Reducing the energy consumption of large scale computing systems through combined shutdown policies with multiple constraints

Anne Benoit[1], Laurent Lefèvre[1], Anne-Cécile Orgerie[2], and Issam Raïs[1]

[1]Univ. Lyon, Inria, CNRS, ENS de Lyon, Univ. Claude-Bernard Lyon 1, LIP
[2]CNRS, IRISA, Rennes, France
Email: Anne.Benoit@ens-lyon.fr, Laurent.Lefevre@inria.fr, Anne-Cecile.Orgerie@irisa.fr, Issam.Rais@inria.fr

Abstract

Large scale distributed systems (high performance computing centers, networks, data centers) are expected to consume huge amounts of energy. In order to address this issue, shutdown policies constitute an appealing approach able to dynamically adapt the resource set to the actual workload. However, multiple constraints have to be taken into account for such policies to be applied on real infrastructures: the time and energy cost of switching on and off, the power and energy consumption bounds caused by the electricity grid or the cooling system, and the availability of renewable energy. In this paper, we propose models translating these various constraints into different shutdown policies that can be combined for a multi-constraint purpose. Our models and their combinations are validated through simulations on a real workload trace.

1 Introduction

Reducing the energy consumption of large scale distributed systems (high performance computing centers, networks, data centers) is a mandatory step to address in order to build a sustainable digital society.

Since more than a decade, several technological solutions have been proposed by system designers in order to help reducing electrical power consumption, as for instance shutdown and slowdown approaches. The first and most explored solution consists in turning on and off some resources depending on platform usage. Several works that studied the energy-related impacts of shutdown techniques did not considered any transition cost for switching between on and off states, but they nonetheless showed the potential impact of such techniques. Yet, aggressive shutdown policies are not always the best solution to save energy [1].

Moreover, supporting on and off of large amount of resources can be risky as it impacts the whole infrastructure of supercomputers (electricity provision, cooling systems, etc.). Resource providers and managers can be human who are responsible of the administration of large supercomputers, but they can also be software components that deal with resources (schedulers, resource management frameworks, etc.). This paper addresses the question on how resource providers and managers can be helped to validate their constraints concerning the turning on and off of large amount of physical computing, storage and networking resources.

Nowadays, hardware components of a datacenter or supercomputer (servers, network switches, data storage, etc.) are not yet energy proportional. In fact, the static part (i.e., the part that does not vary with workload) of the energy consumed, for instance by computing units, represents a high part of the overall energy consumed by the nodes. Therefore, shutting unused physical resources that are idle and not expected to be used in a predicted duration could lead to non negligible energy savings. This paper focuses on turning on and off any kind of resources such as servers, network equipment, memory banks, cores, etc. In the context of this paper, the proposed models and validations will focus on servers (called *nodes*).

Shutdown seems to be an interesting leverage to save energy, but this technique cannot be applied at large scale if no constraint is respected on the target system. This is especially true if three types of constraints

are taken into account: the cost of shutdown and wake-up (in terms of time and energy), electric and thermal constraints imposed to the whole infrastructure. We can see the datacenter as a composition of *IT machines and cooling system, communicating with an electrical provider to deal with various electric related constraints.* Actually, turning off too many nodes could cause the temperature to be too cool and the power used to be under the minimum power capping negotiated with the electrical provider. Likewise, if too many nodes are turned on, and if the energy consumed during shutdown and wake-up sequences is taken into account (which is far from being free), limits fixed by the power provider can greatly be overcome and at the same time, could cause the temperature to raise drastically, creating hotspots. If such constraints are not taken into account, they can put into danger machines composing the studied computing facility.

The proposed solutions in this paper aim at:

- Modeling the shutdown leverage that can be used under actual and future supercomputer constraints;
- Taking into account the impact of On→Off (from on to off state, corresponding to a shutdown operation) and Off→On (from off to on state, corresponding to a wake-up operation) sequences in terms of time, power and energy;
- Taking into account idle and off states observed after such sequences, since they deeply impact the electrical usage of resources;
- Allowing a mono or combined usage of models in order to help resource managers and providers to respect several constraints at the same time.

This paper explores the modeling of several shutdown policies that can be handled by resource providers and that deal with infrastructure constraints:

- The *basic models* allow comparisons with several related works where turning on and off can be free and immediate.
- The *sequence-aware models* focus on the On→Off sequences when providers want to switch off several useless resources and to switch them on again when these resources are needed. These models deal with the availability of scheduling On→Off sequences during gaps and their potential energy benefits.
- The *electricity-aware models* deal with the electrical provision of supercomputers in order to avoid large-scale aggressive electrical demands (due to massive switch on of resources) and to respect power capping requirements.
- The *cooling-aware models* respect the constraints imposed by the cooling infrastructure associated with the supercomputer. They follow the thermal constraints of the system by reducing the number of possible On→Off sequences.
- The *renewable-energy-aware models* support selective shutdown policies by considering the electricity provenance (from renewable energy or from fossil-based energy sources).

The proposed models are described one by one, and their combined usage is illustrated. Such models are validated through simulation on real trace log usage.

The paper is organized as follows. Section 2 presents related work dealing with shutdown policies and technologies at large scale. Section 3 presents the modeling of the various shutdown (On→Off) policies for all models introduced above, and it explains how the models can be used and combined. The experimental setup is described in Section 4, and experimental results are analyzed in Section 5. Finally, Section 6 concludes and presents some future works.

2 Related work

Pioneering work on studying the energy-related impacts of shutdown techniques started in 2001 [2, 3]. These early works did not consider any transition cost for switching between on and off, but they nonetheless showed the potential impact of such techniques. Demaine *et al.* examine the power minimization problem where the objective is to minimize the total transition costs plus the total time spent in the active state [4]. They develop a $(1 + 2\alpha)$-approximation algorithm, with α the transition cost.

However, the parameters considered for this transition cost highly vary across the literature. Gandhi *et al.* take into account the energy cost of switching on servers (no switching off cost as it is estimated to be

negligible in comparison with the switching on cost) [5]. This energy cost is assumed to be equal to the transition time multiplied by the power consumption while in the on state. Lin *et al.* take into account the energy used for the transition, the delay in migrating connections or data, the increased wear-and-tear on the servers, and the risk associated with server toggling [6].

Off-the-shelf hardware is nowadays integrating shutdown policies. Data center resource managers propose techniques or hooks to configure such capabilities. For example, slurm [7], an open-source cluster management system, introduces a *SuspendTime*[1] that represents the minimum idle time after which it allows the node to be switched off. Then, the resource manager is responsible for deciding when to switch on and off servers. It takes decisions either based on pre-determined policy [7], on workload predictions [8], on queuing models [5] or on control theory approach [9].

Shutdown policies are often combined with consolidation algorithms that gather the load on a few servers to favor the shutdown of the others. Employing either reactive or proactive scheduling options [10, 11], consolidation algorithms increase the energy gains brought by shutdown techniques at a cost of a trade-off with performance [12]. The rich diversity in power management techniques and levers can lead to problems if they are not coordinated at the data center level [13]. In this paper, we study shutdown policies (i.e., when to switch off), without combining them to scheduling algorithms and consolidation approaches in order to evaluate the impacts of such policies without interfering with the workload of real platforms and with the users' expected performance.

