the constraints in (6.4) and (6.5) must be met, the ij-th minimum is attained when the constraints in (6.4) and (6.5) are binding, and this proves the validity of (9). Finally, the set of rates in (10) is feasible for the communication cost *if and only if* the constraint in (6.5) is met, and this proves the validity of the feasibility condition in (10).

Moreover, the end-to-end links power cost $\sum_{j=0}^{Q} 2P^{net}(i)\,(F_j(i)t_j(i)/R(i))$ is the product of the end-to-end link formula which is based on Shannon-Hartley in (3) and is continuous, nonnegative and nondecreasing for $R(i) > 0$, $\forall i \in \{1,\ldots,M\}$, with the multi-variable coefficient which can be feasible if only the following equation holds (we use "$\rightarrow$" which means *implies*):

$$\sum_{j=0}^{Q} \frac{2F_j(i)t_j(i)}{R(i)} \leq \overline{T}-T \rightarrow \left(\sum_{j=0}^{Q} \frac{F_j(i)t_j(i)}{R(i)} \right) \leq \frac{(\overline{T}-T)}{2}. \tag{11}$$

Equation (11) is obtained by manipulating equation (6.4). To make the optimization problem easier to solve, we recast the second control variable by rewriting $R(i)$ based on another control variable ($t_j(i)$) as follows:

$$\sum_{j=0}^{Q} \frac{2F_j(i)t_j(i)}{R(i)} \leq \overline{T}-T \rightarrow R(i) \geq \sum_{j=0}^{Q} \left(\frac{2F_j(i)t_j(i)}{\overline{T}-T} \right). \tag{12}$$

Applying the result of equations (11) and (12) in the third term of the objective function, we have that the end-to-end link function ε^{net} which is based on two control variables $\mathcal{G}(R(i);t_j(i))$ can be re-written by changing the second control variable $R(i)$ to a function of other control variable $t_j(i)$ in eq. (13):

$$\varepsilon^{net}(i) = \mathcal{G}\left(R(i);t_j(i)\right) \triangleq \mathcal{H}(t_j(i)). \tag{13}$$

The new formula for energy-aware communication end-to-end link just depends on the summation of time variables for each VM and the main function ($\mathcal{H}(.)$) can be written according to the equation (7). Thus, this proves the third term in (6.1) is *convex*. □

4 Performance Comparisons

This section evaluates the simulated performance of the proposed scheduler for different scenarios and compares it with the IDEAL no-DVFS techniques presented in (Mathew et al., 2012), the Standard (or Real) available DVFS-enabled technique (currently, one of the methods being used in the DVFS-enabled data centers) (Kimura et al., 2006), the Lyapunov method in (Urgaonkar et al., 2010) and the *NetDC* approach (Cordeschi et al., 2013). It is worth to note that the proposed approach can be applied in real data centers, differently from NetDC that relies on calculated fractions of real frequency, which cannot be applied in real environments. Hence, we emphasize that the considered *NetDC* (Cordeschi et al., 2013) and IDEAL no-DVFS techniques (Mathew et al., 2012) work with the continue ranges of frequencies, which is unrealistic and not feasible in real scenarios, while the proposed scheduler could be one of the best viable solutions in networked data centers.

