A survey on techniques for improving the energy efficiency of large-scale distributed systems

ANNE-CECILE ORGERIE
CNRS, IRISA Laboratory, France
MARCOS DIAS DE ASSUNCAO
IBM, Brazil
LAURENT LEFEVRE
INRIA, LIP Laboratory, University of Lyon, France

The concerns about the rising energy consumption of data centers and the need to improve their efficiency have been shared by a large part of the research community and IT industry. As a result, a myriad of techniques have been proposed to curb the energy consumed by information and telecommunications technology. This article surveys techniques and solutions proposed to improve the energy efficiency of key components of data centers, namely computing and network resources. Existing techniques to evaluate and model the energy consumed by these resources are discussed. The paper also describes hardware and software approaches for reducing the energy consumption of entire distributed systems. The main goal of the survey is to review the state of the art and foster research on schemes to make network and computing resources more energy efficient.

Categories and Subject Descriptors: A.1 [**Introductory and Survey**]: General; C.2.0 [**Computer-Communication Networks**]: General; H.3.4 [**Information Systems**]: Systems and Software—*Distributed Systems*

General Terms: Survey, distributed systems, power, communication

Additional Key Words and Phrases: Energy efficiency, distributed systems, computing, networking

1. INTRODUCTION

Although reducing the energy consumption has always been key for sensor networks and battery-constrained devices, it has only recently become a major concern for other ICT systems. Research efforts in distributed computing in general have mostly been driven by performance, but ignoring the implications in energy consumption that some improvements can cause is no longer possible. Nowadays, the price of the energy that a server consumes during its lifetime often exceeds its purchase cost [Barroso 2005]. As a consequence, data centers are reaching power delivery limits, forcing new infrastructures to be set up near power stations to avoid transmission losses [Greenpeace 2011].

In general, the energy that a system consumes comprises two components:

—A fixed (or static) part that depends on the system's size and on the type of used components (computing, data storage and network elements); it is due to leakage currents present in any powered system.

—A variable (or dynamic) part that results from the usage of computing, storage, and network resources; it is caused by system activity and changes in the clock rates.

The energy consumption E of an equipment depends on its power consumption P over time t. Hence, for a given time interval T, the energy is given by:

$$E(T) = \int_0^T P(t)\ dt \tag{1}$$

In a large distributed system such as a Grid or a data center, various computing and networking equipments – such as PCs, switches, routers, and servers – often waste a great amount of energy because they typically remain powered on even when they are idle. For instance, in 2010 the Green Grid consortium carried out a survey about unused servers in 188 data centers, mostly located in the United States [Green Grid 2010], they estimated that on average 10% of the servers are never used, hence wasting energy; an amount that could be saved if the unused servers were switched off or put into low consumption modes. Due to scaling effects, even wastage at the component level can have global consequences on the energy efficiency of an entire distributed system.

Reducing the energy consumption of large-scale distributed systems is a challenging issue that should be addressed at different levels: infrastructure, nodes, and hardware components. This article surveys techniques and solutions proposed to improve the energy efficiency of two key types of components in distributed systems, namely computing and (wired) network resources. The main goal of this work is to review the state of the art and foster research on schemes to make computing and network resources more efficient. Section 2 discusses the different approaches for saving energy in computing resources. Section 3 classifies the research work on energy savings for wired networks. Finally, Section 4 presents concluding remarks.

2. ENERGY-EFFICIENCY OF COMPUTING RESOURCES

Data centers and supercomputers comprise large numbers of servers concentrated in a small area. According to the 37th edition of the Top500 list[1] released in June 2011, the *K Computer* located at the RIKEN Advanced Institute for Computational Science (AICS) in Japan, is the fastest supercomputer in the world (8.2 PFLOPS). The *K Computer* consumes the energy to power nearly 10,000 homes and costs around $10 million a year to operate [Geller 2011]. Such power requirements and costs are generally limiting factors of the scalability of supercomputers and data centers.

The development of supercomputers has long been driven by performance as attested by the top500 list. The green500 list has been a means to reorganize the top500 ranking by considering the energy efficiency of supercomputers, trying to raise awareness about their large power requirements [Feng and Scogland 2009].

Energy can be saved at different levels in a large computing infrastructure. However, the first and difficult step towards saving energy is to discover how much power individual components consume; a mandatory task to design and evaluate new energy-efficient architectures and algorithms.

[1] `http://www.top500.org`

2.1 Measuring and modeling the energy consumption of computing resources : from energy sensors to energy models

The energy consumption of computing resources can either be determined by energy sensors (like wattmeters) or be estimated by energy models. Wattmeters can be completely external equipment or components integrated into Power Distribution Units (PDUs) and temperature sensors. Regarding power measurements, energy sensors integrated into components offer a smaller measurement granularity than external wattmeters. Deploying energy sensors or wattmeters can be costly if it is not done when the whole infrastructure (i.e. cluster or data center) is set up.

Using energy models to estimate the consumption of components or an entire infrastructure is less expensive than deploying wattmeters. However, the models should be lightweight not to interfere with the energy consumption they try to estimate. Models can estimate the energy consumption of racks, devices, processes, services, etc. For example, *PowerTOP*[2] is a Linux software utility developed by Intel and whose goal is to "find the software component(s) that make a laptop use more power than necessary while it is idle". To estimate the power usage, PowerTOP uses Advanced Configuration and Power Interface (ACPI)[3].

As processors are among the most consuming components of computers, several solutions have been proposed to evaluate their consumption at different levels [Castagnetti et al. 2010], including:

- Cycle level estimation where the power consumption of each processor unit is estimated at each clock cycle.
- Instruction level power analysis in which the power consumption of processor instructions are summed to estimate the energy a program consumes.
- Power analysis at the functional level based on analyzing the processor architecture.
- System level power estimation considering the average power of an instruction multiplied by an application's execution time to obtain its energy consumption.

Fan *et al.* [Fan et al. 2007] model the energy consumption according to a CPU's activity, whereas another approach consists in deducing the consumption by using *event-monitoring counters* included in modern processors, starting with Pentium 4 [Merkel and Bellosa 2006].The main issue with such techniques lies in limiting the overhead incurred by the method.

The power P consumed by a processor can be expressed mathematically as the sum of the static power P_{static} and the dynamic power $P_{dynamic}$ [Wang et al. 2010]. $P_{dynamic}$ can be presented as follows:

$$P_{dynamic} = ACV^2f$$

where A is the percentage of active gates, C is the total capacitance load, V the supply voltage and f the frequency [Ge et al. 2005].

[2] PowerTOP `http://www.linuxpowertop.org/powertop.php`

[3] The Advanced Configuration and Power Interface (ACPI) is a standard developed by Hewlett-Packard, Intel, Microsoft, Phoenix Technologies and Toshiba. Its goals are to reduce a computer's power consumption by switching off its components, whereas the operating system manages the power supply of each component.

Another research issue is to estimate the power that applications consume. By providing a model to predict the power consumption and performance of the high performance LINPACK benchmarks, Subramaniam and Feng [Subramaniam and Feng 2010] concluded that the maximum energy efficiency is not always achieved at the highest performance. This leads to an interesting question: *how to measure energy-efficiency?* For infrastructures such as data centers, the most common metric is the *Power Usage Effectiveness* (PUE), introduced by the *Green Grid*[4] and defined as:

$$PUE = \frac{Total\ Facility\ Power}{IT\ Equipment\ Power}$$

Another common metric is the *Data Center Infrastructure Efficiency* (DCiE) [Green Grid 2007] expressed as:

$$DCiE = \frac{1}{PUE} = \frac{IT\ Equipment\ Power}{Total\ Facility\ Power} \times 100\%$$

These two metrics indicate how much power is used by the IT infrastructure and hence, how efficient are the cooling system and other non-IT resources. Other metrics are available, such as the *performance per watt*, used to produce the *Green500* list [Feng and Scogland 2009] and expressed in Floating Point Operations Per Second (FLOPS) per watt.

To evaluate the energy efficiency of an application or middleware, it is interesting to use as criteria the QoS metrics related to the evaluated software. For task scheduling in a Cloud environment, Yeo and Lee [Yeo and Lee 2011] utilize the energy-delay product, where they show that computing nodes that are at most three times slower than the fastest node should be discarded from the Cloud system to achieve an optimal energy-delay product.

Another solution is to use benchmarks to compare the efficiency of various architectures and software. Examples include the *SPECpower*, a benchmark that evaluates the power and performance characteristics of volume server class and multi-node class computers; and JouleSort, a benchmark to evaluate the trade-off between power and performance of computing nodes by sorting a fix number of records using as little energy as possible [Rivoire et al. 2007]. Once the energy consumption of individual computing resources is known, researchers can design and implement new techniques to reduce the energy consumed by the overall infrastructure. In a Grid or Cloud infrastructure, different solutions can be applied to both the node and the infrastructure levels. The following sections discuss research on these two levels.

2.2 Node optimizations

This section analyzes solutions that work at a node level and that can lead to large energy savings at the distributed-system level.

A computing node comprises several components, each of which can be optimized to save energy as shown in Figure 1. This figure also shows what components can

[4] The Green Grid is a non-profit, open industry consortium of end-users, policy-makers, technology providers, facility architects, and utility companies collaborating to improve the resource efficiency of data centers and business computing ecosystems.

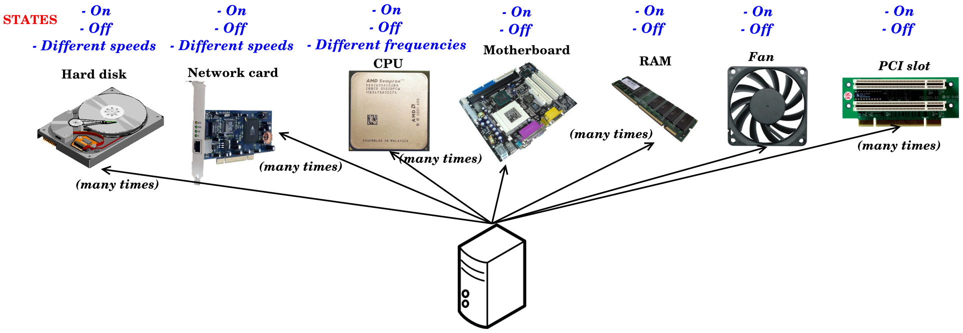

Fig. 1. Possible states per node component.

be switched off or put in lower power-consumption modes. As discussed earlier, a node can typically be in one of several states – varying from fully-on to fully-off – which have their own energy consumptions. For example, a CPU has an off-state and several on-states that correspond to different operating frequencies and voltages [Snowdon et al. 2005].

Several studies focus on minimizing the consumption of specific components, such as Network Interface Cards (NICs) [Gunaratne et al. 2005], hard disks [Allalouf et al. 2009] and CPUs [Dietz and Dieter 2006], with varying results. Table I shows the energy consumed by the components of a typical rack server [Fan et al. 2007].

Component	Peak power	Count	Total	Percentage
CPU	40 W	2	80 W	37.6 %
Memory	9 W	4	36 W	16.9 %
Disk	12 W	1	12 W	5.6 %
PCI slots	25 W	2	50 W	23.5 %
Motherboard	25 W	1	25 W	11.7 %
Fan	10 W	1	10 W	4.7 %
System total			213 W	

Table I. Component peak power breakdown for a typical server .

Despite some the potential benefits of turning components off, this technique has limitations. For example, the motherboard, a high consuming component, can be turned off only if the entire node can (sleep state). This section details node-level techniques to curb the energy consumption of computing resources, including switching off resources (using sleep states), configuring different voltages and frequencies for the CPU, improving the energy-efficiency of software and using energy-efficient hardware and low-level capabilities.

2.2.1 *Energy-aware hardware capabilities.* As shown in Figure 1, manufacturers can improve the energy efficiency of all server components (e.g. power supplies, fans and disks) by providing solutions that work at the hardware level. For example, HPC nodes commonly use hard disk drives – a scenario that may change as SSD become more affordable – whose energy consumption can be reduced by spinning platters down [Carrera et al. 2003]. However, spinning platters up leads to peaks in power consumption and takes time. Hence, the objective is generally to minimize disk accesses to keep platters spun down as long as possible [Carrera et al. 2003].

In addition to improving the energy-efficiency of hardware components, to get closer to proportional computing [Barroso and Holzle 2007], manufacturers could allow individual components such as PCI slots and CPU cores, to be put into sleep state when not in use (independently from the motherboard). During computing phases, network cards may not be required and could hence be turned off. Finally, another way for manufacturers to improve the energy efficiency is by increasing the number of voltages, frequencies and speeds available for each component (CPU, disk, NIC) making it able to adapt its working conditions to the load.

2.2.2 *Sleep state.* While the first and natural idea to save energy is to shut down idle nodes [Chase et al. 2001], this poses a problem: *how to wake them up when required?* Wake-On-LAN, a mechanism implemented on Ethernet cards, allows a remote user to wake up a PC by sending the latter specific packets via the network [Gunaratne et al. 2005]. Under this scheme the Ethernet card must be powered on at all times. The Intelligent Platform Management Interface (IPMI) is a standard hardware that operates independently from the operating system and allows administrators to manage a system remotely through a direct serial connection or via a LAN connection. This interface can also be used to switch nodes on and off remotely [Leangsuksun et al 2006].