Shutdown techniques do not only impact energy consumption, they also affect temperature and consequently cooling systems [14]. They can also be used for limiting the dark silicon effect, i.e., the underutilization of the device integration capacity due to power and temperature effects [15]. This issue has lead to the introduction of user-specified, dynamic, hardware-enforced processor power bounds, as for the Intel's Sandy Bridge family of processors for instance [16]. At a data center level, it translates into power budgeting, where the total power budget is partitioned among the cooling and computing units, and where the cooling power has to be sufficient to extract the heat of the computing power. Given the computing power budget, Zhan *et al.* propose an optimal computing budgeting technique based on knapsack-solving algorithms to determine the power caps for the individual servers [17].

3 Models

In the context of this paper, the proposed models focus on servers (also called nodes). In this section, we first show in Section 3.1 how to characterize the impact of turning on or off a node, as an illustration, in terms of time and power consumption. Next, we provide some definitions in Section 3.2, where we define in particular the reachable states for a node. The core of the section is the definition of the models (Section 3.3), where we introduce all the models that act on node states. We finally explain how to combine the models in Section 3.4.

3.1 Sequence definitions: example of Off→On sequences for nodes

For node i, $Seq_i = \{(t_0; AvgP_0), \ldots, (t_n; AvgP_n)\}$ is the set of timestamps and average power consumption measurements of an Off→On or On→Off sequence, where t_0 and t_n represent the starting and ending time respectively of sequence Seq_i on node i. The length of the sequence is therefore $t_n - t_0$, and at timestamp t_k ($1 \leq k \leq n$), $AvgP_k$ is the average power consumption of node i.

To monitor such sequences, we use an external power monitoring allowing us to trace power consumption of nodes at a rate of one averaged power value per second. Figure 1 illustrates the boot sequence (or Off→On sequence) on Linux as Operating System, widely used on supercomputer infrastructures. First of all, power is supplied to SMPS (Switched-Mode Power Supply), which converts AC to DC. The BIOS (Basic Input Output System) is bootstrapped and launches POST (Power on Self Test), a series of tests by the BIOS, that checks the proper functioning of different hardware components. Then MBR (Master Boot Record), the first or last bytes of the disk, is loaded. MBR permits to launch GRUB (Grand Unified Bootloader),

[1] `http://slurm.schedmd.com/power_save.html`

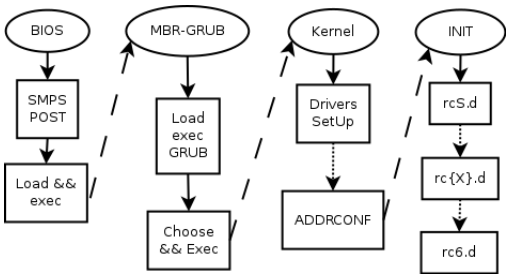

Figure 1: Linux monitored boot sequence.

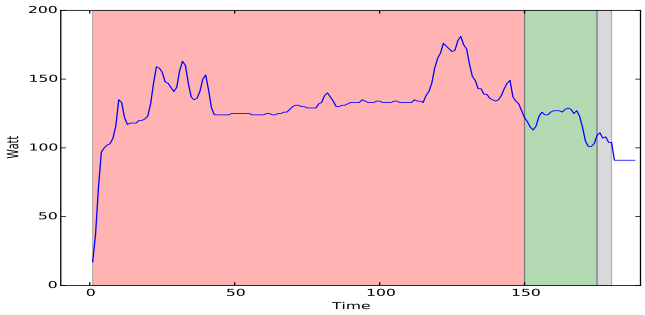

Figure 2: Averaged monitored Off→On sequence of a Taurus node running Linux : BIOS-MBR-GRUB sequence in red; Kernel in green; Init in gray (average of 50 runs).

which is responsible for choosing the kernel to be launched. INIT is the first executed process. It is in charge of running all runlevels ("/etc/rcX.d/").

We monitor the boot sequence (wake-up operation, Off→On) to detect when each event happens. Unfortunately, no information can be extracted between BIOS and GRUB operations. The first event that can be monitored in this sequence is the Kernel launch; this is the main reason of the aggregated sections of BIOS-MBR-GRUB in Figure 2, which shows how the power evolves with time during a monitored boot sequence, on a Taurus node (from Grid5000 experimental platform, the node characteristics are presented in Table 3, Section 4). Each value is an average over 50 runs.

We get the time where kernel starts with the "dmesg" tool (which is a logging of what happened during the launch of the kernel). The INIT monitoring is made by modifying the runlevel script.

Next, we detail the set of possible states accounting for these Off→On and On→Off sequences.

3.2 Definitions: states

Since we wish to account for Off→On and On→Off sequences, we partition the devices took into account, here only nodes, into four distinct sets as illustrated in Figure 3:

- ON in progress: Set of nodes in the Off→On sequence;
- ON: Set of nodes turned on, able to receive computation. This state is divided into two sub-states: Idle and Run;
- OFF in progress: Set of nodes in the On→Off sequence;
- OFF: Set of nodes turned off, unable to receive computation.

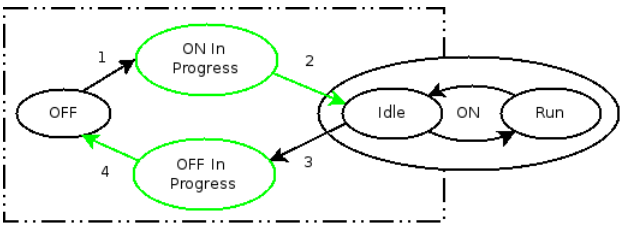

Figure 3: States and transitions during various sequences.

Furthermore, we denote by ALL the set of all nodes. We define our action scope only on the Idle state, i.e., nodes that are turned on but not currently computing. Nodes on the Run state (i.e., currently computing) are not on the action scope of this model, since it is rather the scope of the scheduler to decide to stop computation and turn nodes to the Idle state. We need however to be aware of the nodes in the Run state

since we aim at ensuring a global power capping. Thus, a node goes from the *ON in progress* state to the ON state through the Idle state, and it can leave the ON state only when it is in the Idle state.

On Figure 3, the dotted line square therefore represents the scope of the models described here. We aim at allowing a set of nodes to switch from one state to another, by taking one of the four numbered transitions. Transitions 2 and 4 are automatically taken at the end of the ON in progress or OFF in progress states, while we may decide to trigger transitions 1 or 3.

A node in the ON in progress state could be in several sub-states, according to the Linux boot sequence: BIOS-MBR-GRUB, Kernel, Init, or whatever boot up sequence is defined on the node. We consider other states as atomic.

We use the models as follows: we decide what can be done at the current time-step T_c, knowing that there is an idle interval of length T_{gap} on a given node. In our case, the model decides whether the node should be shut down, given the enforced constraints.

