Your Cluster is not Power Homogeneous: Take Care when Designing Green Schedulers!

Mohammed El Mehdi Diouri, Olivier Glück, Laurent Lefèvre and Jean-Christophe Mignot
INRIA Avalon Team, Ecole Normale Supérieure de Lyon
Laboratoire de l'Informatique du Parallélisme (UMR CNRS 5668, ENS, INRIA, Université de Lyon)
Ecole Normale Supérieure de Lyon, 46 Allée d'Italie, 69364 Lyon Cedex 07, France
{mehdi.diouri, olivier.gluck, laurent.lefevre, jean-christophe.mignot}@ens-lyon.fr

***Abstract*—Future supercomputers will consume enormous amounts of energy. These very large scale systems will gather many homogeneous clusters. In this paper, we analyze the power consumption of the nodes from different homogeneous clusters during different workloads. We classically observe that these nodes exhibit the same level of performance. But we also show that different nodes from a homogeneous cluster may exhibit heterogeneous idle power energy consumption even if they are made of identical hardware. Hence, we propose an experimental methodology to understand such differences. We show that CPUs are responsible for such heterogeneity which can reach 20% in terms of energy consumption. So energy aware (Green) schedulers must take care of such hidden heterogeneity in order to propose efficient mapping of tasks. To consume less energy, we propose an energy-aware scheduling approach taking into account the heterogeneous idle power consumption of homogeneous nodes. It shows that we are able to save energy up to 17 % while exploiting the high power heterogeneity that may exist in some homogeneous clusters.**

I. Introduction

Supercomputers are systems built from a collection of computers performing tasks in parallel, in order to achieve very high performance. Driven by the new scientific challenges, the high performance computing (HPC) community requires more and more highly performant systems. Focusing on performance has led to low energy efficient systems with a very high total cost of ownership (TCO) [1]. According to the TOP500 list[1] published in November 2012, the most performant supercomputer is the Titan platform, a machine with more than 500,000 cores which consumes more than 8MW for a maximum performance of 17 PFlop/s.

In recent years, the HPC community has acknowledged that the energy efficiency of HPC systems is a major concern in designing future exascale systems [2], [3]. The Defense Advanced Research Projects Agency (DARPA) has set to 20 MW, the maximum energy consumption of an exascale supercomputer [4]. Furthermore, the Green500 list[2] raises the awareness of power and energy consumption in supercomputing by reporting the power consumption and energy efficiency of large-scale HPC facilities.

In this paper, we identify hidden sources of heterogeneity in terms of energy consumption for apparently homogeneous clusters. We show through a detailed analysis that the main factor of heterogeneity in the idle power consumption is due to CPUs consumptions. This original result has an impact for the design of efficient energy-aware components like resource managers or schedulers. Then, we propose to take into account the heterogeneity in terms of power consumption in order to build energy-aware schedulers and we evaluate this impact using different scenarios.

This paper is organized as follows. Section 2 describes previous related works. Section 3 presents the experimental infrastructure. In Section 4, we observe the power heterogeneity in nodes from homogeneous clusters. In Section 5, we try to explain the origins of such heterogeneities. Section 6 presents *FLIP* (First Less Idle Power), an energy-aware scheduling policy and provides a validation of this approach. Section 7 concludes the paper and presents some future works.

II. Related work

The issue of energy efficiency in distributed platforms has mainly been taken into account in the context of grids, datacenters, or cloud computing. Combined with hardware optimizations offered by manufacturers, there are mainly two approaches to reduce the energy consumption of distributed platforms: the slowdown and the shutdown approaches.

The slowdown approach consists in dynamically adjusting the performance level of a resource according to the performance level that the application and the user really need. For instance, many studies use DVFS techniques (Dynamic Voltage Frequency Scaling) for adapting the speed of processors [5] depending on the performance required by the application. These techniques have inspired the definition of different energy states characterized by the CPU frequency, voltage and power consumption.

The shutdown approach consists in dynamically turning off unused resources and turning them back only when they are needed. Many works like [6]–[8] are based on this approach and suggest using on and off algorithms in order to avoid consuming energy while machines are idle. However, these works do no consider to select the best nodes to switch off (i.e. the more consuming ones) as they consider that the power consumption is the same for the nodes belonging to a same cluster.