Suspend to disk techniques can be used to reduce the energy consumed during wake-up and booting periods. When a node suspends to disk, all the main memory contents are saved in files on the hard disk drive, thus preserving the state of the operating system (e.g. open applications and documents). All of the node's components are turned off and, at the next state switch, the node will load the hibernate file, restoring the previous state. However, the sleep state is useless for nodes always used or which should always remain on for availability constraints. Other energy-efficient techniques must be used in cases where nodes are constantly switched on.

2.2.3 *Dynamic Voltage and Frequency Scaling.* For a few years, laptop processors have been able to adjust their working frequency and power consumption to conserve battery life. This technology, called Dynamic Voltage and Frequency Scaling (DVFS) [Snowdon et al. 2005], is available on recent High Performance Computing (HPC) nodes and servers present in data centers, Grids and Clouds, but is seldom exploited. Under DVFS, *P-states* (performance states) define the different frequencies supported by a processor. The several P-states – P0, P1, P2. . . Pn where n is processor dependent – enable power savings. For example, under P3 the processor will run slower and use less power than under P1. On Linux, the *CPUfreq* infrastructure allows the control of P-states by *governors* that decide which available frequency, between minimal and maximum, must be chosen. The available governors include the *on-demand*, which adjusts the frequency automatically; the *performance* that chooses the highest frequency; the *user-space*, which allows for setting the frequency manually; the *conservative*, which increases the frequency progressively (unlike the on-demand governor which skips to the maximal frequency when the system is in full load).

C-states correspond to CPU idle states, except for C0, which is an operational state. In C3 the processor can be turned off and must be reset to carry out instructions again, whereas C4 refers to the deeper sleep state. The higher the number, the less energy the processor consumes and the longer it takes to become active again. While keeping the processor idle for a long period can allow for power savings, it is necessary to reduce CPU wake-ups by disabling services and processes not strictly required. From the Linux kernel version 2.6.24, there is a new feature called Dynamic ticks or tick-less kernel (NO_HZ) which allows to wake up the processor only when required. This way, the processor is not woken up too often just to realize that there is no work to perform. *Powernow!* and *Cool'n'Quiet* technologies from AMD and *Speedstep* from INTEL implement the P-states. They can reduce voltage depending on the frequency and deactivate unused processor parts.

While DFVS techniques can save valuable amounts of energy, it should remain clear that reducing the CPU frequency increases the duration of computing tasks, and this phenomenon may increase the overall energy consumption if used insuitably. Moreover, only the dynamic part of the energy consumption can be reduced by DVFS, and this part can represent only a few percent contrary to the static part which is targeted by sleep state.

2.2.4 *Software improvements.* This section presents software improvements that reduce the energy consumed by computing resources. Nowadays, developers of drivers, kernel modules and distributed applications have to consider the implications of their design decisions in energy consumption. For example, waiting loops and active polling may frequently wake up the CPU, which might waste electricity. Examples of applications that wake up the CPU hundreds of times per second and sometimes unnecessarily are listed on PowerTOP's website.[5]

When concerning software, the Operating System (OS) is the first element to consider. OSs have heterogeneous power consumptions and can be optimized to consume less energy. Even different versions of the same OS do not present the same power consumption as illustrated by previous work that shows the power consumed by several Windows versions [EcoInfo 2011] and Linux kernels [LessWatts 2010]. The studies for Windows [EcoInfo 2011] and Linux [LessWatts 2010] have similar methodologies, and the consumption of a node under different versions of the considered OS presents non negligible variations depending on the used version.

Moreover, an OS can regulate its activity and energy consumption to meet thermal or energy constraints, a task that is commonly performed through the standard ACPI [Steele 1998]. The Basic Input Output System (BIOS) – the very first software called when a system boots – is stored in an (EEP)ROM on the motherboard and contains a set of basic functions to initialize hardware, run diagnostics and search for bootable devices where the OS might be available. Although manufacturers develop a BIOS for each motherboard they design, the default setup is generally used to support all the configurations (OS) and check for all the possible devices, thus wasting time and energy.

[5] http://www.linuxpowertop.org/known.php

In a similar way, initializing the Universal Serial Bus (USB) host adapter takes time. HPC nodes often have several USB interfaces, most of which are hardly used. To reduce the time to boot, the majority of BIOS setups can disable USB ports and avoid their initialization. Moreover, USB ports frequently wake up the CPU even if they are not in use. These factors also concern other server components that are generally not used in data centers (e.g. RS-232 serial ports, bluetooth modules and wireless cards) and that could be disabled.

2.3 Grid and data center power management

This section describes techniques for power management that work at the scale of a data center or grid. Some of these techniques coordinate at a wide scale the node-level schemes described in the previous section.

2.3.1 *Local or Green sources of electricity.* The first way to save energy at a data center is to locate it close to where the electricity is generated, hence minimizing transmission losses. For example, Western North Carolina, USA, attracts data centers with its low electricity prices due to abundant capacity of coal and nuclear power following the departure of the region's textile and furniture manufacturing [Greenpeace 2011]. As of writing, this region has three super-sized data centers from Google, Apple and Facebook with respective power demands of 60 to 100 MW, 100 MW and 40 MW [Greenpeace 2011].

Other companies opt for *greener* sources of energy. For example, Quincy (Washington, USA) supplies electricity to data facilities from Yahoo, Microsoft, Dell and Amazon with its low-cost hydroelectrics left behind following the shut down of the region's aluminum industry [Greenpeace 2011]. Several renewable energy sources such as wind power, solar energy, hydro-power, bio-energy, geothermal power and marine power can be considered to power up super-sized facilities.

Another approach is to use *free cooling* which consists in using outside air [Pawlish and Varde 2010] to cool infrastructures. As cooling accounts for about 33% of the power used in a data center [Greenberg et al. 2008], this technique leads companies to locate their facilities in regions and countries with cold climate, such as Sweden[6]. Another free cooling technique is to use sea water, such as in the new Google data center in Hamina, Finland[7].

In spite of these innovative approaches, numerous data facilities have already been built and cannot move. Grid environments, on the other hand, can still take advantage of multiple locations to use green sources of energy with approaches such as *follow-the-sun* and *follow-the-wind* [Figuerola et al. 2009]. As sun and wind provide renewable sources of energy whose capacity fluctuates over time, the rationale is to place computing jobs on resources using renewable energy, and migrate jobs as renewable energy becomes available on resources in other locations.

2.3.2 *Thermal management.* Thermal issues have been the first to be raised because they are the most direct consequences of increasing the number of transistors on processor chips. These issues and energy consumption are interrelated

[6] "Safe, Green and Cool" `http://www.investsweden.se/world/Industries/ICT/Data-centers/`
[7] `http://www.google.com/corporate/datacenter/efficient-computing/efficient-data-centers.html`

since decreasing the heat production will reduce energy consumption. For this reason several algorithms deal with both energy and thermal issues [Patel et al. 2002; Sharma et al. 2005; Merkel and Bellosa 2006].

An HP technical report [Patel et al. 2002] presents the typical case of an infrastructure with a PUE of 1.5, meaning that cooling by itself consumes half the amount of power used by the computing resources. The authors present a solution, called thermal load balancing [Sharma et al. 2005], that takes advantage of the different clusters' location in the Grid and assigns workload based on the thermal management infrastructure and the seasonal and diurnal variations of temperature [Patel et al. 2002]. It takes as example two sites of the same Grid, one located in New Delhi and another in Phoenix. During Summer, the external temperature in New Delhi reaches its peak at midday, at which time it is night in Phoenix where the temperature is lower. Hence, it is preferable to place the workload in Phoenix and use less cooling capacity than in New Delhi.

2.3.3 *Workload consolidation.* Workload consolidation [Chase and Doyle 2001; Chase et al. 2001], investigated in following works [Doyle et al. 2003; Urgaonkar et al. 2008; Verma et al. 2008; Kalyvianaki et al. 2009; Kusic et al. 2008; Jung et al. 2009; Srikantaiah et al. 2008], consists in running multiple tasks on the same physical machine in order to reduce the number of nodes that are switched on. A key component of these systems is to monitor and estimate the workload of the user applications or the arrival of user requests. Several techniques have been applied to estimate the load of a system, such as exponential moving averages [Box et al. 1994], Kalman filters [Kalman 1960], auto-regressive models, and combinations of methods [Kim and Noble 2001; Chase et al. 2001].

Fitted with workload-estimation techniques, Grid systems provide schemes to minimize the energy consumed by the underlying infrastructure while minimizing costs and violations of Service Level Agreements (SLAs). Chase *et al.* [Chase et al. 2001] introduced MUSE, an economy-based system that allocates resources of hosting centers to services aiming to minimize energy consumption. Services bid for resources as a function of delivered performance whilst MUSE switches unused servers off. Kalyvianaki *et al.* [Kalyvianaki et al. 2009] introduced autonomic resource provisioning using Kalman filters. Kusic *et al.* proposed a look-ahead control scheme for constantly optimizing the power efficiency of a virtualized environment [Kusic et al. 2008]. With the goal of maximizing the profit yielded by the system while minimizing the power consumption and SLA violations, the provisioning problem is modeled as a sequential optimization under uncertainty and is solved using the look-ahead control scheme.

In some cases, consolidating workload on fewer nodes may increase the overall energy consumed by the platform if unused nodes are not switched off. Freeh *et al.* [Freeh et al. 2005] show that for some parallel applications, one can save energy and time by executing a program on more nodes at a slower speed rather than on fewer nodes at the fastest speed. Similarly, sometimes using the least power consuming state of processors is more energy consuming than exploring parallelism and as many processors as possible to complete an application faster [de Langen and Juurlink 2006]. Hence, parallelism should be implemented carefully.

Parallel applications with unbalanced load can benefit from using DVFS at nodes with small tasks while they wait to synchronize with nodes with heavier load [Kappiah et al. 2005]. As the middleware can influence the energy consumed by the platform as it maps users' tasks and physical resources, it constitutes a great leverage to improve the energy efficiency of distributed systems. However, although workload consolidation techniques often rely on task scheduling algorithms, energy-efficient scheduling does not always aim at consolidating tasks on fewer nodes, as shown in the next subsection.

Moreover, for all on/off algorithms, an unnecessary wake-up wastes energy by creating spikes in power consumption when the node is woken up and put into sleep mode again. Such algorithms should thus be carefully designed in order not to shut down nodes unnecessarily [Lefèvre and Orgerie 2009].

2.3.4 *Energy-aware task scheduling.* A significant number of algorithms has been proposed for energy-aware scheduling. These algorithms differ on whether they are designed for divisible tasks [Chase et al. 2001] [Wang et al. 2010], whether they require synchronization [Jejurikar and Gupta 2006], if they use DVFS [Wang et al. 2010], or whether they work on homogeneous clusters [Mishra et al. 2003] [Yang et al. 2009]. Some complete energy efficient framework dealing with energy efficiency have been proposed (like the Green-Net framework [Da-Costa et al. 2010]). Energy aware task schedulers are generally classified into three categories [Zhuo and Chakrabarti 2008], namely: off-line scheduling based on *a priori* task information [Yao et al. 1995], on-line scheduling, which is purely dynamic [Kang and Ranka 2010; Zhuo and Chakrabarti 2008] and hybrid approaches including an offline phase where the slack is greedily absorbed and dynamic algorithms operating in the online phase [He and Jia 2008; Shin and Kim 2004].

Chase and Doyle [Chase and Doyle 2001] attempt to minimize the number of joules per operation. Their resource manager gets a set of awake nodes and minimizes its size as much as possible. When a task on a node completes, the manager tries to move the other tasks from this node to the other running nodes. If a new task arrives, the manager tries to put it on the awake nodes, while the other nodes remain off. This algorithm does not include load balancing mechanisms, so some nodes may wore out prematurely while others remain unused.

Under certain scenarios, it is also possible to negotiate the performance degradation with the user (mainly in terms of execution time) to save more energy. Such an approach is described in the work by Wang *et al.* [Wang et al. 2010] where users accept, for example, an increase of 10% in task execution time in order to reduce energy consumption.

Some task scheduling algorithms use DVFS techniques [Jejurikar and Gupta 2006; Snowdon et al. 2005; Fan et al. 2007], which allows for energy savings when the nodes are not fully utilized. DVFS can be used during the execution of non-critical tasks [Chen et al. 2005; Wang et al. 2010] or during communication phases of MPI applications [Lim et al. 2006]. In this case, the processor frequency is adjusted depending on the CPU utilization. Another solution is to use the user-perceived latency, the delay between user input and computer response, to drive voltage scaling [Yan et al. 2005].