4.1 Testbed Setup

The simulation is done by using the CVX solver over Matlab (Grant and Boyd, 2015). We consider three different scenarios: two synthetic workloads, both including multiple VMs and detailed in Table 2 and Table 3, and a real-world workload trace. The main differences between the two synthetic scenarios are the corresponding CPU discrete frequencies and the incoming workload. In order to account for the effects of the reconfiguration costs and the time-fluctuations of the offered workload on the energy performance of the simulated schedulers, we model the offered jobs as an independent identically distributed (i.i.d.) random sequence L_{tot}, whose samples are uniformly distributed over the interval $[\overline{L}_{tot}-a, \overline{L}_{tot}+a]$, with $\overline{L}_{tot} \equiv 8$ $[Gbit]$ with $a = 2$ $[Gbit]$ and $\overline{L}_{tot} = \{8, 70\}$ (i.e., $PMR = 1.25$) (as in (Cordeschi et al., 2014)). Furthermore, we pose $a = 2$ $[Gbit]$ with $PMR = 1.1428$ and $\overline{L}_{tot} = 8$ in scenario 1 and $a = 10$ $[Gbit]$ with $PMR = 1.25$ and $\overline{L}_{tot} = 70$ in scenario 2. The discrete frequency for the first scenario are taken from Intel Nehalem Quad-core Processor (Kimura et al., 2006) called $F1 = \{0.15, 1.867, 2.133, 2.533, 2.668\}$. The second scenario is based on a power-scalable real Crusoe cluster with TM-5800 CPU in (Almeida et al., 2010), e.g., $F2 = \{0.300, 0.533, 0.667, 0.800, 0.933\}$.

Table 2: Default values of the main system parameters for the first test scenario.

Parameter	Value
PE=M	$[1,\ldots,10]$
T_t	7 $[s]$
T	5 $[s]$
R_t	100 $[Gbit/s]$
C_{eff}	1 $[\mu F]$
k_e	0.05 $[Joule/(GHz)^2]$
F	$F1$ $[GHz]$
Q	5
A	100%
$P^{idle}(i)$	0.5 $[Watt]$
ζ_i	0.5 $[mWatt]$
f_i^{max}	2.668 $[GHz]$

Table 3: Default values of the main system parameters for the second test scenario.

Parameter	Value
k_e	0.005 $[Joule/(GHz)^2]$
Q	5
F	$F2$ $[GHz]$
$\overline{L}_{tot}$	70 $[Mbit]$
M	$\{20,30,40\}$
f_i^{max}	0.933 $[GHz]$

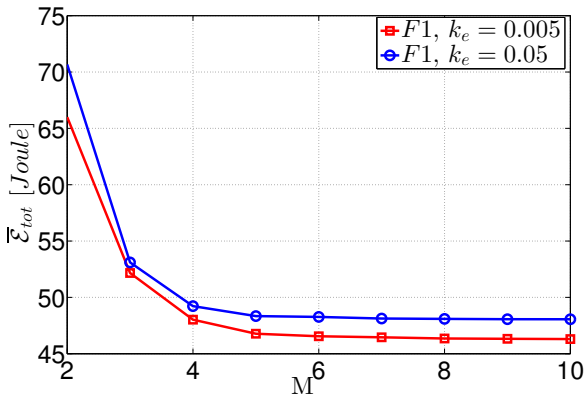
Each simulated point has been numerically evaluated by averaging over 1000 independent runs.

4.2 Simulation Results

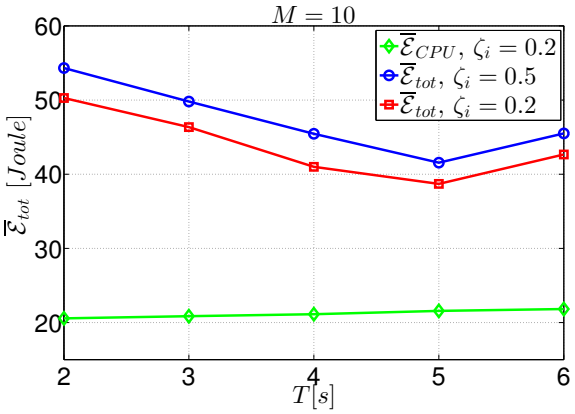
We evaluate the proposed scheduler with the aforementioned scenarios as follows.