Evaluating and measuring power consumption of one node or one process is not new, but it can be a challenging task [9]. For instance, in [10], the authors measure the energy consumed by the nodes of the Grid'5000 Lyon site. For their energy measurements, they use a dedicated energy-sensing

[1]Top500 list: http://www.top500.org/
[2]Green500 list: www.green500.org

infrastructure available on Grid'5000 [11] Lyon site. The authors analyze information on the energy consumed by the nodes and analyze the correlation between the energy logs collected and the user resource reservation requests. In [12], the authors present another way of evaluating application power consumption. They describe a methodology for predicting the power consumption of a computer, depending on performance counters, and then use these counters to predict the power consumption of each single process.

Many works like [13]–[15] assume that nodes from a homogeneous cluster have the same power consumption. In this paper we observe a different reality. Moreover, it has been shown in [16], [17] that nodes from a same cluster may have a different power consumption due to fluctuations caused by the external environment, such as external temperature and position of the node in the rack. However, we show in this paper that in some cases, we may have a significant heterogeneity in terms of power consumption that could not be explained by these reasons. We also try to give some explanations about this power heterogeneity and propose an energy-aware approach based on this heterogeneity in order to reduce the energy consumption of HPC executions.

III. Experimental Infrastructure

In order to analyze and evaluate the power consumption of identical nodes, we used three different clusters from the large scale experimental platform Grid'5000 [11]. The specifications are detailed in Table I.

We monitor the *Sagittaire* and *Taurus* clusters with an energy-sensing infrastructure of external wattmeters from the SME Omegawatt. This energy-sensing infrastructure, which was also used in [10], enables to get at each second the mean power consumption in Watts computed over up to 6000 power samples for each monitored node [18]. We consider that this mean power consumption displayed each second is a very accurate instantaneous power measurement. Logs provided by the energy-sensing infrastructure are displayed lively and stored into a database, in order to enable users to get the power and the energy consumption of one or more nodes between a start date and an end date.

We monitor the *Stremi* cluster with an energy-sensing infrastructure of Raritan power distribution units (PDUs). These PDUs provide the instantaneous power consumption in Watts each three seconds for each monitored node. Logs from these PDUs are saved into a database and can be obtained through SNMP requests.

IV. Do nodes from a homogeneous cluster consume the same power?

In this section, we wonder if nodes from a homogeneous cluster consume the same power. In Section IV-A, we run some specific workloads on nodes from a given homogeneous cluster. In Section IV-B, we show that our first conclusions on this cluster can be generalized to the other clusters. Finally, in Section IV-C, we explain the origin of these differences by analyzing the behavior of nodes from the homogeneous clusters when they are idle: the nodes are switched on but they run nothing else than the operating system.

A. For a given cluster executing different workloads? The Sagittaire case

We start by wondering whether nodes from a given homogeneous cluster that run the same benchmarks consume the same power. In order to answer to this question, we choose the *Sagittaire* cluster since it is the homogeneous cluster that has the highest number of nodes. For the *Sagittaire* cluster, we simultaneously run on each node different intensive workloads that stress some specific parts of the nodes and we measure the power consumption of each node during each workload. The CPU, Memory, and HDDs are the main components we aim to stress. We run each benchmark at the same time on all the nodes of a cluster in order to be sure that the environmental conditions were the same during the power measurements. To achieve this purpose, we select the following benchmarks:

- *cpuburn*[3]: This benchmark heats up any CPU to the maximum possible operating temperature that is achievable by using ordinary software.
- *burnMMX*[4]: This program comes with the *cpuburn* package and specifically stresses cache and memory interfaces.
- *hdparm*[5]: This application is a command line utility for the Linux operating system and is used to set and view parameters of various hard disk drives interfaces such as SATA, PATA, SAS, SAT, and IDE. We use the *-t* option to perform timings of device reads and to stress the HDD.

We consider the following scenarios for each node:

1) All cores run *cpuburn* during 60 seconds.
2) All cores run *burnMMX* during 60 seconds.
3) One core of each node run *hdparm* during 60 seconds.

Figure 1 presents the power profile on all the *Sagittaire* nodes running 60 seconds of *cpuburn*, then *burnMMX*, and *hdparm* benchmarks. Figure 1 shows that the power profiles of the nodes remain almost constant during the 60 seconds of *cpuburn* and during the 60 seconds of *burnMMX* benchmarks. During the 60 seconds of *hdparm* benchmark, the power is oscillating between a minimum and a maximum value of power consumption. This is explained by the fact that this benchmark alternatively writes and reads on/from the hard disk drive [19].