3.3 Model definitions

In this section, we derive several models, assuming that we have some knowledge about the node reservations, i.e., for each node, we have a list of intervals during which the node is in the idle state, and we aim at deciding whether this node can be turned off and then back on, while respecting the constraints of the system and improving the goals. These are model-dependent and are detailed in the next sections.

We therefore provide an entity giving advice on changing the state of a (set of) node, making sure that the overall system responds to the described constraints. This entity is called an actor, and it is acting on the OnOff leverage.

3.3.1 Basic models

Two basic models are used by most papers in the literature (see Section 2): either the nodes are never shut down (No-OnOff model), or there is no cost (time, energy, thermal) to turn on or off a node (LB-ZeroCost-OnOff model: *Lower Bound Zero Cost OnOff Model*), making it very simple to shut-down a node (but very far from reality). In this context, the node consumes nothing when executing an On→Off or Off→On sequence. Thus, there is no cost nor time spent to switch state, and no power peak observed during the sequence. Therefore, switching on or off nodes has no impact on the system. This LB-ZeroCost-OnOff model hence provides a theoretical lower bound on the gains that can be achieved by shutting down nodes.

3.3.2 Sequence-aware models

The sequence-aware models make sure that the sequence observed on a node or set of nodes during On→Off or Off→On sequences does not overcome the fixed constraints (time, energy, etc). Therefore, we need to record a few data for every node composing the studied case, in particular a record of the Off→On sequence and of the On→Off sequence.

Time constrained The first model, Seq-Aw-T (*Sequence-Aware Time*), checks whether there is enough time to perform an On→Off followed by an Off→On sequence on a node, given the available time slot where the node is idle. Let T_{gap} be the size of the "gap", i.e., the interval of idle time of the node. Then, Seq-Aw-T allows us to turn off the node in this time slot if and only if $T_{\text{OnOff}} + T_{\text{OffOn}} \leq T_{\text{gap}}$, where T_{OnOff} (resp. T_{OffOn}) is the time spent by the node during an On→Off (resp. Off→On) sequence.

Energy constrained The Seq-Aw-E model (*Sequence-Aware Energy*) further refines Seq-Aw-T by checking whether turning off the node is beneficial in terms of energy. The minimum time T_s of the gap is now further constrained by the energy savings:

$$T_s = \max\left(T_{\text{OnOff}} + T_{\text{OffOn}}, \frac{E_{\text{OnOff}} + E_{\text{OffOn}} - P_{\text{off}}(T_{\text{OnOff}} + T_{\text{OffOn}})}{P_{\text{idle}} - P_{\text{off}}}\right),$$

where:

- P_{idle} is the power consumption when the node is in the Idle state (unused, but powered on);
- P_{off} is the power consumption when the node is switched off (typically not null and lower than P_{idle});
- E_{OnOff} is the energy consumed during the On→Off sequence;
- E_{OffOn} is the energy consumed during the Off→On sequence.

The first term states, as for SEQ-AW-T, that we need at least a time $T_{\text{OnOff}} + T_{\text{OffOn}}$ to turn off the node (and back on) during the idle interval, and it determines when it is acceptable to do this. The second term ensures that there will be gains in energy: the energy saved by running at P_{off} rather than P_{idle} is $T_s(P_{\text{idle}} - P_{\text{off}})$ during the interval, but the additional energy due to the On→Off and Off→On sequences is $E_{\text{OnOff}} + E_{\text{OffOn}} - P_{\text{off}}(T_{\text{OnOff}} + T_{\text{OffOn}})$. Therefore, if $T_s < T_{\text{gap}}$, where T_{gap} is the size of the "gap", i.e., the interval of idle time of the node, then it is beneficial to turn off (at the beginning of the gap) then on (at the end of the gap) the node, in terms of energy consumption.

3.3.3 Electricity-aware model

The electricity-aware model, ELEC-SF (*Electrical Scalability Factor*), aims at ensuring the safety of the computing facility through its electrical provisioning, given that the following information is provided: how much Watts could be added (ESF-Up) or retrieved (ESF-Down) in the facility in a given duration? We call this the electrical scalability factor (ESF). For instance, between 0 W and $1,000$ W of power usage of IT equipment (W_{IT}), ten Watts can be added in the facility overall usage during one second, as illustrated in Table 1.

From this information, we can define the function $electricalScalabilityFactor(X)$, which returns true if the addition or removal of all nodes in set X will be supported energetically by the infrastructure and the electrical provider, i.e., if the ESF is respected.

In this context, the model allows us to turn off and then on nodes during an idle interval if and only if the global ESF is respected for all nodes at the time of the On→Off and Off→On sequences.

For IT power (W_{IT}) between	ESF_{Up}
$0\ W \rightarrow 1,000\ W$	$+10\ W$ during $1s$
$1,000\ W \rightarrow 10,000\ W$	$+50\ W$ during $1s$
$10,000\ W \rightarrow 100,000\ W$	$+100\ W$ during $1s$

Table 1: Electrical Scalability Factor illustration.

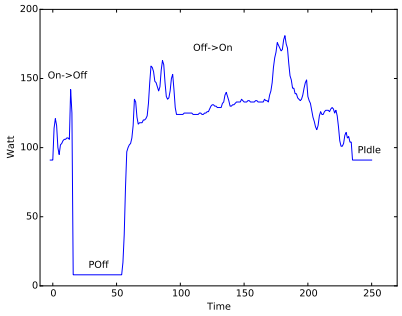

Figure 4: On→Off followed by P_{off}, Off→On, and finally P_{idle} section for Taurus calibrated node (average of 50 runs).

3.3.4 Power-capping-aware model

The POWER-CAP model (*Power-Capping-Aware*) aims at maintaining an average power budget and guaranteeing minimal or maximal electrical power consumption. Indeed, turning on and off components could lead to hard power capping disruption. Figure 4 shows a set of consecutive sequences: an On→Off sequence, a section in the OFF state, an Off→On sequence, and finally a section in the idle state. These experiments represent the shutdown and boot of a node during a gap (i.e., idle interval in the schedule). All the previous actions energetically stress the node, whether it is in an upper or lower way.

Here, we provide some information about the power capping that should be done.

A minimum power capping ($POWERCAP\_Min$) represents a constraint set by the electrical supplier, and it is defined by providing a lower bound on power. A maximum power capping ($POWERCAP\_Max$)

represents power limit fixed by the electrical provider, and it is defined by an upper bound on power. These minimum and maximum power capping values may be a function of the time, i.e., the requirements may change in time.

We introduce the function $PowerSum(X)$, which returns the sum of the power consumed by nodes in X.

We can turn off or on nodes in set X if and only if $PowerSum(ALL) \geq POWERCAP\_Min$ and $PowerSum(ALL) \leq POWERCAP\_Max$ at all time during the sequence.

3.3.5 Cooling system-aware model

The Cool-Aw model (*Cooling System-Aware*) accounts for the cooling system in use. Therefore, we need to record some basic information about the chosen cooling system: the instantaneous needed IT power (W_{IT}), and the Cooling Scalability Factor (CSF_{up} and CSF_{down}) for every level of cooling system, similar to the electrical scalability factor defined in Section 3.3.4.