4.2.1 First Scenario

In the first scenario, which is based on Table 2 parameters, we evaluate the average (per-job) energy consumed by the system for a varying number M of available VMs, and the variation of k_e related to the processing rate $F1$, as shown in Fig. 3a. Based on the synthetic traces of the workload in the first scenario, the comparison in Fig. 3a confirms that by increasing the VMs the consumed energy decreases, with a reduction ranging from 80% (case of $k_e = 0.005$ with the lower plot) to 85% (case of $k_e = 0.05$ with the upper plot). These results confirm the expectations (Baliga et al., 2011) that noticeable energy savings may be achieved by jointly changing the available computing-plus-communication resources. Fig. 3b tests the computing-vs-communication energy trade-off for the first scenario of Table 2 for different values of T. It confirms that small T's values lead



(a) Impact of $\overline{\varepsilon}_{tot}$ by reconfiguration parameter



(b) Computing-vs.- Communication tradeoff

Figure 3: $\overline{\varepsilon}_{tot}$ of the proposed method in the first test scenario, for various k_e (Fig. 3a) and for various T and ζ (Fig. 3b).

to higher per-VM processing rate which leads to increasing the ε_{tot}. While T increases, the proposed scheduler exploits the processing time in order to decrease ε_{tot} (i.e., reduction close to 20% in the two upper curves of Fig. 3b). On the other hand, extremely large T values induce high end-to-end communication rates, which lead to increasing the ε_{tot} (see eq. (6.5)).

We perform another experiment in order to evaluate the energy reduction due to scaling up/down of the computing, reconfiguration and communication rates when the number of VMs increases (i.e., we process the results for the one time implementation over 10 VMs). In detail, Fig. 4 presents the average over 1000 offered jobs of the total energy ε_{tot}, Computation energy ε_{CPU}, reconfiguration energy ε_{Reconf}, and communication energy ε^{net} for our approach, IDEAL, Standard, Lyapunov-based method in (Urgaonkar et al., 2010) and a recent work done in this area in (Cordeschi et al., 2013), in subfigures 4a, 4b, 4c, 4d, respectively. Specifically, Fig. 4a points out that the average total cost for all ap-

proaches decreases by increasing the number of VMs because a lower fraction of L_{tot} is assigned to each VM (i.e., $F_j(i)t_j(i)$), and the needed frequency to process the data within the allowed time decreases, with a major gain in terms of energy. In Fig. 4a, the average energy-saving of the proposed method is approximately 50% and 60% compared to Lyapunov-based and Standard schedulers, respectively. Furthermore, in Fig. 4b we observe that an increase of the number of VMs over $M = 2$ has a reduced effect on the cost of the computing energy. This is due to the system being able to manage the running time for each active discrete frequency even while M is low ($M < 4$): with an increased number of VMs, the system goes to the Idle mode or F_0 for most of the time and less or no time is assigned to the remaining frequencies. Fig. 4c shows the reconfiguration costs. We recall that our approach considers two costs (internal-switching and external-switching) for each $VM(i)$, thus resulting in increased reconfiguration costs compared to NetDC (Cordeschi et al., 2013) and Lyapunov-based scheduler in (Urgaonkar et al., 2010), which consider just probabilities of previous and next active discrete frequencies for each $VM(i)$ (i.e., external-cost of our approach). Lastly, Fig. 4d points out that the communication cost of the proposed technique is lower with respect to the other alternatives and close to IDEAL: according to the third term of the (6.1), the optimization problem tries to find the optimum objective variables when more resources are available. Fig. 4d shows that the proposed scheduler is about 10%, 50%, 65% better than NetDC (Cordeschi et al., 2013), Lyapunov (Urgaonkar et al., 2010), and Standard (Kimura et al., 2006) schedulers, respectively. Indeed, the proposed scheduler is able to find proper running times of the active discrete frequencies for each offered job (it means that $\sum F_j(i)t_j(i)$ is the same for all approaches and equal to L_{tot}).

Figure 5 reports the average execution time (AET) per each job for the first 100 offered jobs with $M = 2$ and $M = 10$ in the first scenario. Specifically, while the number of jobs increases, the AET per-job decreases significantly after some slots: this is due to the proposed scheduler being able to adapt itself to the incoming traffic using optimization technique (see (6.1)), with a consequent reduction in the AET per job.