Moreover, Figure 1 shows that identical nodes from the *Sagittaire* cluster do not consume the same power while running exactly the same benchmark in the same environmental conditions. For the *cpuburn* benchmark, the less consuming node consumes around 225W whereas the most consuming one consumes around 275W. This difference of 50W is very significant. It represents about 22%. For the *burnMMX* benchmark, the power consumption ranges from 215W for the less consuming node to 265W for the most consuming one. With the *hdparm* benchmark, the power consumption ranges from 165W to 215W when we consider the drops and from 170W to 220W when we look at the spikes. Will we notice the same

[3] *cpuburn*: http://manpages.ubuntu.com/manpages/precise/man1/cpuburn.1.html

[4] *burnMMX*: http://pl.digipedia.org/man/doc/view/burnMMX.1

[5] *hdparm*: http://linux.die.net/man/8/hdparm

Cluster name	*Sagittaire*	*Stremi*	*Taurus*
Location	Lyon	Reims	Lyon
Installation year	Beginning of 2005	Beginning of 2011	End of 2012
Power measurement device	OmegaWatt	PDU Raritan	OmegaWatt
Operating system	Debian 6 (Squeeze)	Debian 6 (Squeeze)	Debian 6 (Squeeze)
Number of identical nodes	60 Sun Fire V20z	42 HP Proliant DL165 G7	16 Dell R720
CPUs per node	2 AMD Opteron 250 2.4 GHz	2 AMD Opteron 6164 1.7GHz	2 Intel Xeon 2.3 GHz
Number of cores per node	2 cores	24 cores	12 cores
Memory	2 GB	48 GB	32 GB
Hard disk drive	73 GB SCSI	250 GB SATA	598 GB SCSI
Network	Gigabit Ethernet	Gigabit Ethernet	10 Gigabit Ethernet

TABLE I: Specifications of the experimental infrastructure

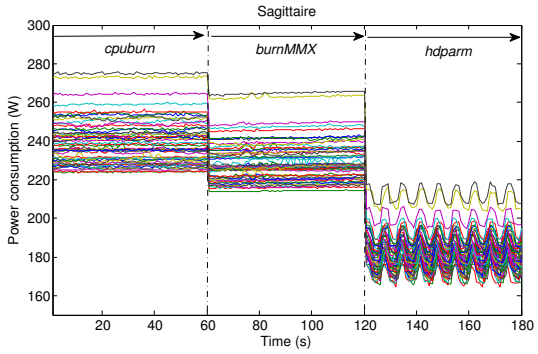

Fig. 1: Power consumption of identical nodes from one homogeneous cluster

differences in terms of power consumption if we consider other homogeneous clusters?

B. What about the other homogeneous clusters? The Stremi *and* Taurus *cases.*

In this section, we wonder if we observe the same differences in terms of power consumption for nodes from other homogeneous clusters: *Stremi* and *Taurus*. For each homogeneous cluster (*Sagittaire*, *Stremi* and *Taurus*), we consider the same scenarios as previously described but for a longer period of time: 10 minutes. Since the power profiles of the *cpuburn* and *burnMMX* benchmarks remain constant for each node and the one of *hdparm* oscillates uniformly around an average value, we compute for each node the mean power consumption during each scenario.

Figure 2 presents the box plots showing the dispersion of the mean power consumption of the nodes for each cluster while running the *cpuburn*, *burnMMX* and *hdparm* benchmarks. Each box plot graphically depicts groups of numerical data through their five-number summaries:

- the smallest observation (sample minimum);
- lower quartile splitting lowest 25% of data;
- median cutting data set in half;
- upper quartile splitting highest 25% of data;
- largest observation (sample maximum).

It also indicates which measurement, if any, should be considered outliers. In order to make it easy to compare, the boxes are plotted on a same scale.

First of all, Figure 2 shows that during each specific benchmark, the power consumption of identical nodes from different homogeneous clusters is not the same. As in Section IV-A, for the *Sagittaire* cluster running *cpuburn*, the less consuming node consumes around 225W whereas the most consuming one consumes around 275W. This difference of 50W is very important (about 22%)! This dispersion is also very significant for the *Stremi* cluster: the mean power consumption per node running *cpuburn* ranges approximately from 250W to 280W. However, for *Taurus* cluster, this dispersion is not as important. Indeed, the mean power consumption per node ranges approximately from 235W to 240W while all cores of each node are fulfilled by *cpuburn*. We notice a similar power dispersion for the *burnMMX* and *hdparm* benchmarks.