We make the assumption that the cooling system has several working *levels*. Thus, several power levels for cooling are available in function of the IT power needed by the cooling system. For instance, between $0\ W$ and $1,000\ W$ of power usage of IT equipment, one Watt can be added in the facility overall usage during one second, as illustrated for CSF_{up} in Table 2.

3.3.6 Renewable energy-aware model

The last defined model, Renew-E (*Renewable Energy-Aware*), assumes that we have the knowledge of the provenance of energy (green or brown) at actual time and near future (predicted). Green energy is provided with specific sources (sun, wind, etc.), while brown energy is mainly provided with fossil materials (coal, oil, etc.). The aim of this model is to minimize the usage of brown energy, and hence to turn off nodes when some brown energy can be saved. We assume that the green energy production is done on-site, for instance through photovoltaic panels. Furthermore, the datacenter does not sell its generated produced green energy. Therefore, no gain can be obtained by turning off nodes when using the green energy. A consequence of this strategy is that it will reduce the number of On→Off sequences for a same waste of usable energy.

At time t, $EnergyProv(Src, t, X)$ checks if the provenance of energy on node X is Src, where Src can be G (for green) or B (for brown). Then, at the beginning of an idle interval (time-step t_{start}, interval of duration T_{gap}), we check whether there exists a time-step t such that $t_{start} < t < t_{start+}T_{\text{gap}}$, and $EnergyProv(B, t, X)$ is true. If this is the case, then we turn off the node at time-step t_{start}.

3.4 Combining the models

The proposed models can be implemented through several software components and organized in a "workflow" of pipelined components. When an On→Off possibility happens in the system, due to a gap in activity, this possibility is analyzed by each model one by one. If each model gives an acceptance due to the observed constraints, the On→Off sequence can be scheduled. We provide an example of combination of models in Figure 5. It works as follows for a given idle interval on a node:

- Seq-Aw-T advises the provider whether there is time to turn off the node and then back on before it is in use again;
- Renew-E may tell the provider to turn off the node, because the current energy source is brown;
- Cool-Aw may prevent the provider to turn off the node if it would stress it too much in terms of temperature;

For IT power (W_{IT}) between	CSF_{up}	Level
$0\ W \rightarrow 1,000\ W$	$+1\ W$ during $1s$	1
$1,000\ W \rightarrow 10,000\ W$	$+10\ W$ during $1s$	2
$10,000\ W \rightarrow 100,000\ W$	$+100\ W$ during $1s$	3

Table 2: Cooling Scalability Factor illustration.

- Finally, ELEC-SF may prevent the provider to turn off the node if it would stress it too much in terms of electric power.

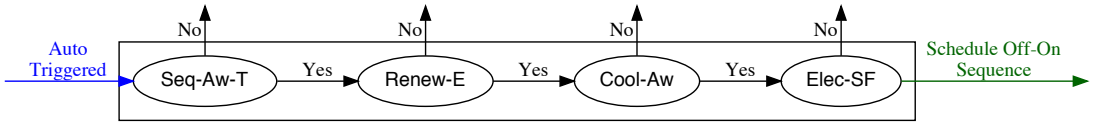

Figure 5: Example of auto-triggered model combination.

From the described combination of models in Figure 5, it is easy to understand that a lot of possibilities are "consistent" through the usage of the described models, in their scope, their concerns, their combinations and combination policies.

4 Experimental setup

To instantiate our models in various configurations, we developed a simulator capable of replaying a real datacenter trace, with real calibrations of nodes and jobs (time, power, energy).

4.1 Calibration of jobs and nodes

Grid'5000[18], a large-scale and versatile testbed for experiment-driven research in all areas of computer science, was used as a testbed. Grid'5000 deploys clusters linked with dedicated high performance networks on several cities in France (Grenoble, Lille, Lyon, Nancy, Nice, Nantes, Rennes). On the Lyon site, the energy consumption of all nodes from all available clusters (nova, orion, sagittaire, taurus) is monitored through a dedicated wattmeter, exposing one power measurement (Watt) per second with a 0.125 Watts accuracy per node. Therefore, we can obtain detailed traces giving the energy consumption of jobs at any time step, and we extract an average power consumption of each job. Thanks to these traces, we are able to replay in a realistic way the jobs and to simulate their corresponding energy consumption.

This testbed also provides management tools like kapower3[2], a utility that allows a user to have control on the power status of a reserved node. We monitored Taurus nodes to calibrate in time, energy and power the Off→On and On→Off sequences, as explained in Section 3.1; the results are detailed in Table **??**

Taurus	Features		Parameters	
Server model	Dell PowerEdge R720		E_{OffOn} (Joules)	23683
CPU model	Intel Xeon E5-2630		T_{OffOn} (seconds)	182
Number of CPU	2		E_{OnOff} (Joules)	1655
Cores per CPU	6		T_{OnOff} (seconds)	15
Memory (GB)	32		P_{idle} (Watts)	91
Storage (GB)	2 x 300 (HDD)		P_{off} (Watts)	8
			T_s (seconds)	286.29

Table 3: Calibration node characteristics and energy parameters for On→Off and Off→On sequences (average on 50 experimental measurements).

[2]kapower3 is a tool from the kadeploy3 software suite: `http://kadeploy3.gforge.inria.fr.`

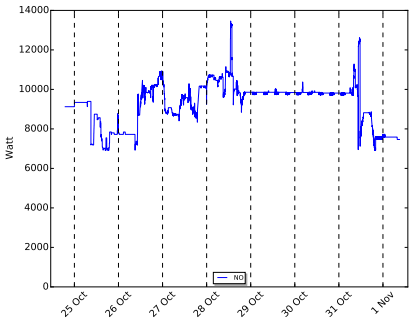

Figure 6: Trace replay with No-OnOff (NO).

Day	#Jobs	Average job power cons. (W)	Average job size (s)
Oct. 24 (7PM to 12AM)	33	157.91	50,401.24
Oct. 25 (Full day)	144	155.08	23,002.74
Oct. 26 (Full day)	277	159.79	12,299.06
Oct. 27 (Full day)	353	154.11	13,819.43
Oct. 28 (Full day)	318	159.96	27,286.17
Oct. 29 (Full day)	171	174.11	41,525.71
Oct. 30 (Full day)	180	174.04	39,453.67
Oct. 31 (Full day)	563	173.39	12,821.24
Nov. 1 (12AM to 8AM)	48	179.25	17,179.17

Table 4: Grid5000 trace statistics.

4.2 Trace and simulation

For our evaluation, we extracted the real workload usage of the Grid'5000 Lyon site from October 24, 2016 to November 1, 2016, thus representing approximately one week of resource utilization on this site.

The trace only contains nodes that were used during this period, which is up to 76. We consider that all nodes in the trace have similar P_{idle}, P_{off}, Off→On and On→Off sequences. This study is focused on shutting down nodes, thus we consider that the scheduled jobs cannot be moved.

We always combine a simulation of Seq-Aw-T with all other models in order to allow a correct execution of the models when an On→Off followed by an Off→On sequence should occur. Therefore, the evaluation of models can be applied on the same workload.