As outlined by the variety of the proposed solutions, scheduling algorithms should be designed for the workload they have to manage (e.g. web servers, computing jobs). For periodic real-time tasks, Aydi *et al.* prove that the optimal solution consists in using the CPUs at either full capacity or at the minimum speed if the utilization is under 100% [Aydi et al. 2001]. In addition to performance goals, energy-efficient job placement algorithms can take into account load balancing [Merkel and Bellosa 2006], thermal management [Patel et al. 2002; Sharma et al. 2005] and network connections [Chen et al. 2008].

2.4 Virtualization techniques

Clouds, already used by numerous companies, differ from Grids as explained by Buyya *et al.* [Buyya et al. 2009] and can be part of the next-generation data centers with virtualized nodes and provisioning on demand. As of writing, Salesforce.com handles 54,000 companies and their 1.5 million employees using only 1,000 servers [Hamm 2009]. Different enterprises, such as IBM [Boss et al. 2007], also support and provide Cloud infrastructures and services to customer companies.

Virtualization is a key feature of Clouds as it can improve the efficiency of large-scale distributed systems [Talaber et al. 2009]. It is now widely used to provide a large number of computing resources and minimize the energy consumption of Cloud infrastructures – an issue as urgent as the energy consumption of data centers and grids [Nathuji and Schwan 2007; Stoess et al. 2007; Talaber et al. 2009].

Even if virtualization adds a software layer that consumes energy [Torres et al. 2008], it allows for finer load consolidation on a virtualized node [Srikantaiah et al. 2008] and provides live migration techniques [Travostino et al. 2006] to strengthen load aggregation. However, these techniques have a cost [Orgerie et al. 2010] and should be carefully studied in order to reduce the overall consumption of Clouds.

This section details the energy gains of virtualization techniques at the Virtual Machine (VM) level, using VM migration techniques, and at the Cloud level.

2.4.1 *Virtualization techniques.* Virtualization brought along ideas on energy management [Talaber et al. 2009; Hermenier et al. 2006]. As nodes can be virtualized and host several virtual machines, virtualization can address some of the limitations in cooling and power delivery faced by large-scale distributed systems.

The overhead posed by VM technologies [Cherkasova and Gardner 2005] has decreased over the years, which has expanded their appeal for running high performance computing applications [Tatezono et al. 2006] and turned virtualization into a mainstream technology for managing and providing resources for a wide user community with heterogeneous software-stack requirements.

While the macro-level resource management performs actions that generally take into account the power consumption of a group of resources or the whole data center, at the host-level the power management is performed by configuring parameters of the hypervisor's scheduler, such as throttling of Virtual CPUs (VCPU) and using other OS specific policies. In the proposed architectures, hosts generally run a local resource manager that is responsible for monitoring the power consumption of the host and optimizing it according to local policies. The power management capabilities available in virtualized hosts have been categorized as [Nathuji and Schwan 2007]: "soft" actions such as CPU idling and throttling; "hard" actions like DVFS; and consolidating in the hypervisor. CPU idling or soft states consist in changing resource allotments of VMs and attributes of the hypervisor's scheduler (e.g. number of credits in Xen's credit scheduler) to reduce the CPU time allocated to a VM so that it consumes less power. Hard actions comprise techniques such as scaling the voltage and frequency of CPUs. Consolidation can also be performed at the host-level where the VCPUs allocated to VMs can be configured to share CPU cores, putting unused cores in idle state, hence saving the energy that would otherwise be used by the additional core to run a VM.

Nathuji and Schwan [Nathuji and Schwan 2007] presented VirtualPower, a power management system for virtualized environments that explores both hardware power scaling and software-based methods to control the power consumption of underlying platforms. VirtualPower exports a set of power states to VM guests that allow them to use and act upon these states thereby performing their own power management policies. The soft states are intercepted by Xen hypervisor and mapped to changes in the underlying hardware such as CPU frequency scaling according to the virtual power management rules. The power management policies implemented in the guest VMs are used as "hints" by the hypervisor rather than executable commands. They also evaluate the power drawn by cores at different frequency/voltage levels and suggest that such technique be used along with soft schemes.

VM-based resource management systems such as Eucalyptus [Nurmi et al. 2008] and OpenNebula [Fontan et al. 2008], allow users to instantiate and customize clusters of virtual machines atop the underlying hardware infrastructure. When applied in a data center environment, virtualization can allow for impressive workload consolidation. For instance, as Web applications usually present variable user population and time-variant workloads, virtualization can be employed to reduce the energy consumed by the data center environment through server consolidation whereby VMs running different workloads can share the same physical host.

2.4.2 *Virtual Machines Migration.* Virtualization needs powerful resource management mechanisms [Grit et al. 2006] to benefit from migrating, pausing and resuming VMs. The design of resource-management policies is challenging (NP-hard problem) and dynamic. Live migration [Clark et al. 2005] greatly improves the capacities and the features of Cloud environments: it facilitates fault management, load balancing, and low-level system maintenance. Migration implies more flexible resource management as virtual machines can move from one host to another. It offers a new stage of virtualization by removing the concept of locality in virtualized environments.

However, this technique is complex and more difficult to use over MAN/WAN [Travostino et al. 2006] than within a cluster. IP addressing is a problem since the system should change the address of the migrated virtual machine which does not remain in the same network domain. Moreover, it impacts the performance of VMs by adding a non negligible overhead [Voorsluys et al. 2009].

2.4.3 *Cloud computing.* Current Web applications demand highly flexible hosting and resource provisioning solutions [Subramanyam et al. 2009]. The rising popularity of social network Web sites, and the desire of current Internet users to store and share increasing amounts of information (e.g. pictures, movies, life-stories, virtual farms) have required scalable infrastructure. Benefiting from economies of scale and recent developments in Web technologies, data centers have emerged as a key model to provision resources to Web applications and deal with their availability and performance requirements. However, data centers are often provisioned to handle sporadic peak loads, which can result in low resource utilization [Iosup et al. 2006] and wastage of energy [Harizopoulos et al. 2009].

A Cloud computing environment can scale dynamically by allocating virtualized resources that are often provided as services over the Internet [Hayes 2008]. Clouds open up new horizons where anything is considered as a service (infrastructure, platform, software, computing, storage) and provide advantages such as cost and reliability. However, customers commonly worry about security and loss of sensitive data when using services from Cloud providers such as Amazon. Accounting is another key challenge as providers need to be competitive and remain economically viable.

The ever-increasing demand for cloud-based services does raise the alarming concern about the energy consumed by data centers. Recent reports [Patterson et al. 2007] indicate that energy consumption is becoming dominant in the Total Cost of Ownership (TCO). In 2005, data centers represented about 1 percent of the total world electricity consumption [Koomey 2011]. For most ICT companies, this consumption has increased: for instance, between 2005 and 2010, the electricity used by Google's servers has more than double [Koomey 2011] leading to more carbon emissions. Electricity becomes the new limiting factor for deploying data center infrastructures.

A range of technologies can be utilized to make cloud computing infrastructures more energy efficient, including better cooling technologies, temperature-aware scheduling [Moore et al. 2005; Fan et al. 2007; Patel et al. 2002], Dynamic Voltage and Frequency Scaling (DVFS) [Snowdon et al. 2005], and resource virtualization [Talaber et al. 2009]. The use of VMs [Barham et al. 2003] brings several benefits including environment and performance isolation; improved resource utilization by enabling workload consolidation; and resource provisioning on demand. Nevertheless, such technologies must be analyzed and used carefully for really improving the energy-efficiency of computing infrastructures [Miyoshi et al. 2002; Orgerie et al. 2010].

By consolidating the workload of user applications into fewer machines [Srikantaiah et al. 2008], unused servers can potentially be switched off or put in low energy consumption modes [Lefèvre and Orgerie 2010]. Yet attracting virtualization is, its sole use does not guarantee reductions in energy consumption. Improving the energy efficiency of Cloud environments with the aid of virtualization generally calls for devising mechanisms that adaptively provision applications with resources that match their workload demands and utilizes other power management technologies such as CPU throttling and dynamic reconfiguration; allowing unused resources to be freed or switched off.

Existing work has proposed architectures that benefit from virtualization for making data centers and Clouds more energy efficient. The problem of energy-efficient resource provisioning is commonly divided into two sub-problems [Liu et al. 2009]: at a micro or host level – discussed earlier – power management techniques are applied to minimize the number of resources used by applications and hence reduce the energy consumed by an individual host; and at a macro-level, generally a Resource Management System (RMS) strives to enforce scheduling and workload consolidation policies that attempt to reduce the number of nodes required to handle the workloads of user applications or place applications in areas of a data center that would improve the effectiveness of the cooling system. Some of the techniques and information commonly investigated and applied at the macro or RMS-level to achieve workload consolidation and energy-efficient scheduling include:

—applications workload estimation;

—the cost of adaptation actions;

—relocation and live-migration of virtual machines;

—information about server-racks, their configurations, energy consumption and thermal states;

—heat management or temperature-aware workload placement aiming for heat distribution and cooling efficiency;

—study of application dependencies and creation of performance models; and

—load balancing amongst computing sites.

Although consolidation fitted with load forecasting schemes can reduce the overall number of resources used to serve user applications, the actions performed by RMSs to adapt the environment to match the application demands can require the relocation and reconfiguration of VMs. This can impact the response time of applications, consequently degrading the QoS perceived by end users. Hence, it is important to consider the costs and benefits of adaptation actions [Verma et al. 2008]. For example, Gueyoung *et al.* [Jung et al. 2009] have explored a cost-sensitive adaptation engine that weights the potential benefits of reconfiguration and their costs. A cost model for each application is built offline and to decide when and how to reconfigure the VMs, the adaptation engine estimates the cost of adaptation actions in terms of changes in the utility, which is a function of the application response time. The benefit of an action is given by the improvement in application response time and the period over which the system remains in the new configuration.

Moreover, consolidation raises the issue of dealing both with necessary redundancy and placement geo-diversity. Cloud providers such as Salesforce.com, who offer to host entire websites of private companies [Hamm 2009], do not want to lose entire websites due to power outages or network access failures. Hence, outages and blackouts should be anticipated and taken into account in resource management policies [Singh and Vara 2009].

We conclude this section by summarizing the existing approaches of energy efficiency for computing resources (Table II). The categories presented in this table are the different research directions detailed in this section.

3. WIRED NETWORKING RESOURCES

The number of Internet users has increased 5 fold between 2000 and 2009[8]. In less than twenty years, the Web has become an essential means of communication for private companies, governments, institutions and other organizations.

[8] Source: `http://www.internetworldstats.com/stats.htm`

Level	Technology	Existing work
Node	Sleep state	[Chase et al. 2001] [Gunaratne et al. 2005] [Leangsuksun et al 2006]
	DVFS	[Snowdon et al. 2005] [Kappiah et al. 2005] [Wang et al. 2010] [Lim et al. 2006] [Chen et al. 2005] [Yan et al. 2005]
	Software improvements	[EcoInfo 2011] [LessWatts 2010] [Steele 1998]
	Hardware capabilities	[Carrera et al. 2003]
Infrastructure	Green sources	[Greenpeace 2011] [Figuerola et al. 2009]
	Thermal management	[Patel et al. 2002] [Sharma et al. 2005] [Merkel and Bellosa 2006]
	Workload consolidation	[Chase and Doyle 2001] [Doyle et al. 2003] [Urgaonkar et al. 2008] [Verma et al. 2008] [Kusic et al. 2008] [Jung et al. 2009] [Srikantaiah et al. 2008] [Freeh et al. 2005] [de Langen and Juurlink 2006] [Lefèvre and Orgerie 2009]
	Task scheduling	[Jejurikar and Gupta 2006] [Mishra et al. 2003] [Yang et al. 2009] [Zhuo and Chakrabarti 2008] [Yao et al. 1995] [He and Jia 2008] [Shin and Kim 2004] [Fan et al. 2007] [Aydi et al. 2001] [Merkel and Bellosa 2006] [Chen et al. 2008] [Da-Costa et al. 2010]
Virtualized environments	Virtual machines	[Nathuji and Schwan 2007] [Stoess et al. 2007] [Talaber et al. 2009] [Torres et al. 2008] [Cherkasova and Gardner 2005] [Hermenier et al. 2006]
	VM migration	[Clark et al. 2005] [Travostino et al. 2006] [Voorsluys et al. 2009]
	Cloud level	[Nurmi et al. 2008] [Fontan et al. 2008] [Barham et al. 2003] [Miyoshi et al. 2002] [Liu et al. 2009] [Verma et al. 2008] [Jung et al. 2009] [Lefèvre and Orgerie 2010]

Table II. Classification of the work on energy-efficiency for computing resources

The ever-increasing number of Internet hosts calls for high performance end-to-end networks, which in turn increases the topology complexity, the number of core components to ensure performance and reliability, and consequently, the energy consumed by the networks [Baldi and Ofek 2009]. Bolla *et al.* [Bolla et al. 2011] forecast that the energy consumption of telecommunication networks will grow 2.5 times by 2018 compared to 2009.

Gupta and Singh [Gupta and Singh 2003], with coarse approximations, have stated that transmitting data through wired networks takes more energy (in bits per Joules) than transmitting it via wireless networks. Energy is indeed one of the main concerns of wireless networks, whereas it is often not the case of wired networks since they are not battery constrained. However, wireless networks are not renowned for their energy-efficiency, and thus, the fact that wired networks are not better is eloquent.