Additionally, Figure 2 shows that for a given benchmark, nodes from a same homogeneous cluster are not always uniformly distributed over the whole power consumption interval. Indeed, most of the nodes from a same homogeneous cluster have a power consumption that is close to the median power consumption of all the nodes. For 50% of the nodes from *Sagittaire* cluster running *cpuburn*, the mean power consumption per node ranges approximately from 235W to 245W. For the other 50%, the mean power consumption per node during *cpuburn* is outside the interval [235W, 245W]. As concerns the *Stremi* cluster, 50% of the nodes running *cpuburn* on all cores, have a mean power consumption per node ranging approximately from 262W to 272W while the 50% other nodes are outside this power interval. However, it is not really the same for *Taurus* cluster where nodes seem to be more uniformly distributed. For a fixed cluster, we notice a similar power distribution as well for the *burnMMX* and *hdparm* benchmarks.

The first observations we made concerning the extent and the distribution of the power consumption for nodes from the *Taurus* cluster are not as important as they are for *Sagittaire* and *Stremi* clusters. This could be because for this cluster we only have 16 nodes (which is not very representative) while we have 42 nodes for *Stremi* and 60 nodes for *Sagittaire*. Another explanation could be related to the age of the *Taurus* cluster. Indeed, nodes from this new cluster have been used for only 6 months and for this reason they may still be homogeneous from a power consumption point of view. We will detail this point in Section V.

Furthermore, for a given cluster, we notice that the power dispersion is almost the same while considering different benchmarks. Indeed, for one cluster, this power dispersion seems to be slid over the y axis when we move from one benchmark to another. For instance, with *Sagittaire*, the mean power consumption per node ranges approximately from 225W to 275W for *cpuburn*, from 215W to 265W for *burnMMX* and from 170W to 210W for *hdparm*. Thus, for the *Sagittaire*

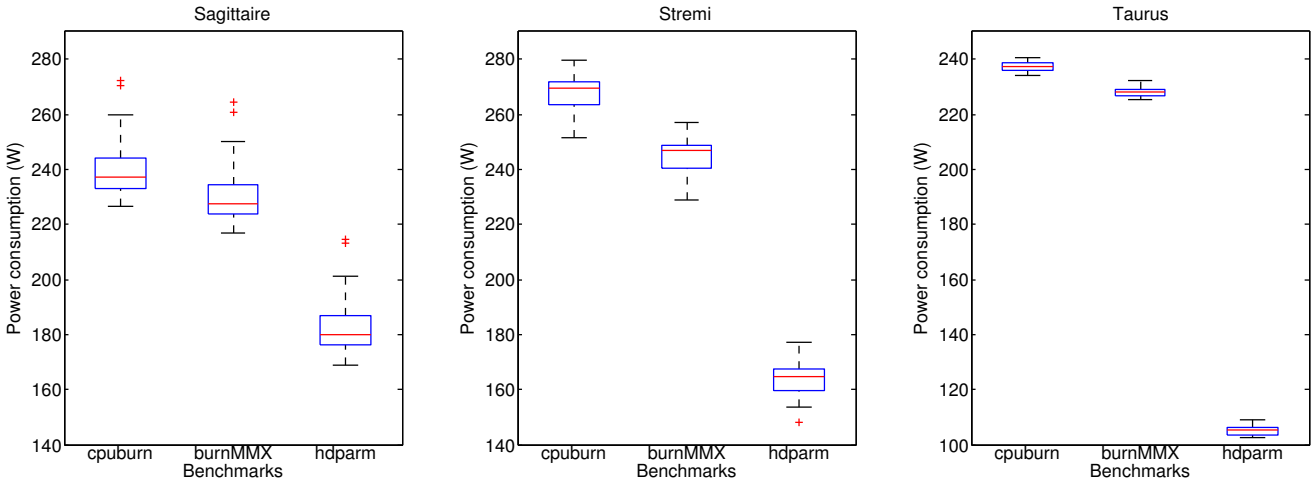

Fig. 2: Power consumption of nodes from three different homogeneous clusters

cluster, we always observe a difference around 50W whatever the considered benchmark. This is also the case when we consider the two other clusters. These last observations lead us to wonder whether this power dispersion remains noticeable even when nodes are idle, that is to say the nodes are switched on but running nothing else than the operating system.