Figure 6 represents the profile of accumulated power consumption of nodes in Lyon for the extracted trace replayed with our previously exposed hypothesis. Table 4 presents statistics for this trace, day by day in various points of interest: number of jobs, average job consumption, and average job size. This week was chosen because of its representativeness of the workload variability that we observed overall on this platform by looking at larger traces. Indeed, for this week, the power consumption trace exhibits important peaks (Oct. 28), short peaks (Oct. 31), short climbs (Oct. 24 to 25), important climbs (Oct. 26) and sustained stable sections (Oct. 29), as shown in Figure **??** Variability of the trace can also be witnessed in Table 4 for various usages either concerning number of jobs (for instance, the differences between Oct. 25 and 28), average job consumption (for instance, Oct. 27 vs 31) or average job size variability (for instance, Oct. 26 vs 29) witnessed from one day to another.

The following section presents the results of the simulator on the extracted trace with calibration of Taurus node while applying previously defined models. Note that we always combine the models (except No-OnOff and LB-ZeroCost-OnOff) with Seq-Aw-T to ensure that the node is in the On state when it starts computing on the trace (and hence that we decide to turn off the node only if it can be turned on before the end of the interval).

5 Experimental validations

This section presents the results of simulation for all the models in the previously exposed experimental setup (Section 4). All graphs in this section represent a trace replay for one or multiple combined models with specific inputs. Table 5 presents the energy consumption in Joules of all models in the figures included in this section.

5.1 Sequence-aware models: Seq-Aw-T and Seq-Aw-E

Figure 7 shows results of No-OnOff, Seq-Aw-T, Seq-Aw-E, and LB-ZeroCost-OnOff. Between the two sequence-aware models, we can witness minor differences on the complete replay, for instance on Oct.

Model	Total energy consumed (Joules)	# On→Off & Off→On	% Saved
No-OnOff	6,083,698,688	0	0,0
LB-ZeroCost-OnOff	3,983,408,384	1794	34.52
Seq-Aw-T	4,015,736,064	964	33.99
Seq-Aw-E	4,015,201,024	844	34.00
Elec-SF *max*	4,611,556,352	819	24.19
Elec-SF *max*/2	5,078,084,608	767	16.53
Elec-SF *max*/4	5,461,449,728	647	10.22
Elec-SF *max*/8	5,828,239,360	451	4.19
Power-Cap 2000 min	4,401,067,520	855	27.65
Power-Cap 4000 min	4,593,668,096	761	24.49
Power-Cap 6000 min	5,059,857,408	617	16.82
Renew-E	4,132,427,520	423	32.07
Cool-Aw split 2	4,927,842,304	851	18.99
Cool-Aw split 7	5,054,783,488	831	16.91
All	5,386,375,168	315	11.46

Table 5: Trace replay's energy consumption (in Joules), number of (On→Off, Off→On) sequences added with models, and percentage of energy saved compared to No-OnOff.

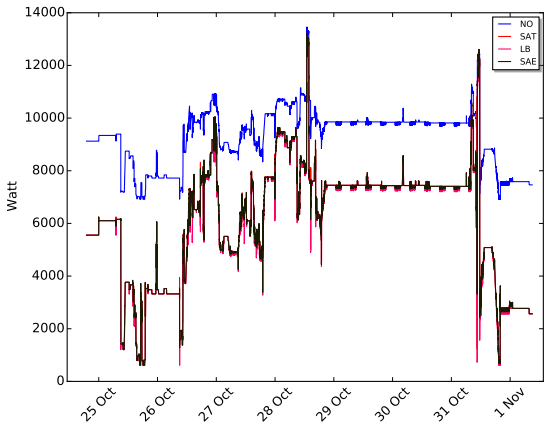

Figure 7: Trace replay No-OnOff (NO), Seq-Aw-T (SAT), Seq-Aw-E (SAE) and LB-ZeroCost-OnOff (LB).

31 at 4:40, where Seq-Aw-T allows more Off→On sequences to be scheduled. This is the reason why the difference between the overall energy consumption of these models is thin. Both of these models lead to major energy savings, respectively 34.00% and 33.99% of energy savings compared to No-OnOff, as shown in Table 5. In comparison with the No-OnOff trace replay, major power peaks are witnessed because of the application of these models. For instance, on Oct. 31, after a peak of work around $12000W$, a very low peak is witnessed around $1000W$. Such behaviors could lead to abrupt thermal changes and thus to hotspots and cool spots, so to possible deterioration of the computing nodes.

We also compare with LB-ZeroCost-OnOff, the model with immediate On→Off with zero cost, and we see that there is no significant difference in energy consumption observed when we accurately describe the cost of On→Off and Off→On sections. However, the number of On→Off that are effectively triggered is significantly lower, since we would not be able to resume the execution in practice if we were using the LB-ZeroCost-OnOff model.

5.2 Electricity-aware model: Elec-SF

This section presents the results for Elec-SF. ESF_{Max} is set to the maximum value witnessed during the No-OnOff replay (for ESF_{Up} and ESF_{Down}). For other ESF replays, we divided ESF_{Max} by a factor to simulate more constrained electrical context.

Figure 8 presents No-OnOff, Seq-Aw-T and Elec-SF with ESF set to max values witnessed during the No-OnOff replay. We can note that Elec-SF does not give the same results as Seq-Aw-T, 33.99% and 24.19% of energy savings compared to No-OnOff as showed in Table 5, respectively. Thus, from extracted ESF factors from No-OnOff, we cannot get the same results as Seq-Aw-T.

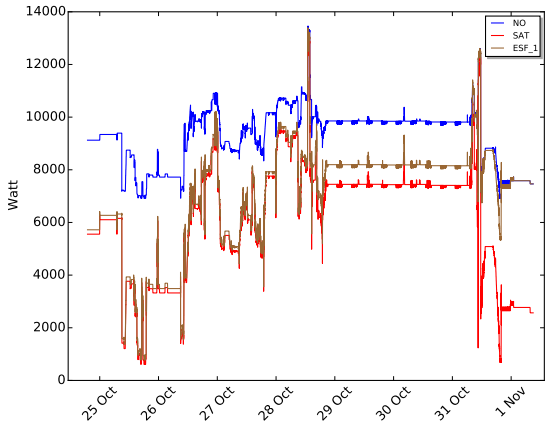

Figure 8: No-OnOff (NO), Elec-SF with max factor (ESF_1), and Seq-Aw-T (SAT).

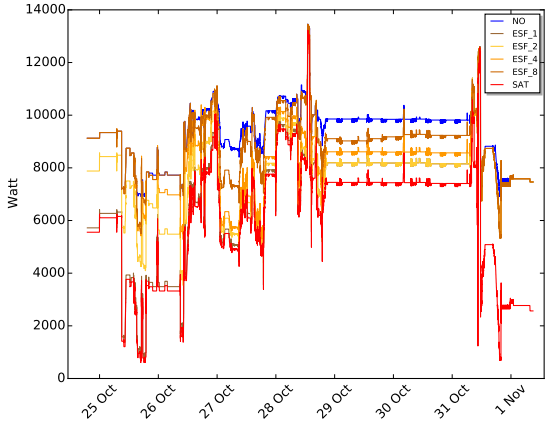

Figure 9: No-OnOff (NO), Elec-SF with all factors (ESF_1, ESF_2, ESF_4, ESF_8), and Seq-Aw-T (SAT).