The energy issue is becoming more present in wired networks because of the need for maintaining network connectivity at all times [Christensen et al. 2004].

This rising energy demand can yet be greatly reduced. For example, studies have shown for a few years that network links, especially at the network edges, are lightly utilized [Christensen et al. 2004; Odlyzko 2003].

Moreover, the difference in terms of power consumption between an idle and a fully utilized Ethernet link is negligible [Gunaratne et al. 2005]. These observations have led researchers to propose several approaches to take advantage of link underutilization and reduce the idle power consumption.

The energy consumption of networks is not only incurred by powering networking equipment (routers, switches, links, hubs, etc.), but also by end-hosts that demand high availability and full-time connectivity even if the network is not used.

Current research in energy conservation in wired networks aims to reduce the energy consumption of networking devices while guaranteeing the same Quality of Service (QoS) to users. This *same-for-less* approach is transparent to users whereas network managers, designers and providers should be aware of network usage and energy consumption in order to provide “green network” solutions.

Several methodologies can be used at the macro level with routing strategies and traffic aggregation and at the micro level with hardware improvements in NICs and switches, for example. However, interoperability and backward compatibility with existing protocols and products is generally a requirement.

3.1 Measuring and modeling the energy consumption

Similar to computing infrastructures, before being able to save energy with new technologies and mechanisms, researchers and designers need to know how energy is consumed by networking equipment. This preliminary analysis is key to understand how energy can be saved and to design energy models of networking equipment that will be used to validate new hardware components and new algorithms.

3.1.1 *Estimating and modeling the energy consumed by networks.* The Internet is this familiar network that encompasses thousands of private and public networks with different scales and functions, linked by various electronic and optical networking technologies. Due to its heterogeneity and to privacy and security constraints, the Internet’s topology and density are difficult to evaluate. As a result, precisely estimating the energy consumed by the Internet remains a challenge.

R. Bolla *et al.* have decomposed the Italian network [Bolla et al. 2011]. They forecast the power consumption for a typical network equipment by type of network, and the total number of such devices in the considered overall network (Table III from [Bolla et al. 2011]). This allows one to deduce the energy consumption of the overall network and the portion induced by each of its parts. Hence, a home network equipment is the least consuming, but is the most important contributor to the overall energy consumed by the network (79%) because of the large number of such devices. Therefore, reducing the energy consumption of the least consuming equipment can still represent a lot of energy due to the scaling effect. Similar results based on the topology of an Italian Internet Service Provider (ISP) are presented by Chiaraviglio *et al.* [Chiaraviglio et al. 2010].

Baliga *et al.* model the energy consumption of the Internet depending on the access rate [Baliga et al. 2007]. According to their results, with a peak access rate of 100 Mbps, the Internet accounts for 1% of the electricity consumed in broadband

	power consumption (W)	number of devices	overall consumption (GWh/year)
Home	10	17,500,000	1,533
Access	1,280	27,344	307
Metro/Transport	6,000	1,750	92
Core	10,000	175	15
	Overall network consumption		1,947

Table III. 2015-2020 network forecast: device density and energy requirements in the business-as-usual case (BAU). Example based on the Italian network.

enabled countries; a consumption that can increase up to 4% under a peak access rate of 1 Gbps.

On the client-side, Bianzino *et al.* estimate the power consumption of end-user PCs while browsing the Web depending on the hardware platform, operating system, browser and website. They point out that tabbed browsing cause several scripts to run in parallel while users typically interact with only one tab at any given time, thus wasting energy.

The growth of optical-network usage reflects the increasing demand in bandwidth by new Internet applications (e.g. e-science applications). The energy consumption of optical networks is thus becoming an important issue. Bathula and Elmirghani [Bathula and Elmirghani 2009b] proposed a model expressing the energy required to transmit an optical bit across a Wavelength Routed Node (WRN). Their basic assumption is that an optical network consumes energy in two cases: when transmitting a bit over fiber and when a WRN switches an optical signal. This model, which takes the bit error rate into account, is also applied to other work [Bathula and Elmirghani 2009a] with Optical Burst Switched (OBS) networks that allow the dynamic sub-wavelength switching of data.

Baliga *et al.* [Baliga et al. 2009] present a power consumption model for optical IP networks, which they extrapolate to estimate the energy consumed by the Internet. Similarly to the Internet, Optical IP networks are split into three main parts: the access network, the metropolitan and edge network, and the network core. The power consumption model is described as a function of the access rate to users, and so, the global network consumption is the sum of the per user power consumption for all the components in the network. Thus, this model relies greatly on traffic estimation. A representative example is taken for each type of equipment and the power consumption of this equipment under typical load is included in the model to compute the total network consumption.

Instead of studying the underlying network technology of backbones which are mainly optical networks [Zhang et al. 2010], Chiaraviglio *et al.* [Chiaraviglio et al. 2009] consider a real IP backbone network with real traffic generated by one of the largest ISPs in Italy. The ISP network is divided into four levels, namely: core, backbone, metro and feeder nodes. Even though core nodes are the most power consuming – as far as nodes are concerned – they represent less than 10% of the total energy consumption while feeders represent more than 65%, backbone nodes account for 19% and metro nodes for around 6%. These values do not consider air conditioning, which commonly increases the power consumption by 40% to 60%.

Interconnection networks are another well studied type of network used to relay data and control traffic between computation and storage elements in parallel computer systems [Shang et al. 2006]. The energy consumption model presented by Soteriou *et al.* [Soteriou et al. 2005] for such network type is based on estimating the utilization of links and output buffers. Indeed, contrary to other technologies, in these networks, links are the dominant power consumers.

A holistic approach is presented by Kim *et al.* [Kim et al. 2005] where they present a power consumption model of the entire cluster interconnections based on energy profiles of the three major components: switches, NICs and links. This model leads them to propose energy-efficient designing techniques.

A consensual and commonly adopted way to represent a network topology is to model it as an undirected graph where the vertices are the nodes and the edges are the bidirectional links [Chiaraviglio et al. 2009; Hasan et al. 2010]. Each vertex and each edge has an associated power cost function with parameters varying using random theory. As the network topology is often unknown, random graph theory is used to generate graphs with particular properties that match the observed properties of these networks. Random graph theory can also be used to estimate the number of devices that can eventually be powered off to save energy [Chiaraviglio et al. 2009].

3.1.2 *Measuring and modeling the energy consumed by networking equipment.* The most precise view of the energy consumed by wired networks is focused on network devices as it is possible to plug energy sensors at this small scale. Wired networks consist of several devices such as routers, switches, bridges, optical repeaters, hubs, firewalls, links, coaxial cables, optical fibers and NICs.

Each type of equipment has its own characteristics (architecture and functionalities) and services. Thus, each of these devices presents an energy consumption that is influenced by various parameters such as equipment type, traffic, number of connected devices, manufacturer, number of switched-on interfaces (ports for routers and switches), used protocols and QoS services (which can add processing time if high level operations such as flow identification and packet inspecting are required) and energy saving modes. The following paragraphs describe models for different types of network devices: routers and switches. A router is a device that interconnects two or more networks and forwards data packets from one network to another (routing and forwarding tasks).

Several models have been proposed to describe the energy consumed by an equipment. For example, the energy consumed by routers has been studied in previous work [Shang et al. 2006; Wang et al. 2002; Chabarek et al. 2008] where different parameters are used to model the consumption. A general model for router power consumption depending on the number of active line cards has been proposed by Chabarek *et al.* [Chabarek et al. 2008]. This model is based on results from experiments with different line cards.

Shang *et al.* [Shang et al. 2006] take a different approach where the router actions are broken up into 7 basic operations: input buffer write, routing decision, virtual-channel allocation, switch arbitration, input buffer read, switch traversal and outgoing link traversal. Each operation dissipates dynamic power. Switch arbitration and allocation are not considered since they are negligible. Each of the remaining operations is detailed and mathematical formulas are provided for each power consumption. These models are integrated into their online power estimator embedded in each router in order to dynamically monitor local power consumption. Hence, these models replace an actual expensive wattmeter for each networking equipment. One can hope that such energy sensors will be directly included in future router architecture and will be available to router administrators.

The two previous power models are generic and can be applied to any router. Wang *et al.* [Wang et al. 2002] designed a power consumption model for CMOS (complementary metal-oxide-semiconductor) routers based on the switch capacitance and switching activity. To estimate the switch capacitance, the model includes FIFO buffers, crossbar switch and arbiters modeling, since these three components are the basic building blocks of routers. This model is then applied to two different commercial routers: the Alpha 21364 and the IBM 8-port 12X Infiniband router. The model validation is compared with success to designers' estimates.

The crossbar switch is one of the possible switching fabric architectures. Switch fabrics are responsible for an important part of the router power consumption; 90% in the case of the IBM Infiniband router and 26-35% for the Alpha 21364 (input buffers contribute 46-61% of total power) [Wang et al. 2002]. Ye *et al.* [Ye et al. 2002] studied four widely-used switch fabrics, namely crossbar, fully-connected, Banyan and Batcher-Banyan. The authors show that the fully connected switch has the lowest power consumption, and that the relation between power consumption and traffic throughput is almost linear for the crossbar, the fully-connected and the Batcher-Banyan networks.

A switch, commonly viewed as a layer 2 equipment in the OSI model, connects network segments in LANs. Quantitative models of energy consumption in several optical switching devices are provided by Tucker [Tucker 2010]. These models decompose the device's consumption into the consumption of its hardware components (e.g. power supply and transistors). The author studies two classes of switches: linear analog switches which pass the input signal to the appropriate output without altering the waveform of the signal, and digital switches that operate at the bit level and generally incorporate highly nonlinear and logic devices.

Hlavacs *et al.* [Hlavacs et al. 2009] use linear regression to model the relation between the measured power consumption and the injected traffic in typical residential and professional switches. Some results are surprising: the Netgear residential switch (Netgear FS608v2, 8 ports) consumes less energy under high bandwidth values, and the 3Com professional switch (3Com 3824, 24 ports) achieves the optimum ratio between energy consumption and bandwidth when operating between 100 and 1000 kbit/s. As a conclusion, they show that energy consumption and bandwidth are linked, but that the dependence is quite small and not linear.

This heterogeneous behavior has led Mahadevan *et al.* to propose a benchmarking framework to compare the power characteristics of a variety of network devices (routers and switches) [Mahadevan et al. 2009]. The power model for each equipment is based on the base chassis power, the number of linecards, the number of active ports, the port capacity, the port utilization, the Ternary Content Address-able Memory (TCAM) and the firmware. This benchmarking framework is able to predict the power consumed by different switches currently used in data centers within a 2% error margin.

As we go from core networks to access networks, the bandwidth capacity and number of components per equipment decrease, but the number of deployed equipment increases. As both the number of network devices and their bandwidth capacity affect the energy consumption, possible energy savings are greater at both core and access levels. In fact, in 2000 in U.S. [Gupta and Singh 2003], 95,000 LAN switches were deployed consuming 3.2 TeraWatt hours per year; and 3,257 routers were deployed consuming 1.1 TeraWatt hours per year.

However, when no empirical measurements of energy consumption are available for a given network equipment, researchers use generic mathematical models to evaluate their algorithms and frameworks. Such models express the relation between traffic load and energy consumption with simple functions, including linear, cubic, logarithmic, step functions [Restrepo et al. 2009]. These energy profiles may not be realistic [Mahadevan et al. 2009] as they represent an ideal situation, a scenario where the power consumed is proportional to the load. Such a proportional model is often used and referred to as the power consumption model for future networking equipment [Mahadevan et al. 2009].

Several models have been proposed for the different network components [Wang et al. 2002; Ananthanarayanan and Katz 2008; Shang et al. 2006], for different types of networks [Baliga et al. 2009; Chiaraviglio et al. 2009; Baliga et al. 2008], and for the whole Internet [Baldi and Ofek 2009; Baliga et al. 2007; Chiaraviglio et al. 2009]. Combined with real energy measurements, they allow researchers to validate their new frameworks and algorithms. As for computing resources, energy-efficient techniques targeting networking resources can be divided into different levels from hardware optimizations to network-wide solutions. The following sections describe these techniques starting from low-level optimizations.

3.2 Hardware and low-level optimizations

A way to reduce the energy consumption of a network is to increase the efficiency of its components. For that reason manufacturers of network equipment are constantly proposing *green* routers and switches [Ananthanarayanan and Katz 2008]. D-Link, Cisco, Netgear are among the manufacturers proposing new *green* functionalities in their products such as adapting the power transmission to the link length and to the load, power off buttons, and more energy efficient power supplies.

These new products come often as result of green initiatives (e.g. GreenTouch[9], GreenStar Network[10] and the Energy Consumption Rating initiative[11]) and study groups (e.g. IEEE 802.3 Energy Efficient Ethernet Study Group) which aim to standardize and to enforce new regulations in terms of energy consumption for network equipment.