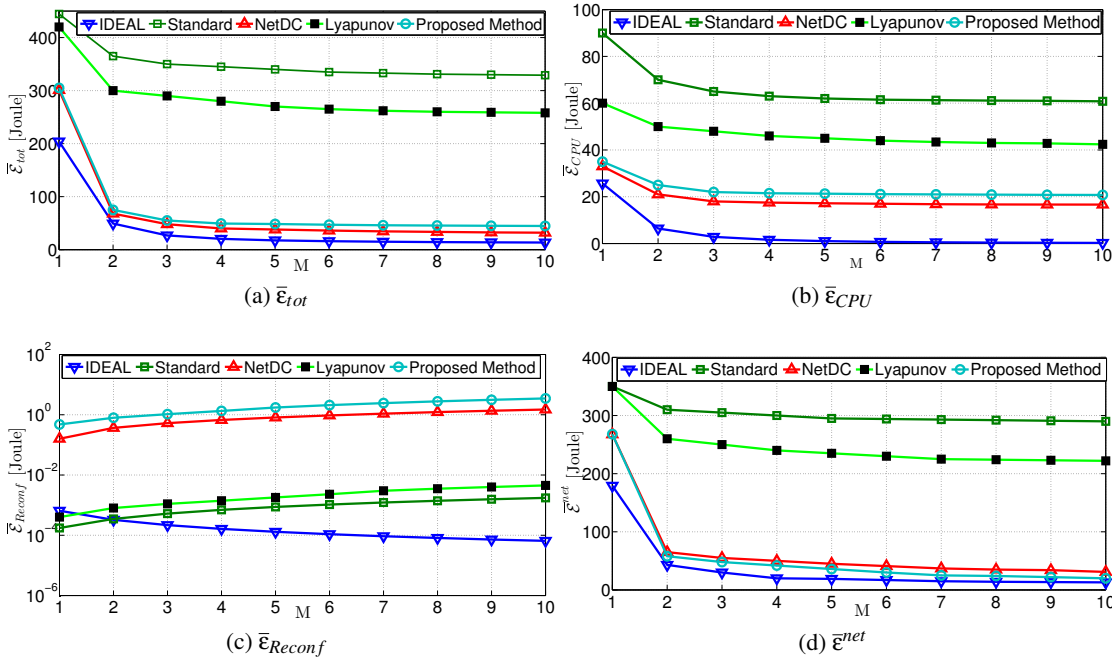
4.2.2 Second Scenario

In the second scenario we evaluate the energy consumption of the proposed scheduler for a high amount of jobs and VMs. Figs. 6 and 7 present the total average consumed energy for 20, 30, and 40 VMs and high volume of incoming jobs. In Fig. 6, we show the results of a sensitivity analysis carried out with respect to the parameters T (maximum computing time), R_t (maximum network data transfer rate) and the communication coefficient ζ in order to evaluate the energy consumption of the proposed method while facing various SLA ranges. Fig. 6a shows that, by fixing $T = 5$ and $\zeta = 0.5$ and increasing the R_t data center communication boundary by a factor of 10, the proposed scheduler saves more energy (approximately 15% with a high number of VMs). This confirms that the scheduler can save energy depending on the assigned communication boundary. Then, we repeat the experiment considering a fixed value of $R_t = 100$ and varying the range for T and ζ. The results shown in Fig. 6b confirm that the best value for T is 5. Moreover, we observe that the energy consumption is less sensitive to the choice of the communication coefficient ζ; anyway, the energy costs is lower for smaller values of ζ (i.e. $\zeta = 0.2$).

Observing Fig. 7, we note that by increasing the number of VMs (system with high resources), the energy consumption significantly decreases even increasing the job volumes. In detail, the energy reduction of proposed method compared to Standard (Kimura et al., 2006) and Lyapunov (Urgaonkar et al., 2010) is about 20% and 15%, respectively, and this saving increases for an increasing number of VMs. Moreover, it is interesting to note that the gap between the proposed method and the NetDC and IDEAL decreases by increasing the VMs: note that the NetDC (Cordeschi et al., 2013) and IDEAL (Mathew et al., 2012) schedulers works with continues CPU frequencies speed that cannot be applied in a real environment due to CPU hardware limitations. We can conclude that our approach works properly even with high number of VMs.