Figure 9 presents No-OnOff, Seq-Aw-T and Elec-SF with ESF_{Max} divided by $(1, 2, 4, 8)$. One can note that the higher the ESF factors, the closest to Seq-Aw-T we can get. For instance, around Oct. 25 at 11:45, the one with the lowest overall power usage (thus with the highest number of Off→On sequences allowed) is the Seq-Aw-T replay, then we have ESF_{Max}; $ESF_{Max}/2$ comes third and so on until $ESF_{Max}/8$, which is merged with No-OnOff. The influence of Elec-SF could also be clearly witnessed around Oct. 31 at 4:40, where Seq-Aw-T is the only model allowing such an important peak, while none of ESF models can allow such a behavior, because of fixed constraints.

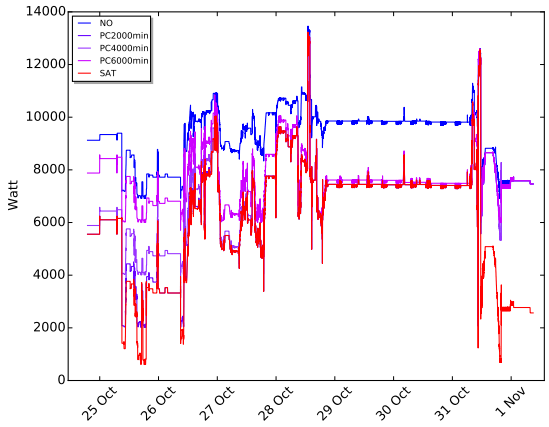

Figure 10: No-OnOff (NO), Power-Cap (with $POWERCAP\_Min = 2000, 4000, 6000$) and Seq-Aw-T (SAT).

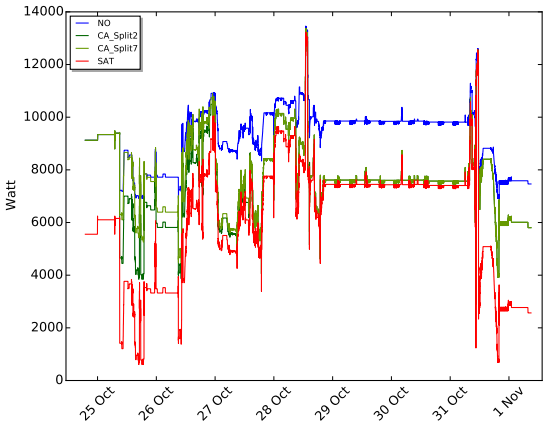

Figure 11: No-OnOff (NO), Cool-Aw (CA_Split2 and CA_Split7), and Seq-Aw-T (SAT).

5.3 Power-Cap

This section presents the results on the Power-Cap model. We set a maximum and a minimum power cap throughout the simulation. We then modulate the minimal power cap to see how it acts with the trace replay. As a reminder, to only evaluate the shutdown leverage, scheduled jobs are fixed. Thus, we did not vary the maximal power cap because it highly depends on jobs and also because the difference between P_{idle} and P_{off} is more important than the difference between the peak witnessed during the Off→On or On→Off sequences and P_{idle}.

Figure 10 shows results of No-OnOff, Seq-Aw-T and Power-Cap with $2000, 4000$ and 6000 as $POWERCAP\_Min$. Even with the highest minimum power cap, here $6000W$, we still make important energy savings (around 16.82 % compared to No-OnOff). The stratified power usage for every respected power cap was expected. In fact, a lower power cap permits more Off→On sequences to be scheduled and thus, more energy savings. The lowest cap constraint ($2000W$) shows that we can respect a minimum power capping and still have a close to minimum consumption.

5.4 Cool-Aw

Figure 11 represents the replay with No-OnOff, Seq-Aw-T, and Cool-Aw models with two different set-ups. With *Split*7, we simulate a "smooth" scalability with 7 levels. We set the upper class (from 14000 W to 12000 W, noted [14000 : 12000], class 1) CSF_{Max} to ESF_{Max}. Then, every 2000 W size class under it divides ESF_{Max} by i, with i the class number. For instance, the [12000 : 10000] class has a CSF_{Max} factor of $ESF_{Max}/2$, until the [2000 : 0] class with a CSF_{Max} factor of $ESF_{Max}/7$. Second, for *Split*2, two levels are set. CSF_{Max} of [14000 : 7000] is set at ESF_{Max} and [7000 : 0] is set at $ESF_{Max}/7$, which represents a more abrupt set-up. The logic is the same for CSF_{Min}.

*Split*2 allows more On→Off sequences to be scheduled, and thus gets better energy savings. *Split*2 stays longer with ESF_{Max} as the CSF_{Max} factor. For instance, note that from Oct. 25 at 7:00 AM to Oct. 27, *Split*2 is closer to Seq-Aw-T whereas *Split*7 is above both of them in Figure 11. Such a behavior is due to a less constrained setup in *Split*2 in the upper classes.

5.5 Renew-E

Figure 12 presents an example of the usage of Renew-E, Seq-Aw-T and No-OnOff. The provenance of energy is "Green" from start to Oct. 29 at 10:00 AM, "Brown" the rest of the time. As a reminder, this

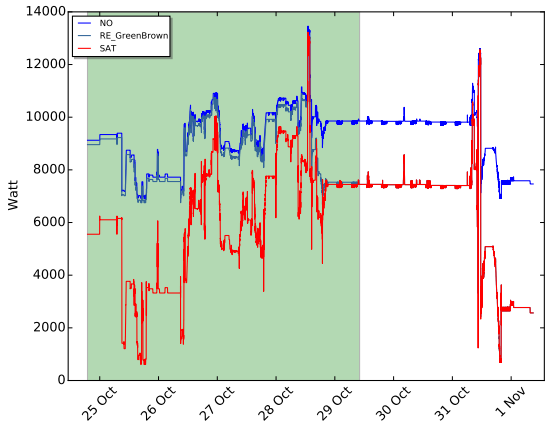

Figure 12: No-OnOff (NO), Renew-E (RE_GreenBrown, green then brown energy) and Seq-Aw-T (SAT).

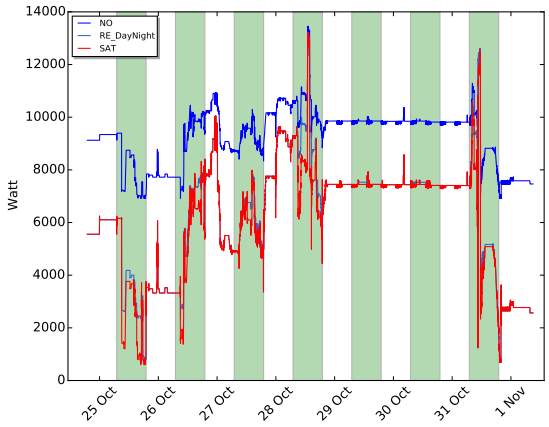

Figure 13: No-OnOff (NO), Renew-E (RE_DayNight, alternating green and brown energy) and Seq-Aw-T (SAT).