As result of these initiatives, several hardware solutions have been proposed, from improving hardware capabilities (e.g. power saving by cable length, smart

[9] `http://greentouch.org`

[10] `http://greenstarnetwork.com`

[11] `http://www.ecrinitiative.org/`

cooling FAN, disk spin down and DVFS) to new functionalities (e.g. power on/off button, low power mode, port auto power down). Power saving according to a cable length is similar to the power saving modes of wireless equipment that adjust their radio range according to the distance of their neighbors. The specifications of typical Ethernet twisted-pair cables precise a maximum length of 100 meters. Hence, router and switch ports are basically set for the maximal cable length, but at homes and enterprise environments, they are most commonly used with cables measuring a couple of meters.

Networking equipment can also benefit from low-level improvements coming from research on energy-efficient computing resources [Gupta and Singh 2007b] such as disk spin down and DVFS presented in Section 2. Power on/off buttons enable home equipment to be manually switched off when they are not used. Wake-On-LAN techniques allow them to be switched on and off remotely.

When ports are not used at their full capacity, they can be put in low power modes and when they are not used at all (no plugged link), they can be automatically powered down. This latter improvement is based on sleeping methods detailed in the next section. Hardware optimizations also include re-engineering the network component to use more energy-efficient technologies and to reduce the complexity of current devices [Roberts 2009].

3.3 Shutdown: sleeping methods

As discussed earlier, network links – especially those at the edges of a network – are lightly utilized [Christensen et al. 2004; Odlyzko 2003]. Therefore, in order to save energy researchers have proposed techniques that switch off network equipment (i.e. put them into sleep mode) when they are not used [Gupta and Singh 2003; Chiaraviglio et al. 2008]. Despite its benefits, this technique raises several problems including connectivity loss and long re-synchronization periods. In addition, constantly switching network devices off and on can be more energy consuming than keeping them on all the time.

New mechanisms have been designed to address these issues. Examples of mechanisms include proxying techniques to maintain connectivity [Nordman and Christensen 2010] and to quickly re-synchronized both ends of a link [Gupta and Singh 2007a]. Here we first review the main techniques and algorithms for switching off network components. Then, we describe proxying techniques to enable longer switching-off periods.

3.3.1 *On/off and sleeping techniques and frameworks.* A seminal work dealing with on/off approaches was performed by Gupta and Singh [Gupta and Singh 2003], where they proposed two solutions to reduce energy wastage: 1) to switch off network interfaces of LAN switches during packet inter-arrival times, which requires to extend the interface's memory buffers to minimize packet loss; 2) to modify routing protocols in order to aggregate all traffic going through parallel routes into one route during low-traffic periods and thus, to enable unused network links to be switched off.

The first solution, uncoordinated and passive, takes advantage of periods without traffic, whereas the second, coordinated and active, tries to increase these periods. Therefore, the first solution requires to improve low-level capabilities, whereas the second deals with the control plane. These approaches are not straightforward since the network presence of sleeping elements should be maintained. According to the experiments performed by Gupta and Singh [Gupta and Singh 2007b], links of an on-campus backbone switch can be powered off from 40% to more than 80% of the time.

Observing that current network devices do not have power management primitives, Bolla *et al.* [Bolla et al. 2011] proposed standby primitives (sleeping mode) for backbone devices using virtualization capabilities of layer 2 protocols (mainly represented by MPLS and Ethernet) and completely transparent to layer 3 protocols (such as IP). A network node is devoted to collecting traffic load information from routers, applying necessary reconfigurations, and switching elements on and off to meet QoS constraints[12]. This solution also requires periodic wake-ups of sleeping hardware for failure detections.

Hu *et al.* [Hu et al. 2011] proposed reconfigurable routers to deal with traffic and route aggregation and management of routers under sleeping states. Dynamic Ethernet Link Shutdown (DELS), presented by Gupta and Singh [Gupta and Singh 2007a], takes sleeping decisions based on buffer occupancy, the behavior of previous packet arrival times and a configurable maximum bounded delay.

The problem of finding the minimum set of nodes and links to be powered on while guaranteeing full connectivity and maximum link utilization is NP-hard and has been modeled by Chiaraviglio *et al.* [Chiaraviglio et al. 2008]. This problem is also studied by Yamanaka *et al.* [Yamanaka et al. 2010] with different QoS constraints (hop limit, bandwidth limit reliability and stability). Numerous heuristics have been proposed to solve it [Gupta and Singh 2007a; Nedevschi et al. 2008; Soteriou and Peh 2003; Chiaraviglio et al. 2008; 2009; Restrepo et al. 2009], but most do not consider some practical problems of the on/off approach: switching on and off requires time, it leads to a network reconfiguration because of topology change, and a wake-up method is required to determine how and when nodes and links should be switched on again. To solve the latter problem, several solutions can be envisaged such as a periodic wake-up [Nedevschi et al. 2008], an automatic wake-up on sensing incoming traffic [Gupta and Singh 2003], or a "dummy" packet sent to wake up a neighboring node [Gupta et al. 2004].

Instead of a sleeping mode, the IEEE 802.3az task group[13] proposes Low Power Idle (LPI) [Christensen et al. 2010], a standard that has been adopted in September 2010 [Reviriego et al. 2011]. The basic idea – to transmit data as fast as possible to spend as much time as possible in low power idle mode – is based on the following principles: the highest rate provides the most energy-efficient transmission (in Joules/bit) and the LPI mode consumes minimal power. The LPI transition is initiated by the transmitter. Then, a periodical refreshment is done to detect failures and facilitate fast transition. Transmissions are deferred to pre-defined time as shown in Figure 2 presented in [Reviriego et al. 2010].

T_s is sleep time (the time needed to enter sleep mode); T_w is wake-up time (the

[12] QoS described in terms of maximum link utilization and backup availability while allowing the largest number of line cards to sleep

[13] The IEEE 802.3az (Energy Efficient Ethernet) task group develops standards for reducing power consumption of Ethernet devices (`http://www.ieee802.org/3/az/public/index.html`).

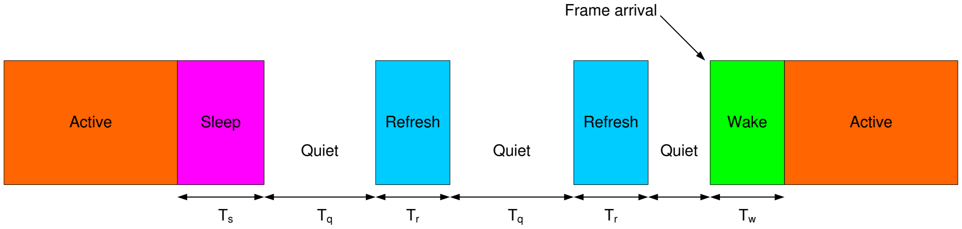

Fig. 2. Transitions between the active and sleep modes in Energy-Efficient Ethernet.

time required to exit sleep mode). The transceiver spends T_q in the quiet (energy-saving) period but also has short periods of activity (T_r) to refresh the receiver state. Currently, LPI seems the most promising solution at the NIC level to save energy. The first NICs enabling this green functionality have been commercialized at the end of 2011.

3.3.2 *Proxying techniques.* In 1998, a study showed that PCs are generally powered on even when they are not in use; an observation that has led Irish & Christensen [Irish et al. 1998] to look for a solution consisting in completely powering off PCs during idle periods. Even though powering off network components such as links or switches is not yet commonplace, switching off PCs include powering off their NICs, and thus dealing with loss of network connectivity.

A way to address these issues is to use a proxy to answer to non-urgent messages on behalf of a node and to wake-up the node when required. This solution, detailed in existing work [Christensen et al. 2004; Gunaratne et al. 2005; Jimeno et al. 2008], is based on a Network Connectivity Proxy (NCP) that handles network presence requests such as ARP, ICMP and DHCP, and keeps connections alive. This proxy solution can be implemented in the NICs themselves as proposed by Sabhanatarajan *et al.* [Sabhanatarajan et al. 2008]. Agarwal *et al.* proposed an improved NIC that answers to network requests while the node is in suspend-to-RAM state [Agarwal et al. 2009]. Another approach illustrated by Agarwal *et al.* [Agarwal et al. 2010] is to use dedicated on-demand proxy servers.

Taking advantage of low link utilization and path redundancy, the shutdown approach uses idle periods in network traffic. Energy savings can also be made during low-demand period with the slowdown approach.

3.4 Slowdown: adapting to the needs

The difference in power consumption between an idle and a fully utilized Ethernet link is negligible [Gunaratne et al. 2005]. To keep communicating NICs synchronized, idle bit patterns are continuously transmitted even when no actual data is sent. As a consequence, the power consumption levels of non-energy aware equipment are the same when idle and transmitting. An exception are the 1000BASE-T specifications, designed to operate at 10 Mb/s, 100 Mb/s and 1 Gb/s in order to keep backward compatibility with previous specifications. It has been observed that running Ethernet links at lower data rates decreases the power consumption of NICs and switches [Zhang et al. 2008].

Similarly to computing resources, the energy consumption of networking devices is not proportional to their usage [Gunaratne et al. 2005]. To get closer to proportionality, the transmission rate of Ethernet devices can be adapted to the load, similarly to DVFS techniques. Under this technique, called Adaptive Link Rate (ALR) [Gunaratne et al. 2005; Gunaratne et al. 2006], a high and a low-buffer thresholds are determined. When the buffer occupancy reaches the high-buffer threshold, the link rate is switched to a higher value, and when it goes under the low-buffer threshold, the link rate decreases. The difficulty lies in finding the good values for these thresholds in order to avoid packet losses and oscillations since rate switching takes times. An ALR prototype is implemented by Zhang *et al.* [Zhang et al. 2008] showing that switching times are in the order of milliseconds. However, the IEEE 802.3az group, which studied ALR before designing LPI, showed that under more generic conditions, ALR suffers from too lenghty switching times and frequent oscillations which impede its ability to save energy.

A similar approach, called Dynamic Adjustment of Link Width (DAWL), is proposed by Alonso *et al.* [Alonso et al. 2004]. The thresholds are also based on link utilization. Nedevschi *et al.* provide a rate adaptation mechanism [Nedevschi et al. 2008], and as they outline, the distribution of operating rates and their corresponding power consumption greatly influence the efficiency of slowdown techniques.

3.5 Coordination: network-wide management and global solutions

The optimization, shutdown and slowdown approaches focus on specific network components. However, energy efficiency improvements have also to be made at wider scales. For example, routing algorithms [Chabarek et al. 2008; Restrepo et al. 2009; Zhang et al. 2010] and network protocols [Irish et al. 1998; Blackburn and Christensen 2009] can be improved to save energy.

Coordinated power management schemes benefit from previously cited techniques, such as on/off and adapting rate techniques, and take decisions at a wider scale. This implies that they make greater energy savings. For example, in low-demand scenarios with network redundancy, entire network paths can be switched off, and the traffic is routed on other paths [Idzikowski et al. 2010; Shang et al. 2006].

Figure 3 [Bolla et al. 2011] shows different options for the same transfer scenario:

(a) no power-aware optimizations;

(b) shutdown approaches;

(c) slowdown approaches; and

(d) shutdown and slowdown approaches combined.

Both shutdown and slowdown approaches increase the total transfer time and reduce the energy consumption. The combination of these two techniques gives the best results in terms of energy consumption and the worst results with regard to transfer time. However, depending on the QoS constraints, this delay can be acceptable and can be shortened using traffic prediction. An analysis of these two methods is provided by Nedevschi *et al.* [Nedevschi et al. 2008] showing that both methods can be beneficial depending on the power profile of the network equipment and the utilization of the network itself.

Among the solutions to improve the energy efficiency at a network scale, we

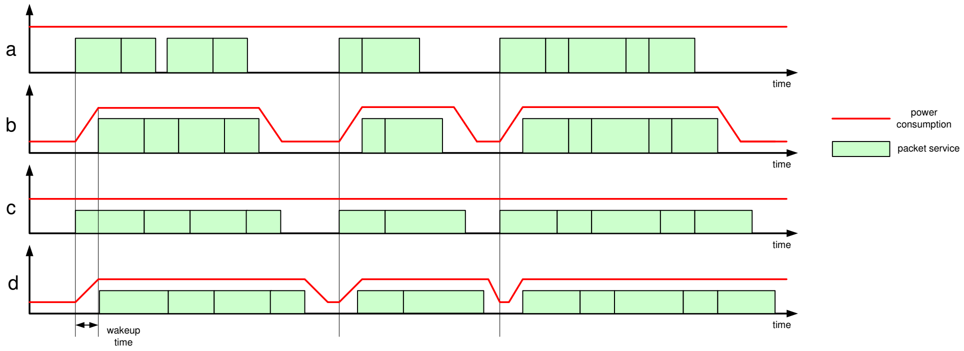

Fig. 3. Packet service times and power consumptions in the following cases: (a) no power-aware optimizations, (b) only idle logic, (c) only performance scaling, (d) performance scaling and idle logic.

distinguish three categories: the green design of network protocols which concern improvements of current protocols to make them more energy-efficient; the clean-slate approaches which propose new energy-aware network architectures, and the energy-aware frameworks which consider the entire network consumption as the quantity to optimize.