4.3 Performance comparisons under real-world workload traces

The previous conclusions are confirmed by the numerical results of this subsection, that refers to a *real-world* workload trace represented in Fig. 8: this is the same real-world workload trace considered in (Urgaonkar et al., 2007). We perform preliminary experiments and we found that the best parameter values for this workload are $k_e = 0.5$ $[Joule/(MHz)^2]$ and $T = 1.2$ $[s]$. Furthermore, in order to maintain the (numerically evaluated) PMR of the workload trace of Fig. 8 at 1.526, we assume that each job has a mean length of 0.533 $[Mbit]$, so that at each slot the input workload has an intensity of 16 $[Mbit/slot]$. It is worth that this workload scenario is characterized by a higher variance with respect to the previous one



(a) $\overline{\varepsilon}_{tot}$ (b) $\overline{\varepsilon}_{CPU}$

(c) $\overline{\varepsilon}_{Reconf}$ (d) $\overline{\varepsilon}^{net}$

Figure 4: Comparison of average energy terms of eq. (6.1) with PMR=1.25 for the considered approaches in the first scenario.

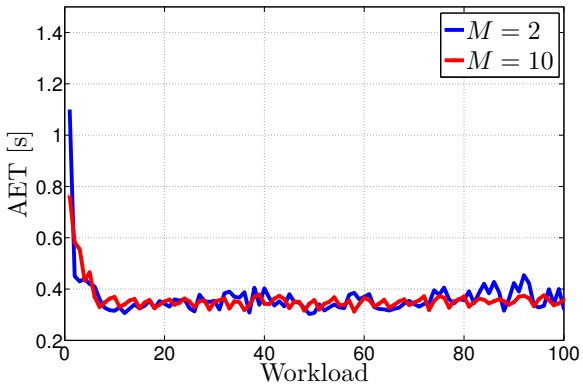
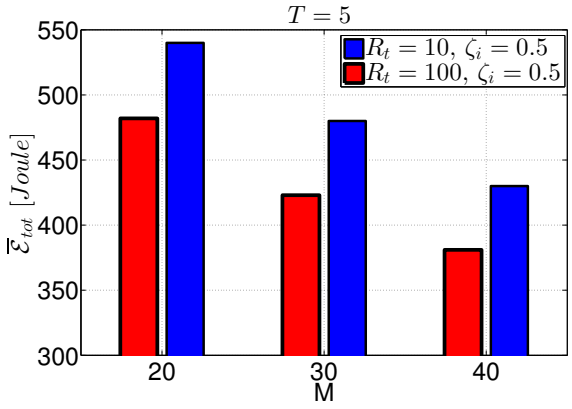
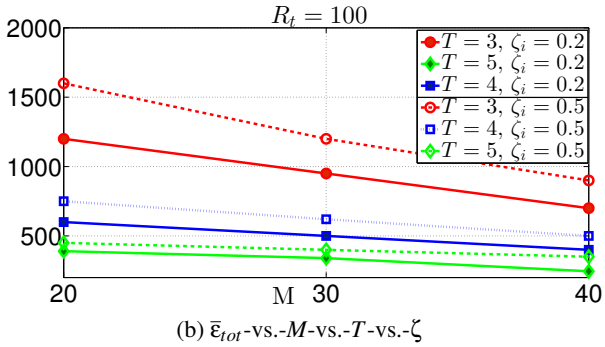


Figure 5: Average execution time (AET) per-job for the first 100 jobs.

(as testified by the higher PMR). However, even in this case the average energy reduction of the proposed scheduler, of NetDC and Lyapunov schedulers with respect to the Standard alternative is 82%, 85%, and 19%, respectively. In particular, the corresponding average energy saving of the proposed scheduler compared to the Lyapunov alternative is 76%; moreover, its gap over the IDEAL scheduler remains limited to 30%.