C. When nodes are idle?

In this section, we analyze the idle power consumption of nodes from the three different homogeneous clusters. At this end, we measure during 10 minutes the power consumption of each node while it is idle: it is switched on but running nothing else than the operating system. Thus, we measure only the power consumption of the hardware and the operating system running on this hardware. It lets the hardware promote the cores to a low power consumption states, also known as C-states. We remind that the operating system used is the same for all the nodes of the three homogeneous clusters.

For each homogeneous cluster, we compute the mean power consumption per node during these 10 minutes of idleness. Figure 3 shows that the idle power consumption is not the same for identical nodes from homogeneous clusters. Indeed, the idle power consumption of *Sagittaire* cluster ranges approximately from 165W for the less consuming node to 215W for the most consuming one. The idle power consumption of *Stremi* cluster approximately ranges from 140W to 170W while the one of *Taurus* cluster ranges from 95W to 100W. For each homogeneous cluster, the extent between the two extreme consuming idle nodes is the same as the one we observed for active nodes running a specific benchmark: 50W for *Sagittaire*, 30W for *Stremi* and 5W for *Taurus*.

For each homogeneous cluster, the box plots when the nodes are idle seems to be stackable to the ones that we observed when the nodes are executing a specific workload. This means that the power dispersion of the nodes is the same even when these nodes are idle. Therefore, the power differences observed in identical nodes are not due to the different workloads they are running.

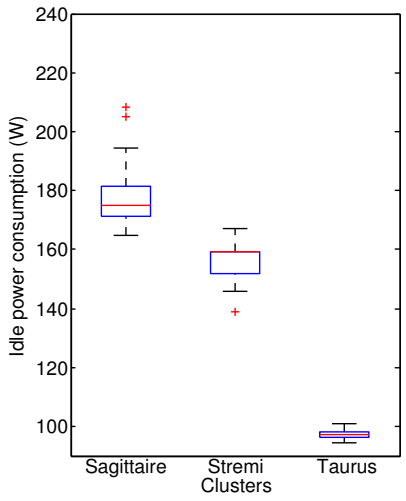

Fig. 3: Idle power consumption of nodes from three different homogeneous clusters

Now that we know that even if identical nodes do not consume the same power when they are running a same benchmark and even when they are idle, we would like to know if the power differences that we observed while nodes are running a given benchmark are due only to the differences in terms of idle power consumption. So, for each benchmark, we subtracted the idle power consumption from the mean power consumption per node during a benchmark. We plot on Figure 4 this extra power consumption per node for the three considered clusters.

Figure 4 shows that the extra power consumption per node during a specific benchmark is the same for all the nodes from a given benchmark. Indeed, for the *Sagittaire* cluster, this extra power consumption per node is around 62W for *cpuburn*, approximately 52W for *burnMMX* and 5W for the *hdparm* benchmark. It shows that the differences of power consumption per node during a workload are almost entirely

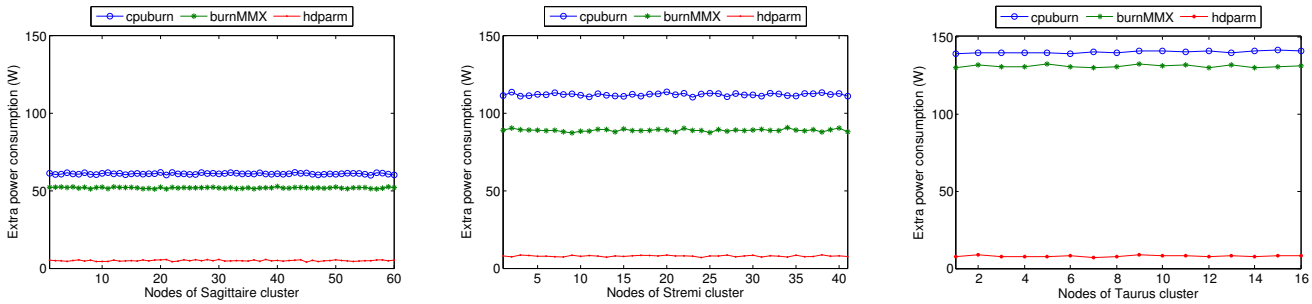

Fig. 4: Extra power consumption of nodes from three different homogeneous clusters during different workloads

due to the differences in terms of idle power consumption. In the next section, we analyze why we observe these differences in terms of idle power consumption whereas the nodes are identical from the hardware point of view.