3.5.1 *Green design of network protocols.* Each network layer has its own protocol and associated energy cost. Wang and Singh [Wang and Singh 2004] analyzed the energy consumption of TCP and isolated the cost of each operation (e.g. copies, computing checksums). They showed that several techniques can be employed to reduce energy consumption, such as zero copy, maintaining the send buffer on the NIC and copying data in large chunks.

In 1998, Irish & Christensen [Irish et al. 1998] proposed a "Green TCP/IP" with a connection-sleep option; an option that works on the basis of client-server applications with connection-oriented communications. It allows a power-managed TCP/IP client to inform a TCP/IP server that it is going to a sleeping power mode. The server thus keeps the connection alive, but it does not send any packet to the sleeping client.

Energy savings can also be made at the routing protocol level. Cianfrani *et al.* [Cianfrani et al. 2010] adapted the Open Shortest Path First (OSPF) protocol for allowing routers to power off network links during low-traffic periods. A power-aware routing protocol for IP routers using linecard and chassis reconfiguration is proposed by Chabarek *et al.* [Chabarek et al. 2008]. "GreenTE" an intra-domain power-aware traffic engineering mechanism with a good routing stability is proposed by Zhang *et al.* [Zhang et al. 2010]. A centralized routing and on/off management is described by Gelenbe and Silvestri [Gelenbe and Silvestri 2009]; it presents good QoS results in terms of delay, packet loss and jitter.

Traffic grooming is used in optical wavelength routing network to group flows to avoid optical-electronic-optical conversions (OEOs) and thus save energy. Zhang *et al.* [Zhang et al. 2010] compare static and dynamic grooming policies under various loads, and the best approach depends on the considered scenario. Van Heddeghem *et al.* [Van Heddeghem et al. 2010] show that optical end-to-end grooming consumes about half the power of the link-by-link grooming on a realistic scenario.

3.5.2 *Clean-slate approaches.* The IP was initially designed in the 1970s. By 1980, around 200 hosts were connected to the – to be – Internet and nowadays, there are over 2 billion Internet users in the world[14]. As sometimes several years are required to deploy new network protocols, these protocols may have been designed to traffic conditions that are not the case by the time they are rolled out. Clean-slate approaches disregard current design constraints (such as interoperability) and propose new innovative architectures that improve the QoS of current and future applications.

These new architectures comprise bufferless routers [Hayenga et al. 2009], optimal network design for specific applications [Baliga et al. 2009], synchronous IP switching to synchronize router's operations and schedule traffic in advance [Baldi and Ofek 2009], pure optical switching architectures [Tucker 2010] and routing algorithms dealing with flows instead of packets [Roberts 2009].

3.5.3 *Energy-aware frameworks.* To be energy efficient, network-wide solutions should be:

—adapted to the network topology (e.g. redundancy, multi-paths);

—adapted to the traffic (e.g. bursts, always low);

—adapted to the scenario usage (e.g. P2P, Web servers);

—realistic in terms of technology (e.g. compatibility, interoperability); and

—scalable, reliable, fast, fault-tolerant, efficient and secure.

Coordinated power management schemes benefit from previously cited techniques, such has on/off and adaptive link rate, and take decisions at a wider scale, which implies that they make greater energy savings. For example, in low-demand scenarios with network redundancy, entire network paths can be switched off, and the traffic routed on other paths [Idzikowski et al. 2010; Shang et al. 2006; Steinder et al. 2008; Chiaraviglio and Matta 2010]. However, these solutions require major changes such as dynamic routing protocols that can handle shutdown nodes and links, a centralized management to coordinate switch off [Gelenbe and Silvestri 2009] and higher-level cooperation between ISPs and Content Providers (CP) [Chiaraviglio and Matta 2010]. Bandwidth provisioning frameworks can also be adapted to save energy [Orgerie and Lefèvre 2011].

Network virtualization is another a promising solution to enable energy savings. Tzanakaki *et al.* [Tzanakaki et al. 2011] propose an energy-aware planning of virtual infrastructure (VI).[15] They show that their framework can save up to 40% of energy. Wang *et al.* [Wang et al. 2008] present a network management primitive for Virtual ROuters On the Move (VROOM). It simplifies the management of routers by virtualizing them and allowing them to freely move from one physical node to another, thus avoiding logical topology changes.

[14] Source: `http://www.worldometers.info/`

[15] The objective of VI planning is to identify the topology and determine the virtual resources required to implement a dynamically reconfigurable VI based on both optical network and IT resources.

Approach	Technology	Existing work
Hardware	Improving and re-engineering	[Gupta and Singh 2007b] [Ananthanarayanan and Katz 2008] [Roberts 2009]
Shutdown	On/off and sleeping	[Gupta and Singh 2003] [Nedevschi et al. 2008] [Gupta et al. 2004] [Bolla et al. 2011] [Hu et al. 2011] [Chiaraviglio et al. 2008] [Yamanaka et al. 2010] [Soteriou and Peh 2003] [Christensen et al. 2010] [Gupta and Singh 2007a]
	Proxying	[Irish et al. 1998] [Christensen et al. 2004] [Jimeno et al. 2008] [Sabhanatarajan et al. 2008] [Agarwal et al. 2009] [Agarwal et al. 2010] [Gunaratne et al. 2005]
Slowdown	Rate adaptation	[Zhang et al. 2008] [Gunaratne et al. 2006] [Nedevschi et al. 2008] [Alonso et al. 2004]
Coordination	Network protocols	[Irish et al. 1998] [Wang and Singh 2004] [Cianfrani et al. 2010] [Zhang et al. 2010] [Van Heddeghem et al. 2010] [Chabarek et al. 2008] [Zhang et al. 2010] [Gelenbe and Silvestri 2009] [Blackburn and Christensen 2009]
	Clean-slate	[Baldi and Ofek 2009] [Hayenga et al. 2009] [Tucker 2010] [Baliga et al. 2009]
	Frameworks	[Chiaraviglio et al. 2009] [Shang et al. 2006] [Chiaraviglio et al. 2009] [Chiaraviglio and Matta 2010] [Steinder et al. 2008] [Restrepo et al. 2009] [Wang et al. 2008] [Tzanakaki et al. 2011] [Orgerie and Lefèvre 2011]

Table IV. Taxonomy of work on improving the energy efficiency of networking resources according to their levels and approaches.

To summarize, Table IV proposes a synthesis of the research work discussed in this section. The work is categorized by network level and type of approach. The three first approaches are applied at the node level whereas the last one is applied at the network level.

4. CONCLUSION

This survey discussed techniques for improving the energy efficiency of computing and networking resources in large-scale distributed systems. As discussed, during the past decade, solutions have been proposed for improving the energy efficiency of computing and networking resources. For computing resources the solutions work at different levels, from individual nodes to entire infrastructures where they take advantage of recent advanced functionalities such as virtualization. In parallel, for wired networks, shutdown techniques have been extensively studied and evaluated to limit the number of resources that can remain idle and consume energy unnecessarily. There are also techniques for adapting the performance of both computing and network resources (and their energy usage) to the needs of applications and services. These approaches are often combined and applied in a coordinated way in large scale distributed systems.

After exploring studies and models for estimating the energy consumption of these resources, we presented a classification of existing solutions and research work. While many research directions are studied to save energy, several key open problems remain: are virtualization and Cloud computing the panacea for saving energy? Which architecture is the most energy efficient: centralized, totally distributed or in-between? How to make a trade-off between energy and performance? How to reach energy proportionality? Nonetheless, one of the main leverages to reduce the electric bill and the carbon footprint of IT infrastructure is to increase the energy awareness of users and providers.

REFERENCES

Agarwal, Y., Hodges, S., Chandra, R., Scott, J., Bahl, P., and Gupta, R. 2009. Somniloquy: Augmenting Network Interfaces To Reduce PC Energy Usage. In *USENIX Symposium On Networked Systems Design And Implementation (NSDI)*. USENIX Association, Berkeley, CA, USA, 365–380.

Agarwal, Y., Savage, S., and Gupta, R. 2010. SleepServer: A software-only approach for reducing the energy consumption of PCs within enterprise environments. In *USENIX Annual Technical Conference (ATC)*. USENIX Association, Berkeley, CA, USA.

Allalouf, M., Arbitman, Y., Factor, M., Kat, R., Meth, K., and Naor, D. 2009. Storage modeling for power estimation. In *ACM Israeli Experimental Systems Conference (SYSTOR)*.

Alonso, M., Martinez, J. M., Santonja, V., and Lopez, P. 2004. Reducing power consumption in interconnection networks by dynamically adjusting link width. In *International European Conference on Parallel and Distributed Computing (Euro-Par)*. 882–890.

Ananthanarayanan, G. and Katz, R. 2008. Greening the switch. Tech. rep., EECS Department, University of California, Berkeley.

Aydi, H., Mejia-Alvarez, P., Mossé, D., and Melhem, R. 2001. Dynamic and Aggressive Scheduling Techniques for Power-Aware Real-Time Systems. In *IEEE Real-Time Systems Symposium (RTSS)*. 95–105.

Baldi, M. and Ofek, Y. 2009. Time For A "Greener" Internet. In *International Workshop On Green Communications (GreenCom, IEEE ICC Workshop)*.

Baliga, J., Ayre, R., Hinton, K., Sorin, W., and Tucker, R. 2009. Energy Consumption in Optical IP Networks. *Journal of Lightwave Technology 27,* 13, 2391–2403.

Baliga, J., Ayre, R., Hinton, K., and Tucker, R. 2009. Architectures for energy-efficient IPTV networks. In *Conference on Optical Fiber Communication (OFC)*.

Baliga, J., Ayre, R., Sorin, W., Hinton, K., and Tucker, R. 2008. Energy Consumption In Access Networks. In *Conference On Optical Fiber Communication/National Fiber Optic Engineers Conference (OFC/NFOEC)*. 1–3.

Baliga, J., Hinton, K., and Tucker, R. 2007. Energy Consumption Of The Internet. In *Joint International Conference On Optical Internet And Australian Conference On Optical Fibre Technology (COIN-ACOFT)*. 1–3.

Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., Neugebauer, R., Pratt, I., and Warfield, A. 2003. Xen and the Art of Virtualization. In *ACM Symposium on Operating Systems Principles (SOSP)*. 164–177.

Barroso, L. and Holzle, U. 2007. The Case for Energy-Proportional Computing. *Computer 40,* 12, 33–37.

Barroso, L. A. 2005. The Price of Performance. *ACM Press, Queue 3*, 48–53.

Bathula, B. and Elmirghani, J. 2009a. Energy Efficient Optical Burst Switched (OBS) Networks. In *GreenComm: International Workshop On Green Communications*.

Bathula, B. and Elmirghani, J. 2009b. Green networks: Energy efficient design for optical networks. In *IFIP International Conference on Wireless and Optical Communications Networks*.

Blackburn, J. and Christensen, K. 2009. A Simulation Study Of A New Green BitTorrent. In *International Workshop On Green Communications (GreenCom, IEEE ICC Workshop)*. 1–6.

Bolla, R., Bruschi, R., Christensen, K., Cucchietti, F., Davoli, F., and Singh, S. 2011. The Potential Impact of Green Technologies in Next Generation Wireline Networks - Is There Room for Energy Savings Optimization? *IEEE Communications*.

Bolla, R., Bruschi, R., Cianfrani, A., and Listanti, M. 2011. Enabling backbone networks to sleep. *IEEE Network 25,* 2, 26–31.

Bolla, R., Bruschi, R., Davoli, F., and Cucchietti, F. 2011. Energy Efficiency in the Future Internet: A Survey of Existing Approaches and Trends in Energy-Aware Fixed Network Infrastructures. *IEEE Communications Surveys and Tutorials 13,* 2, 223–244.

Boss, G., Malladi, P., Quan, D., Legregni, L., and Hall, H. 2007. Cloud Computing. Tech. rep., IBM.

Box, G., Jenkins, G., and Reinsel, G. 1994. *Time Series Analysis: Forecasting and Control*, 3rd ed. Prentice-Hall International, Inc.

Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., and Brandic, I. 2009. Cloud Computing and Emerging IT Platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility. *Future Generation Computer Systems 25,* 6, 599–616.

Carrera, E., Pinheiro, E., and Bianchini, R. 2003. Conserving disk energy in network servers. In *International conference on Supercomputing (ICS)*. 86–97.

Castagnetti, A., Belleudy, C., Bilavarn, S., and Auguin, M. 2010. Power Consumption Modeling for DVFS Exploitation. In *Euromicro Conference on Digital System Design: Architectures, Methods and Tools (DSD)*. 579–586.

Chabarek, J., Sommers, J., Barford, P., Estan, C., Tsiang, D., and Wright, S. 2008. Power awareness in network design and routing. In *IEEE International Conference on Computer Communications (INFOCOM)*. 457–465.

Chase, J., Anderson, D., Thakar, P., Vahdat, A., and Doyle, R. 2001. Managing energy and server resources in hosting centers. In *ACM symposium on Operating systems principles (SOSP)*. 103–116.