(a) $\overline{\varepsilon}_{tot}$-vs.-$M$-vs.-$R_t$



(b) $\overline{\varepsilon}_{tot}$-vs.-$M$-vs.-$T$-vs.-$\zeta$

Figure 6: $\overline{\varepsilon}_{tot}$ of the proposed method for various R_t (Fig. 6a) and for various T and ζ (Fig. 6b).

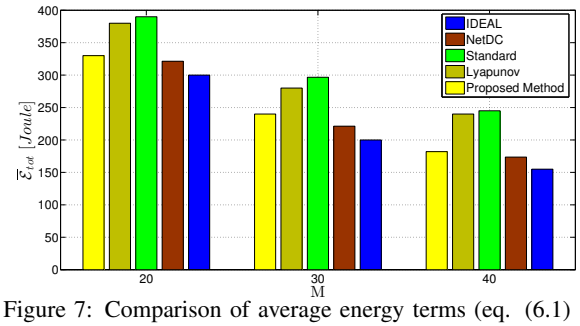


Figure 7: Comparison of average energy terms (eq. (6.1) with PMR=1.25) for the second scenario.

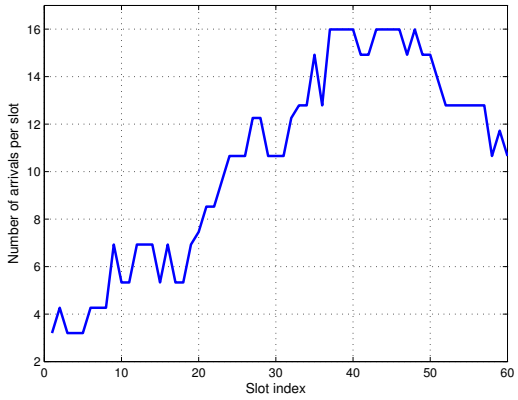


Figure 8: Measured workload trace: PMR = 1.526.

5 Conclusion and future research directions

The goal of this paper is to provide an adaptive and online energy-aware resource provisioning and scheduling of VMs in DVFS-enabled networked data centers. Also, it is aimed at summarizing key techniques and mathematical policies that minimize the data center energy consumption, which is split into three sub-problems subject to total computing and communication time's constraints, while meeting given SLAs. In the process, we identified the sources of energy consumptions in data centers and presented a high-level solution to the related sub-problems. The numerical results highlight that the proposed approach can guarantee significant average energy savings over the Standard and Lyapunov alternatives. Our proposed scheduler can manage not only the online workloads, but also the inter-switching costs among the active discrete frequencies for each VM. An interesting achievement is that, when communication costs are considered, our method is able to approach the IDEAL algorithm significantly faster than Lyapunov, Standard and NetDC models, respectively. Under soft latency constraints, the energy efficiency of the DVFS based systems could be, in principle, improved by allowing multiple jobs to be temporarily queued at the middleware layer of the cloud systems. This paper is just a first effort in a new line of research. Future extensions of the present work, currently left as open issues, include: management of the admission control using split workload estimation, improved data center model that considers more than one VM per physical server, and introduction of economic aspects (such as variable VMs cost) in the optimization problem.

Acknowledgement

The first three authors acknowledge the support of the University of Modena and Reggio Emilia through the project *SAMMClouds: Secure and Adaptive Management of Multi-Clouds*.

REFERENCES

Almeida, J., Almeida, V., Ardagna, D., Cunha, Í., Francalanci, C., and Trubian, M. (2010). Joint admission control and resource allocation in virtualized servers. *Journal of Parallel and Distributed Computing*, 70(4):344–362.

Azodolmolky, S., Wieder, P., and Yahyapour, R. (2013). Cloud computing networking: challenges and opportunities for innovations. *Communications Magazine, IEEE*, 51(7):54–62.

Baliga, J., Ayre, R. W., Hinton, K., and Tucker, R. (2011). Green cloud computing: Balancing energy in processing, storage, and transport. *Proceedings of the IEEE*, 99(1):149–167.