V. What is the origin of the differences in the idle power consumption?

In this section, we try to find an explanation to the differences in terms of idle power consumption. Section V-A presents the experimental methodology and the following subsections our successive investigations.

A. Experimental Methodology

In order to find out the origin of such differences, we decide to take out from the *Sagittaire* cluster two nodes whose power consumptions are very different: *sagittaire-54* and *sagittaire-57* respectively consuming 167W and 205W when they are idle (a difference of 38W).

First, we suspected that the measurement conditions were not the same for these two nodes. To this end, we checked if these differences are due to the environmental conditions, the positions of nodes in the rack, or to the non-calibration of our wattmeters. Then, we thought that these differences were due to some internal components that may have a heterogeneous power consumption. Thus, we interchanged some internal components (power supply units, HDDs, RAMs) of the two considered nodes in order to detect which were heterogeneous in terms of power consumption. Furthermore, we measured the power consumption of the CPUs and the fans cooling the CPUs using PowerMon2 [20], an internal wattmeter from Renci iLab[6].


Fig. 5: Internal investigation of *sagittaire-54* node

PowerMon 2 is composed of eight individual channels. The three first channels provides the power consumption of the 3.3V, 5V and 12V lines of the motherboard while the five last ones can be used to measure the power consumption of other hardware components such as the 12V lines of CPUs or the 5V lines of the hard disk drives. This device uses an Atmel ATmega168 8-bit microcontroller which communicates with the host computer to provide eight current/voltage lines and timestamps at 1024 Hz per channel and 3072 Hz aggregate. The communication from the PowerMon 2 and the computer is done by an USB port. Voltage and current are detected using an Analog Devices ADM1191 digital power monitor IC on each power channel. Figure 5 presents a photo of the node *sagittaire-54* connected to *PowerMon 2* during our experimentations.

B. Environmental conditions

We first thought that even if the environmental conditions were the same during our power measurements, the temperature may be not uniformly distributed in the experimental room and then, the position of the nodes in the rack might explain the discrepancies of the idle power consumption per node. Thus, we interchanged the positions of these two nodes in the rack. When we measured the power consumption in these new positions, we noticed that the idle power consumption of the two nodes remained unchanged and *sagittaire-54* was still the less consuming node. Therefore, the position in the rack is not the reason why we observe these differences in terms of power consumption.

Our second hypothesis was that the wattmeters used were not calibrated and hence were measuring the power consumption with a constant gap for each outlet. To verify this, we used another wattmeter, *WattsUp Pro*[7]. Figure 6 presents the power consumption measured by two different wattmeters during different workloads. Figure 6 shows that the power consumptions displayed with *WattsUp Pro* is sensibly the same to the ones registered with the *Omegawatt*. Thus, these differences in terms of idle power consumption are not due to the wattmeters that we used.

C. Specific Hardware Components

Then, we supposed that these differences may come from the power supply units of the nodes that may lose their

[6]Renci iLab project website: http://ilab.renci.org/powermon

[7]WattsUp Pro: https://www.wattsupmeters.com/secure/products.php?pn=0

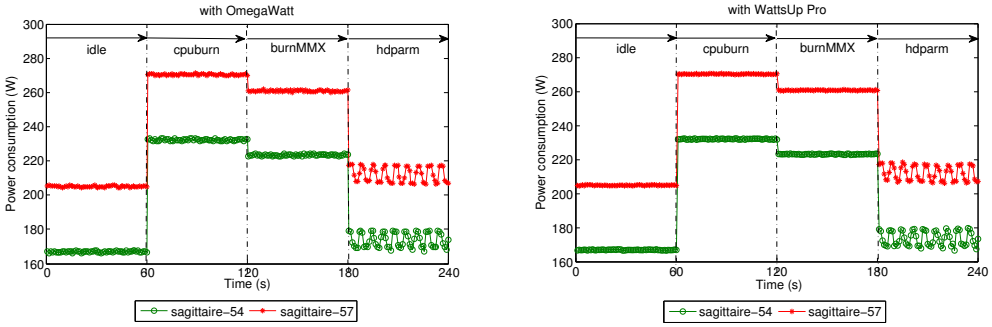

Fig. 6: Comparaison of the power profiles of *sagittaire-54* and *sagittaire-57* using OmegaWatt (left) and WattsUp Pro (right)

efficiency over the time and then generate some power fluctuations. At this end, we interchanged the power supply units of these two nodes and measured the idle power consumption. Once again, we noticed that the idle power consumption remained the same for *sagittaire-54* and for *sagittaire-57*. This means that these discrepancies in the idle power consumption are not due to the power supply units.