Chase, J. and Doyle, R. 2001. Balance of Power: Energy Management for Server Clusters. In *Workshop on Hot Topics in Operating Systems (HotOS)*.

Chen, G., He, W., Liu, J., Nath, S., Rigas, L., Xiao, L., and Zhao, F. 2008. Energy-aware server provisioning and load dispatching for connection-intensive internet services. In *USENIX Symposium on Networked Systems Design and Implementation (NSDI)*. 337–350.

Chen, G., Malkowski, K., Kandemir, M., and Raghavan, P. 2005. Reducing Power with Performance Constraints for Parallel Sparse Applications. In *International Parallel and Distributed Processing Symposium (IPDPS)*.

Cherkasova, L. and Gardner, R. 2005. Measuring CPU overhead for I/O processing in the Xen virtual machine monitor. In *USENIX Annual Technical Conference (ATEC)*.

Chiaraviglio, L., Ciullo, D., Leonardi, E., and Mellia, M. 2009. How Much Can The Internet Be Greened? In *GreenComm: International Workshop On Green Communications*.

Chiaraviglio, L. and Matta, I. 2010. GreenCoop: Cooperative Green Routing with Energy-efficient Servers. In *International Conference on Energy-Efficient Computing and Networking (e-Energy)*.

Chiaraviglio, L., Mellia, M., Lombardo, A., Panarello, C., and Schembra, G. 2010. Energy Saving And Network Performance: A Trade-Off Approach. In *International Conference On Energy-Efficient Computing And Networking*.

Chiaraviglio, L., Mellia, M., and Neri, F. 2008. Energy-Aware Networks: Reducing Power Consumption By Switching Off Network Elements. Tech. rep., FEDERICA-Phosphorus tutorial and workshop (TNC).

Chiaraviglio, L., Mellia, M., and Neri, F. 2009. Energy-aware backbone networks: A case study. In *International Workshop On Green Communications (GreenCom, IEEE ICC Workshop)*.

Christensen, K., Gunaratne, C., Nordman, B., and George, A. 2004. The next frontier for communications networks: power management. *Computer Communications 27,* 18, 1758–1770.

Christensen, K., Reviriego, P., Nordman, B., Bennett, M., Mostowfi, M., and Maestro, J. 2010. IEEE 802.3az: the road to energy efficient ethernet. *IEEE Communications Magazine 48,* 11, 50–56.

Cianfrani, A., Eramo, V., Listanti, M., Marazza, M., and Vittorini, E. 2010. An Energy Saving Routing Algorithm for a Green OSPF Protocol. In *IEEE Conference on Computer Communications (INFOCOM) Workshops*.

Clark, C., Fraser, K., Hand, S., Hansen, J. G., Jul, E., Limpach, C., Pratt, I., and Warfield, A. 2005. Live migration of virtual machines. In *Symposium on Networked Systems Design & Implementation (NSDI)*. 273–286.

Da-Costa, G., Dias de Assuncao, M., Gelas, J.-P., Georgiou, Y., Lefèvre, L., Orgerie, A.-C., Pierson, J.-M., Richard, O., and Sayah, A. 2010. Multi-facet approach to reduce energy consumption in clouds and grids: The green-net framework. In *e-Energy 2010 : First International Conference on Energy-Efficient Computing and Networking*. Passau, Germany, 95–104.

de Langen, P. and Juurlink, B. 2006. Leakage-aware multiprocessor scheduling for low power. In *International conference on Parallel and Distributed Processing (IPDPS)*.

Dietz, H. and Dieter, W. 2006. Compiler and runtime support for predictive control of power and cooling. In *Parallel and Distributed Processing Symposium (IPDPS)*.

Doyle, R., Chase, J., Asad, O., Jin, W., and Vahdat, A. 2003. Model-based resource provisioning in a Web service utility. In *USENIX Symposium on Internet Technologies and Systems (USITS)*.

EcoInfo 2011. From windows 95 to windows 7. EcoInfo report.

Fan, X., Weber, W.-D., and Barroso, L. A. 2007. Power provisioning for a warehouse-sized computer. In *ACM International symposium on Computer architecture (ISCA)*. 13–23.

Feng, W.-C. and Scogland, T. 2009. The Green500 List: Year One. In *Workshop on High-Performance, Power-Aware Computing (HPPAC, IPDPS workshop)*.

Figuerola, S., Lemay, M., Reijs, V., Savoie, M., and St. Arnaud, B. 2009. Converged Optical Network Infrastructures in Support of Future Internet and Grid Services Using IaaS to Reduce GHG Emissions. *Journal of Lightwave Technology 27,* 12, 1941–1946.

Fontan, J., Vazquez, T., Gonzalez, L., Montero, R. S., and Llorente, I. 2008. OpenNebula: The open source virtual machine manager for cluster computing. In *Open Source Grid and Cluster Software Conference – Book of Abstracts*.

Freeh, V., Pan, F., Kappiah, N., Lowenthal, D., and Springer, R. 2005. Exploring the Energy-Time Tradeoff in MPI Programs on a Power-Scalable Cluster. In *IEEE International Parallel and Distributed Processing Symposium, (IPDPS)*.

Ge, R., Feng, X., and Cameron, K. 2005. Performance-constrained Distributed DVS Scheduling for Scientific Applications on Power-aware Clusters. In *ACM/IEEE Conference on Supercomputing (SC)*.

Gelenbe, E. and Silvestri, S. 2009. *Optimisation of Power Consumption in Wired Packet Networks*. Vol. 22. Springer, 717–729.

Geller, T. 2011. Supercomputing's exaflop target. *Communications of the ACM 54*, 16–18.

Green Grid 2007. The Green Grid Data Center Power Efficiency Metrics: PUE and DCiE. Green Grid report.

Green Grid 2010. Unused Servers Survey Results Analysis. Green Grid report.

Greenberg, A., Hamilton, J., Maltz, D., and Patel, P. 2008. The cost of a cloud: research problems in data center networks. *ACM SIGCOMM Computer Communication Review 39*, 68–73.

Greenpeace 2011. How dirty is your data? Greenpeace report.

Grit, L., Irwin, D., Yumerefendi, A., and Chase, J. 2006. Virtual Machine Hosting for Networked Clusters: Building the Foundations for "Autonomic" Orchestration. In *International Workshop on Virtualization Technology in Distributed Computing (VTDC)*.

Gunaratne, C., Christensen, K., and Nordman, B. 2005. Managing energy consumption costs in desktop PCs and LAN switches with proxying, split TCP connections, and scaling of link speed. *International Journal of Network Management 15,* 5, 297–310.

Gunaratne, C., Christensen, K., and Suen, S. 2006. Ethernet Adaptive Link Rate (ALR): Analysis Of A Buffer Threshold Policy. In *IEEE Global Telecommunications Conference (GLOBECOM)*. 1–6.

Gupta, M., Grover, S., and Singh, S. 2004. A Feasibility Study For Power Management In LAN Switches. In *IEEE International Conference On Network Protocols (ICNP)*. 361–371.

GUPTA, M. AND SINGH, S. 2003. Greening of the internet. In *SIGCOM Conference on Applications, technologies, architectures, and protocols for computer communications*. 19–26.

GUPTA, M. AND SINGH, S. 2007a. Dynamic Ethernet Link Shutdown For Energy Conservation On Ethernet Links. In *IEEE International Conference On Communications (ICC'07)*. 6156–6161.

GUPTA, M. AND SINGH, S. 2007b. Using Low-Power Modes For Energy Conservation In Ethernet LANs. In *IEEE International Conference On Computer Communications (INFOCOM)*. 2451–2455.

HAMM, S. 2009. With Sun, IBM Aims for Cloud Computing Heights.

HARIZOPOULOS, S., SHAH, M., MEZA, J., AND RANGANATHAN, P. 2009. Energy Efficiency: The New Holy Grail of Data Management Systems Research. In *Conference on Innovative Data Systems Research (CIDR)*.

HASAN, M., FARAHMAND, F., PATEL, A., AND JUE, J. 2010. Traffic Grooming in Green Optical Networks. In *IEEE International Conference on Communications (ICC)*.

HAYENGA, M., JERGER, N., AND LIPASTI, M. 2009. SCARAB: A single cycle adaptive routing and bufferless network. In *IEEE/ACM International Symposium on Microarchitecture (MICRO)*. 244–254.

HAYES, B. 2008. Cloud computing. *Communication of the ACM 51,* 7, 9–11.

HE, X. AND JIA, Y. 2008. Procrastination Scheduling for Fixed-Priority Tasks with Preemption Thresholds. In *IFIP International Conference on Network and Parallel Computing (NPC)*. 255–265.

HERMENIER, F., LORIANT, N., AND MENAUD, J.-M. 2006. Power Management in Grid Computing with Xen. In *Frontiers of High Performance Computing and Networking – ISPA 2006 Workshops*. Number 4331 in LNCS. 407–416.

HLAVACS, H., COSTA, G. D., AND PIERSON, J.-M. 2009. Energy consumption of residential and professional switches. In *IEEE International Conference on Computational Science and Engineering (CSE)*. 240–246.

HU, C., WU, C., XIONG, W., WANG, B., WU, J., AND JIANG, M. 2011. On the design of green reconfigurable router toward energy efficient internet. *IEEE Communications Magazine 49,* 6, 83–87.

IDZIKOWSKI, F., ORLOWSKI, S., RAACK, C., WOESNER, H., AND WOLISZ, A. 2010. Saving energy in IP-over-WDM networks by switching off line cards in low-demand scenarios. In *Optical Network Design and Modelling (ONDM)*.

IOSUP, A., DUMITRESCU, C., EPEMA, D., LI, H., AND WOLTERS, L. 2006. How are Real Grids Used? The Analysis of Four Grid Traces and Its Implications. In *IEEE/ACM International Conference on Grid Computing*.

IRISH, L., , AND CHRISTENSEN, K. 1998. A "Green TCP/IP" to Reduce Electricity Consumed by Computers. In *IEEE Southeast Conference*. 302–305.

JEJURIKAR, R. AND GUPTA, R. 2006. Energy aware task scheduling with task synchronization for embedded real-time systems. In *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*. 1024–1037.

JIMENO, M., CHRISTENSEN, K., AND NORDMAN, B. 2008. A Network Connection Proxy to Enable Hosts to Sleep and Save Energy. In *IEEE International Performance, Computing and Communications Conference (IPCCC)*. 101–110.

JUNG, G., JOSHI, K., HILTUNEN, M., SCHLICHTING, R., AND PU, C. 2009. A cost-sensitive adaptation engine for server consolidation of multitier applications. In *ACM/IFIP/USENIX International Conference on Middleware (Middleware)*. 1–20.

KALMAN, R. 1960. A New Approach to Linear Filtering and Prediction Problems. *Transactions of the ASME – Journal of Basic Engineering 82,* Series D, 35–45.

KALYVIANAKI, E., CHARALAMBOUS, T., AND HAND, S. 2009. Self-Adaptive and Self-Configured CPU Resource Provisioning for Virtualized Servers Using Kalman Filters. In *ACM International Conference on Autonomic Computing (ICAC)*. 117–126.

KANG, J. AND RANKA, S. 2010. Dynamic slack allocation algorithms for energy minimization on parallel machines. *Journal of Parallel and Distributed Computing 70*, 417–430.

Kappiah, N., Freeh, V., and Lowenthal, D. 2005. Just In Time Dynamic Voltage Scaling: Exploiting Inter-Node Slack to Save Energy in MPI Programs. In *ACM/IEEE Conference on Supercomputing (SC)*.

Kim, E., Link, G., Yum, K., Vijaykrishnan, N., Kandemir, M., Irwin, M., and Das, C. 2005. A holistic approach to designing energy-efficient cluster interconnects. *IEEE Transactions on Computers 54,* 6, 660–671.

Kim, M. and Noble, B. 2001. Mobile network estimation. In *ACM International Conference on Mobile Computing and Networking (MobiCom)*. 298–309.

Koomey, J. 2011. Growth in data center electricity use 2005 to 2010.

Kusic, D., Kephart, J., Hanson, J., Kandasamy, N., and Jiang, G. 2008. Power and performance management of virtualized computing environments via lookahead control. In *IEEE International Conference on Autonomic Computing (ICAC)*. 3–12.

Leangsuksun et al, C. 2006. IPMI-based Efficient Notification Framework for Large Scale Cluster Computing. In *International Symposium on Cluster Computing and the Grid Workshops (CCGrid)*.

Lefèvre, L. and Orgerie, A.-C. 2009. Towards Energy Aware Reservation Infrastructure for Large-Scale Experimental Distributed Systems. *Parallel Processing Letters 19,* 3, 419–433.

Lefèvre, L. and Orgerie, A.-C. 2010. Designing and Evaluating an Energy Efficient Cloud. *The Journal of SuperComputing 51,* 3, 352–373.

LessWatts 2010. Server power measurements. LessWatts report.