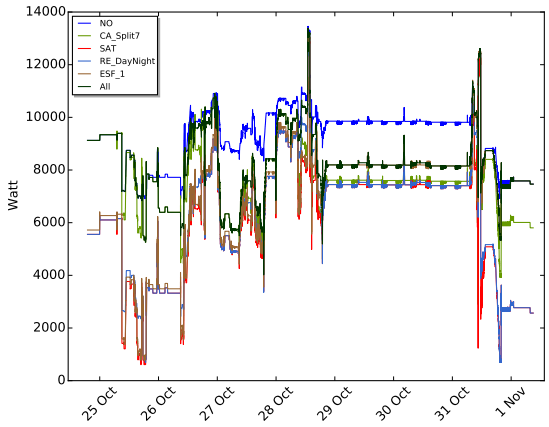

Figure 14: Independent models and all combined models (All).

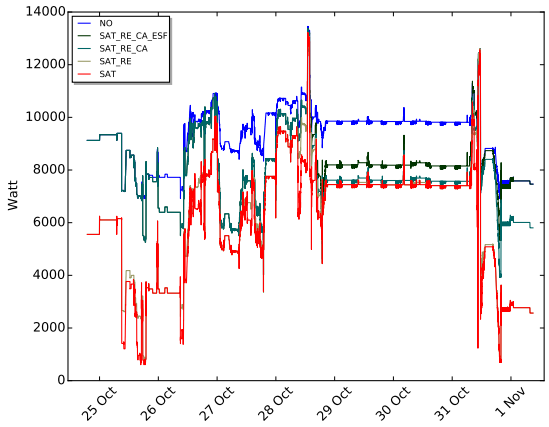

Figure 15: Progressively combined models.

model minimizes the usage of "Brown" energy by scheduling an On→Off sequence on a node if its current idle section contains "Brown" energy. This is why from start to Oct. 29 at 10:00 AM, almost no node is turned off (Renew-E very close to No-OnOff). The shift between Renew-E and No-OnOff at the beginning means that a few nodes are not used in the "Green" section. Around Oct. 29 at 10:00 AM, nodes start to shutdown due to the shift of the provenance (from "Green" to "Brown").

Figure 13 presents a typical usage of renewable energy. We set "Green" provenance during the day (from 7:00 AM to 7:00 PM) and "Brown" provenance at night. We can see that such a model with this input is very close to Seq-Aw-T. "Green" periods can be witnessed for example Oct. 24 at 10:45 PM or Oct. 28 at 8:00 PM (basically where Seq-Aw-T is not fused to Renew-E). One can note that the energy benefits of Renew-E (32.07%) are very close to Seq-Aw-T (33.99%) with 2.2 times less On→Off sequences scheduled (it means that the Lyon site from Grid'5000 is extensively used during the day).

5.6 Combining the models

Figure 14 presents all the models previously exposed (Elec-SF with ESF_{Max}, Cool-Aw with *Split*7, Renew-E with DayNight, Seq-Aw-T and No-OnOff) and "All" is the combination of all of them. The combination of all the models matches a behavior of one of the models that is part of the combination. For instance, around Oct. 28 at 8:00 PM or Oct. 25 at 10:45 PM, we recognize the behavior of Renew-E where nodes stay up during green provenance. At the beginning, it matches the behavior of Cool-Aw with *Split*7 and it is very constrained at the beginning. Between Oct. 29 at 10:00 AM and Oct. 31 at 2:45 AM, we recognize the constraints set by Elec-SF with ESF_{Max} not being able to have the same gain as Seq-Aw-T. And finally, the behavior around the peak on Oct. 31 at 4:40 where "All" cannot go as low as Renew-E or Seq-Aw-T is similar to the behavior seen with Cool-Aw and Elec-SF.

Combined models	Total energy consumed	# On→Off
Seq-Aw-T	4,015,736,064	964
Seq-Aw-T, Renew-E	4,132,487,936	440
Seq-Aw-T, Renew-E, Cool-Aw	5,162,120,192	342
Seq-Aw-T, Renew-E, Cool-Aw, Elec-SF (All)	5,386,375,168	315

Table 6: Progressively combined models.

While Figure 14 presents all models independently in a defined configuration and their combination (All), Figure 15 progressively combines the models together. For instance, "SAT_RE" represents the combination of Seq-Aw-T and Renew-E models, and "SAT_RE_CA_ESF" corresponds to "All" in Figure 14. Table 6 represents the energy consumption and the number of On→Off sequences scheduled during the combined models of Figure 15.

One can note that each added model brings more constraints and thus allows less On→Off sequences to be scheduled, compared to the previous combination, as shown in Table 6. Thus, the chosen combination of models does have an effect on energy consumption and the number of scheduled sequences.

6 Conclusion

In this paper, we have explored the shutdown leverage as a technique to save energy on large scale computing systems. While it is often assumed that nodes can be turned off at no cost, we explore realistic scenarios where several constraints (power capping, electricity, thermal) may prevent us from turning off a node at a given time step. We formally define models targeting various scenarios. Furthermore, we explain how these models can be combined together. A possible usage of these models is illustrated through a set of simulations on a real workload trace, showing the gain in energy that can be achieved given the constraints on the platform, and providing clear guidelines about when a node can be turned off. Overall, the gain of the non-realistic model where nodes are instantaneously turned off during an idle period is very small over the sequence-aware model that turns off a node only if there is time to turn it on again before the next computation, and accounts for the power consumption during the Off→On and On→Off sequences. Other models (*electricity-aware, power-capping-aware, cooling system-aware, renewable energy-aware*) further constrain the number of Off→On, hence leading to more energy consumed, but better matching real-life scenarios.

This is a first building block of a more general approach that could include other leverages, such as dynamic voltage and frequency scaling. Future work also includes the investigation of other usages for the models. While we have focused on an always running actor making local decisions, we could also consider that the actor makes its own decisions, based on the whole list of intervals of idle time on each node to take into account past decisions, while following the rules of the model at any point in time. Eventually, the models should be combined with a clever scheduler, that will decide when to execute jobs in order to minimize the energy consumption, while ensuring that all constraints are satisfied.

Acknowledgments

We would like to thank the reviewers for their valuable comments and suggestions, which greatly helped improve the quality of this paper. This work is integrated and supported by the ELCI project, a French FSN ("Fond pour la Société Numérique") project that associates academic and industrial partners to design and provide software environment for very high performance computing.

References

[1] A.-C. Orgerie, L. Lefèvre, and J.-P. Gelas, "Save Watts in Your Grid: Green Strategies for Energy-Aware Framework in Large Scale Distributed Systems," in *IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, pp. 171–178, Dec 2008.