After that, we interchanged some internal components of *sagittaire-54* and *sagittaire-57*: the hard disk drives (HDD) and the modules of random access memory (RAM). The idle power measurements for these two nodes remained unchanged when interchanging their HDDs or their RAMs. Hence, the differences of idle power consumption are due neither to the hard disk drives nor to the modules of random access memory.

D. CPU power analysis

We would have liked to swap the CPU from one node to the other but we did not since the CPU were glued to the motherboard and swapping them may have damaged the nodes. Instead of this, we use *PowerMon2* to measure independently the 12V that energize the CPUs and the associated fans of *sagittaire-54* and *sagittaire-57*. Figure 7 shows the power consumption profiles drawn by the *idle*, *cpuburn*, *burnMMX* and *hdparm* benchmarks It is important to note that the internal power consumption shown on Figure 7 only includes the CPUs and the associated fans. This explains the differences with regards to measurements plotted in Figure 6.

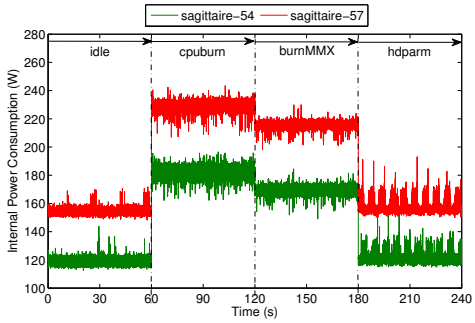

Fig. 7: The CPUs and associated fans consumption of two specific nodes from *Sagittaire* while running different workloads

First, we notice from Figure 7 that when the two nodes are idle, the power consumption of the CPUs and the associated fans is approximately 118W for *sagittaire-54* and 155W for *sagittaire-57*. We almost retrieve the difference of 38W that we observed when measuring externally with *OmegaWatt* wattmeter. Thus, we can conclude that the difference of power consumption is due to the difference of the idle power consumption of the CPUs and the associated fans.

From this internal power consumption, we observe that CPUs and the associated fans represent a great part of the whole power consumption of the node, even if it is idle. Indeed, it is 118W (see Figure 7) over 167W (see Figure 6) for the whole node *sagittaire-54* and 155W over 205W for the whole node *sagittaire-57*. This represents a ratio of 70% for *sagittaire-54* and 75% for *sagittaire-57*.

Let assume that the power consumption of the CPUs and the associated fans is depicted by the following model:

$$P_{CPUs\&Fans} = P_{idle}^{static} + \Delta P_{application}^{dynamic} \quad (1)$$

P_{idle}^{static} is the power consumption of the CPUs and the associated fans when the node is idle. $\Delta P_{application}^{dynamic}$ is the extra power consumption of the CPUs and the associated fans when the related node is running an application.

Even for different benchmarks (*cpuburn*, *burnMMX* and *hdparm*), we still notice the difference of 38W, confirming that this difference is constant whatever the workload running on the two machines. Hence, this difference in power consumption is not due to the workload running depicted by $\Delta P_{application}^{dynamic}$ but on the difference in the idle power consumption P_{idle}^{static} of the CPUs and the associated fans.

Having noticed that the power heterogeneity of identical nodes comes from the CPUs, we analyzed whether these differences also have an impact on the performance of the nodes or not. We measured the FLOPS (FLoat Operations Per Seconds) for each node of the *Sagittaire* cluster but we only found marginal differences between the nodes. Therefore, it seems there is no causal relationship between the power heterogeneity and the performance of each node.

Due to fluctuations in the manufacturing process, different CPUs use different power, even if they have the same frequency and specifications. Specifically for *Sagittaire* and *Stremi* clusters, a complementary explanation is that after some failures, the CPUs or the fans cooling the CPUs may have been worn since this can provoke heterogeneous energy consumption. This may also be due to a heterogeneous accumulation of dust. So, even if the end user can have the feeling of benefiting from a completely homogeneous cluster, heterogeneity at least in terms of energy consumption is present. Indeed, an increasing proportion of power consumption comes

from leakage power that might actually change over the time, due to electromigration process [21]. This could explain why we observe significant differences in terms of idle power consumption for *Sagittaire* cluster which has been running for 8 years in comparison with *Taurus* cluster running only for 6 months. Indeed, the differences we observed in *Taurus* cluster are marginal (5%) and could be due to heterogeneity in the manufacturing processes.