Lim, M., Freeh, V., and Lowenthal, D. 2006. Adaptive, Transparent Frequency and Voltage Scaling of Communication Phases in MPI Programs. In *ACM/IEEE Conference on Supercomputing (SC)*.

Liu, J., Zhao, F., Liu, X., and He, W. 2009. Challenges towards elastic power management in Internet data centers. In *IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW)*. 65–72.

Mahadevan, P., Sharma, P., Banerjee, S., and Ranganathan, P. 2009. A Power Benchmarking Framework For Network Devices. *Networking 5550*, 795–808.

Merkel, A. and Bellosa, F. 2006. Balancing power consumption in multiprocessor systems. *ACM SIGOPS Operating Systems Review 40,* 4, 403–414.

Mishra, R., Rastogi, N., Zhu, D., Mossé, D., and Melhem, R. 2003. Energy aware scheduling for distributed real-time systems. In *International Symposium on Parallel and Distributed Processing (IPDPS)*.

Miyoshi, A., Lefurgy, C., Van Hensbergen, E., Rajamony, R., and Rajkumar, R. 2002. Critical power slope: understanding the runtime effects of frequency scaling. In *ACM International conference on Supercomputing (ICS)*. 35–44.

Moore, J., Chase, J., Ranganathan, P., and Sharma, R. 2005. Making scheduling "cool": Temperature-aware workload placement in data centers. In *USENIX Annual Technical Conference (ATEC)*.

Nathuji, R. and Schwan, K. 2007. VirtualPower: coordinated power management in virtualized enterprise systems. In *ACM SIGOPS symposium on Operating systems principles (SOSP)*. 265–278.

Nedevschi, S., Popa, L., Iannaccone, G., Ratnasamy, S., and Wetherall, D. 2008. Reducing Network Energy Consumption Via Sleeping And Rate-Adaptation. In *USENIX Symposium On Networked Systems Design & Implementation (NSDI)*. 323–336.

Nordman, B. and Christensen, K. 2010. Proxying: The Next Step In Reducing IT Energy Use. *IEEE Computer 43,* 1, 91–93.

Nurmi, D., Wolski, R., Crzegorczyk, C., Obertelli, G., Soman, S., Youseff, L., and Zagorodnov, D. 2008. Eucalyptus: A Technical Report on an Elastic Utility Computing Architecture Linking Your Programs to Useful Systems. Technical report 2008-10, Department of Computer Science, University of California, Santa Barbara.

Odlyzko, A. 2003. Data Networks are Lightly Utilized, and will Stay that Way. *Review of Network Economics 2*, 210–237.

Orgerie, A.-C. and Lefèvre, L. 2011. Energy-Efficient Bandwidth Reservation for Bulk Data Transfers in Dedicated Wired Networks. *Journal of Supercomputing, Special Issue on Green Networks*.

Orgerie, A.-C., Lefèvre, L., and Gelas, J.-P. 2010. Demystifying Energy Consumption in Grids and Clouds. In *Work in Progress in Green Computing, IGCC Workshop*.

Patel, C., Sharma, R., Bash, C., and Graupner, S. 2002. Energy Aware Grid: Global Workload Placement based on Energy Efficiency. Tech. rep., HP Laboratories.

Patterson, M., Costello, D., Grimm, P., and Loeffler, M. 2007. Data center TCO: a comparison of high-density and low-density spaces. In *Thermal Challenges in Next Generation Electronic Systems (THERMES)*.

Pawlish, M. and Varde, A. 2010. Free cooling: A paradigm shift in data centers. In *International Conference on Information and Automation for Sustainability (ICIAFS)*. 347–352.

Restrepo, J., Gruber, C., and Machuca, C. 2009. Energy Profile Aware Routing. In *IEEE International Conference on Communications (ICC Workshops)*. 1–5.

Reviriego, P., Christensen, K., Rabanillo, J., and Maestro, J. 2011. An Initial Evaluation of Energy Efficient Ethernet. *IEEE Communications Letters 15,* 5, 578–580.

Reviriego, P., Maestro, J., Hernandndez, J., and Larrabeiti, D. 2010. Burst Transmission for Energy-Efficient Ethernet. *IEEE Internet Computing 14,* 4, 50–57.

Rivoire, S., Shah, M., Ranganathan, P., and Kozyrakis, C. 2007. JouleSort: a balanced energy-efficiency benchmark. In *ACM SIGMOD international conference on Management of data*. 365–376.

Roberts, L. 2009. A radical new router. *IEEE Spectrum 46,* 7, 34–39.

Sabhanatarajan, K., Gordon-Ross, A., Oden, M., Navada, M., and George, A. 2008. Smart-NICs: Power Proxying For Reduced Power Consumption In Network Edge Devices. In *IEEE Computer Society Annual Symposium On VLSI (ISVLSI)*. 75–80.

Shang, L., Peh, L.-S., and Jha, N. 2006. PowerHerd: a distributed scheme for dynamically satisfying peak-power constraints in interconnection networks. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems 25,* 1, 92–110.

Sharma, R., Bash, C., Patel, C., Friedrich, R., and Chase, J. 2005. Balance of Power: Dynamic Thermal Management for Internet Data Centers. *IEEE Internet Computing 9,* 1, 42–49.

Shin, D. and Kim, J. 2004. Dynamic voltage scaling of periodic and aperiodic tasks in priority-driven systems. In *Asia and South Pacific Design Automation Conference (ASP-DAC)*. 653–658.

Singh, T. and Vara, P. 2009. Smart metering the clouds. In *IEEE International Workshops on Enabling Technologies*. 66–71.

Snowdon, D., Ruocco, S., and Heiser, G. 2005. Power management and dynamic voltage scaling: Myths and facts. In *Workshop on Power Aware Real-time Computing*.

Soteriou, V., Eisley, N., and Peh, L.-S. 2005. Software-Directed Power-Aware Interconnection Networks. In *International Conference On Compilers, Architectures And Synthesis For Embedded Systems (CASES)*. 274–285.

Soteriou, V. and Peh, L.-S. 2003. Dynamic Power Management For Power Optimization Of Interconnection Networks Using On/Off Links. In *Symposium On High Performance Interconnects*.

Srikantaiah, S., Kansal, A., and Zhao, F. 2008. Energy Aware Consolidation for Cloud Computing. In *Workshop on Power Aware Computing and Systems (HotPower)*.

Steele, J. 1998. ACPI thermal sensing and control in the PC. In *Wescon*.

Steinder, M., Whalley, I., Hanson, J., and Kephart, J. 2008. Coordinated management of power usage and runtime performance. In *IEEE Network Operations and Management Symposium (NOMS)*. 387–394.

Stoess, J., Lang, C., and Bellosa, F. 2007. Energy management for hypervisor-based virtual machines. In *USENIX Annual Technical Conference (ATC)*. 1–14.

Subramaniam, B. and Feng, W.-C. 2010. Statistical power and performance modeling for optimizing the energy efficiency of scientific computing. In *IEEE/ACM International Conference on Green Computing and Communications (GreenCom)*. 139–146.

Subramanyam, S., Smith, R., van den Bogaard, P., and Zhang, A. 2009. Deploying Web 2.0 Applications on Sun Servers and the Opensolaris Operating System. Sun BluePrints 820-7729-10, Sun Microsystems.

Talaber, R., Brey, T., and Lamers, L. 2009. Using Virtualization to Improve Data Center Efficiency. Tech. rep., The Green Grid.

Tatezono, M., Maruyama, N., and Matsuoka, S. 2006. Making wide-area, multi-site MPI feasible using Xen VM. In *Workshop on Frontiers of High Performance Computing and Networking (in conjunction with ISPA)*. LNCS, vol. 4331. Springer, 387–396.

Torres, J., Carrera, D., Hogan, K., Gavalda, R., Beltran, V., and Poggi, N. 2008. Reducing wasted resources to help achieve green data centers. In *IEEE International Symposium on Parallel and Distributed Processing (IPDPS)*. 1–8.

Travostino, F., Daspit, P., Gommans, L., Jog, C., de Laat, C., Mambretti, J., Monga, I., van Oudenaarde, B., Raghunath, S., and Wang, P. Y. 2006. Seamless live migration of virtual machines over the man/wan. *Future Generation Computer Systems 22,* 8, 901–907.

Tucker, R. 2010. Green Optical Communications - Part II: Energy Limitations in Networks. *IEEE Journal of Selected Topics in Quantum Electronics, Special Issue on Green Photonics 17*, 261–274.

Tzanakaki, A., Anastasopoulos, M., Georgakilas, K., Buysse, J., Leenheer, M. D., Develder, C., Peng, S., Nejabati, R., Escalona, E., Simeonidou, D., Ciulli, N., Landi, G., Brogle, M., Manfredi, A., Lopez, E., Riera, J., Garcia-Espin, J., Donaldio, P., Parladori, G., and Jimenez, J. 2011. Energy Efficiency in integrated IT and Optical Network Infrastructures: The GEYSERS approach. In *Workshop on Green Communications and Networking (in conjunction with INFOCOM)*.

Urgaonkar, B., Shenoy, P., Chandra, A., Goyal, P., and Wood, T. 2008. Agile dynamic provisioning of multi-tier internet applications. *ACM Transactions on Autonomous and Adaptive Systems 3,* 1, 1–39.

Van Heddeghem, W., De Groote, M., Vereecken, W., Colle, D., Pickavet, M., and Demeester, P. 2010. Energy-efficiency in telecommunications networks: Link-by-link versus end-to-end grooming. In *Conference on Optical Network Design and Modeling (ONDM)*.

Verma, A., Ahuja, P., and Neogi, A. 2008. pMapper: Power and Migration Cost Aware Application Placement in Virtualized Systems. In *ACM/IFIP/USENIX International Middleware Conference (Middleware)*. 243–264.

Voorsluys, W., Broberg, J., Venugopal, S., and Buyya, R. 2009. Cost of Virtual Machine Live Migration in Clouds: A Performance Evaluation. Tech. rep., Clouds Laboratory, University of Melbourne, Australia.

Wang, B. and Singh, S. 2004. Computational Energy Cost Of TCP. In *IEEE International Conference On Computer Communications (INFOCOM)*. 206–216.

Wang, H.-S., Peh, L.-S., and Malik, S. 2002. A Power Model for Routers: Modeling Alpha 21364 and InfiniBand Routers. In *Symposium on High-Performance Interconnects*. 21–27.

Wang, L., von Laszewski, G., Dayal, J., and Wang, F. 2010. Towards Energy Aware Scheduling for Precedence Constrained Parallel Tasks in a Cluster with DVFS. In *Conference on Cluster, Cloud and Grid Computing (CCGrid)*. 368–377.

Wang, Y., Keller, E., Biskeborn, B., van der Merwe, J., and Rexford, J. 2008. Virtual routers on the move: live router migration as a network-management primitive. *ACM SIGCOMM Computer Communication Review 38,* 4, 231–242.

Yamanaka, N., Shimizu, S., and Shan, G. 2010. Energy efficient network design tool for green IP/Ethernet networks. In *Conference on Optical Network Design and Modeling (ONDM)*.

Yan, L., Zhong, L., and Jha, N. 2005. User-perceived latency driven voltage scaling for interactive applications. In *Design Automation Conference (DAC)*. 624–627.

Yang, C.-Y., Chen, J.-J., Kuo, T.-W., and Thiele, L. 2009. An approximation scheme for energy-efficient scheduling of real-time tasks in heterogeneous multiprocessor systems. In *Design, Automation Test in Europe Conference Exhibition (DATE)*. 694–699.

Yao, F., Demers, A., and Shenker, S. 1995. A scheduling model for reduced CPU energy. In *Annual Symposium on Foundations of Computer Science (FOCS)*. 374–382.

Ye, T. T., Micheli, G., and Benini, L. 2002. Analysis of power consumption on switch fabrics in network routers. In *ACM Design Automation Conference (DAC)*. 524–529.

Yeo, S. and Lee, H. 2011. Mathematical modeling of performance and utility consumption for a heterogeneous cloud computing environment. *Computer PP,* 99.

Zhang, B., Sabhanatarajan, K., Gordon-Ross, A., and George, A. 2008. Real-Time Performance Analysis Of Adaptive Link Rate. In *IEEE Conference On Local Computer Networks (LCN)*. 282–288.

Zhang, M., Yi, C., Liu, B., and Zhang, B. 2010. GreenTE: Power-aware traffic engineering. In *IEEE International Conference on Network Protocols (ICNP)*. 21–30.

Zhang, S., Shen, D., and Chan, C.-K. 2010. Energy efficient time-aware traffic grooming in wavelength routing networks. In *IEEE Global Telecommunications Conference (GLOBECOM)*.

Zhang, Y., Chowdhury, P., Tornatore, M., and Mukherjee, B. 2010. Energy Efficiency in Telecom Optical Networks. *IEEE Communications Surveys Tutorials 12,* 4, 441–458.

Zhuo, J. and Chakrabarti, C. 2008. Energy-efficient dynamic task scheduling algorithms for DVS systems. *ACM Transactions on Embedded Computing Systems 7*, 17:1–17:25.

Received September 2011;