[2] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat, and R. P. Doyle, "Managing Energy and Server Resources in Hosting Centers," in *ACM Symposium on Operating Systems Principles (SOSP)*, pp. 103–116, 2001.

[3] E. Pinheiro, R. Bianchini, E. V. Carrera, and T. Heath, "Load Balancing and Unbalancing for Power and Performance in Cluster-Based Systems," in *Workshop on Compilers and Operating Systems for Low Power*, pp. 182–195, 2001.

[4] E. D. Demaine, M. Ghodsi, M. T. Hajiaghayi, A. S. Sayedi-Roshkhar, and M. Zadimoghaddam, "Scheduling to Minimize Gaps and Power Consumption," in *Annual ACM Symposium on Parallel Algorithms and Architectures (SPAA)*, pp. 46–54, 2007.

[5] A. Gandhi, V. Gupta, M. Harchol-Balter, and M. A. Kozuch, "Optimality analysis of energy-performance trade-off for server farm management," *Performance Evaluation*, vol. 67, no. 11, pp. 1155–1171, 2010.

[6] M. Lin, A. Wierman, L. L. H. Andrew, and E. Thereska, "Dynamic Right-sizing for Power-proportional Data Centers," *IEEE/ACM Transactions on Networking (TON)*, vol. 21, pp. 1378–1391, Oct. 2013.

[7] A. B. Yoo, M. A. Jette, and M. Grondona, *International Workshop Job Scheduling Strategies for Parallel Processing (JSSPP)*, ch. SLURM: Simple Linux Utility for Resource Management, pp. 44–60. Springer, 2003.

[8] A.-C. Orgerie and L. Lefèvre, "ERIDIS: Energy-Efficient Reservation Infrastructure for Large-scale Distributed Systems," *Parallel Processing Letters*, vol. 21, no. 02, pp. 133–154, 2011.

[9] R. Urgaonkar, U. C. Kozat, K. Igarashi, and M. J. Neely, "Dynamic resource allocation and power management in virtualized data centers," in *IEEE Network Operations and Management Symposium (NOMS)*, pp. 479–486, April 2010.

[10] D. Gmach, J. Rolia, L. Cherkasova, and A. Kemper, "Resource Pool Management: Reactive Versus Proactive or Let's Be Friends," *Computer Networks*, vol. 53, pp. 2905–2922, Dec. 2009.

[11] B. Pernici, C. Cappiello, M. Fugini, P. Plebani, M. Vitali, I. Salomie, T. Cioara, I. Anghel, E. Henis, R. Kat, D. Chen, G. Goldberg, M. vor dem Berge, W. Christmann, A. Kipp, T. Jiang, J. Liu, M. Bertoncini, D. Arnone, and A. Rossi, "Setting Energy Efficiency Goals in Data Centers: The GAMES Approach," in *Energy Efficient Data Centers* (J. Huusko, H. de Meer, S. Klingert, and A. Somov, eds.), vol. 7396 of *Lecture Notes in Computer Science*, pp. 1–12, Springer, 2012.

[12] S. Srikantaiah, A. Kansal, and F. Zhao, "Energy aware consolidation for cloud computing," in *USENIX Conference on Power Aware Computing and Systems (HotPower)*, pp. 1–5, 2008.

[13] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, and X. Zhu, "No "power" struggles: Coordinated multi-level power management for the data center," in *ACM International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS)*, pp. 48–59, 2008.

[14] W. Zhang, Y. Wen, Y. W. Wong, K. C. Toh, and C. H. Chen, "Towards Joint Optimization Over ICT and Cooling Systems in Data Centre: A Survey," *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 1596–1616, 2016.

[15] H. Esmaeilzadeh, E. Blem, R. St. Amant, K. Sankaralingam, and D. Burger, "Power Limitations and Dark Silicon Challenge the Future of Multicore," *ACM Transactions on Computer Systems (TOCS)*, vol. 30, pp. 11:1–11:27, Aug. 2012.

[16] B. Rountree, D. H. Ahn, B. R. de Supinski, D. K. Lowenthal, and M. Schulz, "Beyond DVFS: A First Look at Performance under a Hardware-Enforced Power Bound," in *IEEE International Parallel and Distributed Processing Symposium Workshops PhD Forum (IPDPSW)*, pp. 947–953, May 2012.

[17] X. Zhan and S. Reda, "Techniques for energy-efficient power budgeting in data centers," in *ACM/EDAC/IEEE Design Automation Conference (DAC)*, pp. 1–7, May 2013.

[18] R. Bolze, F. Cappello, E. Caron, M. Dayde, F. Desprez, E. Jeannot, Y. Jégou, S. Lanteri, J. Leduc, N. Melab, G. Mornet, R. Namyst, P. Primet, B. Quétier, O. Richard, T. El-Ghazali, and I. Touche, "Grid'5000: A Large Scale And Highly Reconfigurable Experimental Grid Testbed," *International Journal of High Performance Computing Applications*, vol. 20, no. 4, pp. 481–494, 2006.

Biographies

Anne Benoit received the PhD degree from Institut National Polytechnique de Grenoble in 2003, and the Habilitation à Diriger des Recherches (HDR) from École Normale Supérieure de Lyon (ENS Lyon) in 2009. She is currently an associate professor in the Computer Science Laboratory LIP at ENS Lyon, France. She is the author of 38 papers published in international journals, and 78 papers published in international conferences. She is the advisor of 8 PhD theses. Her research interests include algorithm design and scheduling techniques for parallel and distributed platforms, and also the performance evaluation of parallel systems and applications, with a focus on energy awareness and resilience. She is Associate Editor of IEEE TPDS, JPDC, and SUSCOM. She is the program chair of several workshops and conferences, in particular she is program chair for HiPC'2016, program co-chair for ICPP'2017, and technical papers chair for SC'2017. She is a senior member of the IEEE, and she has been elected a Junior Member of Institut Universitaire de France in 2009.

Laurent Lefèvre is a permanent researcher in computer science at Inria (the French Institute for Research in Computer Science and Control). He is a member of the *Algorithms and Software Architectures for Distributed and HPC Platforms* (Avalon) team from the LIP laboratory in Ecole Normale Superieure of Lyon, France. He has organized several conferences in high performance networking and computing and he has been member of several program committees. He has co-authored more than 100 papers published in refereed journals and conference proceedings. His interests include energy efficiency in large-scale distributed systems, high performance computing, distributed computing and networking, high performance networks protocols and services.

Anne-Cécile Orgerie received her PhD. degree in Computer Science from École Normale Supérieure de Lyon (France) in September 2011. From October 2011 to September 2012, she was working as a postdoc at the Department of Electrical and Electronic Engineering at the University of Melbourne (Australia) on the PetaFlow project (French-Japanese project). She has been a full time researcher at CNRS in the IRISA laboratory (Rennes, France) since October 2012. Her research interests focus on energy efficiency, cloud computing, and distributed systems.

Issam Raïs is a Phd Student in the *Algorithms and Software Architectures for Distributed and HPC Platforms* (Avalon) team from the LIP laboratory in Ecole Normale Superieure of Lyon, France. His interests include energy efficiency for high performance computing systems.