Canali, C. and Lancellotti, R. (2014). Exploiting ensemble techniques for automatic virtual machine clustering in cloud systems. *Automated Software Engineering*, 21(3):319–344.

Canali, C. and Lancellotti, R. (2016). Parameter Tuning for Scalable Multi-Resource Server Consolidation in Cloud Systems. *Communications Software and Systems*, 11(4):172 – 180.

Chase, J. S., Anderson, D. C., Thakar, P. N., Vahdat, A. M., and Doyle, R. P. (2001). Managing energy and server resources in hosting centers. *ACM SIGOPS Operating Systems Review*, 35(5):103–116.

Cordeschi, N., Shojafar, M., Amendola, D., and Baccarelli, E. (2014). Energy-efficient adaptive networked datacenters for the qos support of real-

time applications. *The Journal of Supercomputing*, 71(2):448–478.

Cordeschi, N., Shojafar, M., and Baccarelli, E. (2013). Energy-saving self-configuring networked data centers. *Computer Networks*, 57(17):3479–3491.

Cugola, G. and Margara, A. (2012). Processing flows of information: From data stream to complex event processing. *ACM Computing Surveys (CSUR)*, 44(3):15.

Gmach, D., Rolia, J., and Cherkasova, L. (2012). Selling t-shirts and time shares in the cloud. In *Proc. of 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2012, Ottawa, Canada, May 13-16, 2012*, pages 539–546.

Grant, M. and Boyd, S. (2015). Cvx: Matlab software for disciplined convex programming.

Herbert, S. and Marculescu, D. (2007). Analysis of dynamic voltage/frequency scaling in chip-multiprocessors. In *ISLPED*, pages 38–43. ACM/IEEE.

Kimura, H., Sato, M., Hotta, Y., Boku, T., and Takahashi, D. (2006). Emprical study on reducing energy of parallel programs using slack reclamation by dvfs in a power-scalable high performance cluster. In *IEEE CLUSTER'06*, pages 1–10. IEEE.

Mathew, V., Sitaraman, R. K., and Shenoy, P. (2012). Energy-aware load balancing in content delivery networks. In *INFOCOM, 2012 Proceedings IEEE*, pages 954–962. IEEE.

Mishra, A., Jain, R., and Durresi, A. (2012). Cloud computing: networking and communication challenges. *Communications Magazine, IEEE*, 50(9):24–25.

Qian, Z., He, Y., Su, C., Wu, Z., Zhu, H., Zhang, T., Zhou, L., Yu, Y., and Zhang, Z. (2013). Timestream: Reliable stream computation in the cloud. In *Proceedings of the 8th ACM European Conference on Computer Systems*, pages 1–14. ACM.

Shojafar, M., Cordeschi, N., Amendola, D., and Baccarelli, E. (2015). Energy-saving adaptive computing and traffic engineering for real-time-service data centers. In *Communication Workshop (ICCW), 2015 IEEE International Conference on*, pages 1800–1806. IEEE.

Urgaonkar, B., Pacifici, G., Shenoy, P., Spreitzer, M., and Tantawi, A. (2007). Analytic modeling of multitier internet applications. *ACM Transactions on the Web (TWEB)*, 1(1):2.

Urgaonkar, R., Kozat, U. C., Igarashi, K., and Neely, M. J. (2010). Dynamic resource allocation and power management in virtualized data centers. In *NOMS*, pages 479–486. IEEE.

Wang, L., Zhang, F., Arjona Aroca, J., Vasilakos, A. V., Zheng, K., Hou, C., Li, D., and Liu, Z. (2014). Greendcn: a general framework for achieving energy efficiency in data center networks. *Selected Areas in Communications, IEEE Journal on*, 32(1):4–15.

Warneke, D. and Kao, O. (2011). Exploiting dynamic resource allocation for efficient parallel data processing in the cloud. *Parallel and Distributed Systems, IEEE Transactions on*, 22(6):985–997.