VI. Exploiting the Power Heterogeneity for the Design of Green Schedulers: the *FLIP* Approach

This section presents FLIP (First Less Idle Power), an energy-aware approach that takes advantage of the differences in the idle power consumption between nodes from a homogeneous cluster in order to save energy consumption during the execution of applications over large scale distributed systems.

A. Methodology

Unrealistic scheduling approaches assume that all the nodes of a same cluster have the same power consumption. In [13]–[15], the approaches consider that the energy consumption of the N nodes from a same cluster is equal to the energy consumption of a specific node multiplied by N. In order to study the differences between a scheduling based on the real consumption of nodes and an unrealistic approach where all nodes are supposed to have the same consumption, we distinguished three scheduling policies relying on unrealistic assumptions:

- *Homogeneous_MIN*: all the nodes have the same power as the less consuming node of the cluster. This is the idealistic unreal case.
- *Homogeneous_MEDIAN*: all the nodes have the same power as the median consuming node of the cluster. This is the unrealistic scenario that is the most representative of reality.
- *Homogeneous_MAX*: all the nodes have the same power as the more consuming node of the cluster. This is the worst unrealistic case.

We show in this paper that the reality is different. That's why we now consider the following realistic scheduling approaches that do not suppose equal energy consumption for all the nodes. Hence, we consider the overall energy consumption is equal to the sum of the measured energy for each node. We distinguish the three realistic scheduling policies:

- *OrderedIDs*: the nodes assigned to the user follow the ascending order of the node IDs. This is the default policy on our execution platform.
- *FLIP*: the nodes assigned to the user follow the ascending order of the idle power consumption per node. This is the approach that we propose in this paper: we assign to the user the idle less consuming nodes in priority.
- *FMIP*: the nodes assigned to the user follow the descending order of the idle power consumption per node. This is the worst realistic case.

In order to apply these scheduling policies, we first measure the mean idle power consumption per node from each considered cluster. At this end, we compute the mean idle power consumption with a high number of power measurements so as to get an accurate average.

As concern the unrealistic scheduling policies, we consider by simulation that the measured power consumption of a specific node (less, median or more consuming node) is the same for the other nodes that are running the same application.

Regarding the realistic scheduling policies, we sort the nodes into the ordered lists depending on the scheduling policy as described previously. Figure 8 shows the power consumption of the ordered lists of nodes in the three clusters (cf. Table I) according to the considered scheduling policies. Each time a user wants to reserve a given number of nodes or to execute an application using a certain number of machines, a given scheduling policy assigns him in priority the first available nodes within its ordered list.

B. Evaluation

In order to evaluate the *FLIP* scheduling approach, we consider *Sagittaire*, *Stremi* and *Taurus* clusters. We apply the methodology presented previously in Section VI-A in order to obtain the ordered lists of nodes with their corresponding idle power consumption per node.

1) Applying the shutdown approach with FLIP: First, for each scheduling policy, we consider that the first needed nodes of the corresponding ordered list are switched on and that we apply the shutdown approach (see Section II) on the remaining unused nodes by switching them off. This is done in order to evaluate the power saved from the infrastructure owner point of view (the end user may not be impacted depending on the pricing policy of the site as explained in section VI-B). We evaluate the gain of the shutdown approach by computing the power savings that we obtain by switching off the unused nodes with FLIP as compared to the savings obtained with the other scheduling policies.

Figure 9 shows the power saved on each cluster if we switch off the unused nodes from FLIP instead of switching off the unused ones from the other policies. The x axis represents the number of unused nodes that are switched off. The y axis represents the difference of power saved if we turn off the unused nodes with FLIP instead of turning off the unused nodes with another scheduling policy. Each subfigure corresponds to each cluster. The three subfigures are on the same y scale.

First, we observe that depending on the cluster, the power saved thanks to FLIP is more or less important. In *Taurus*, the power saved seems to be insignificant (less than 20W) in comparison with *Sagittaire* and *Stremi* (more than 200W). The reason is that *Taurus* is much less power heterogeneous in comparison with *Sagittaire* and *Stremi*.

The curves for the realistic scheduling policies are always positive. This means that the *FLIP* scheduling policy is more power saving than the other policies. Indeed, this is due to the fact that remaining unused nodes are among the more consuming ones. Hence, *FLIP* lets the owner save more power when switching off the remaining nodes.

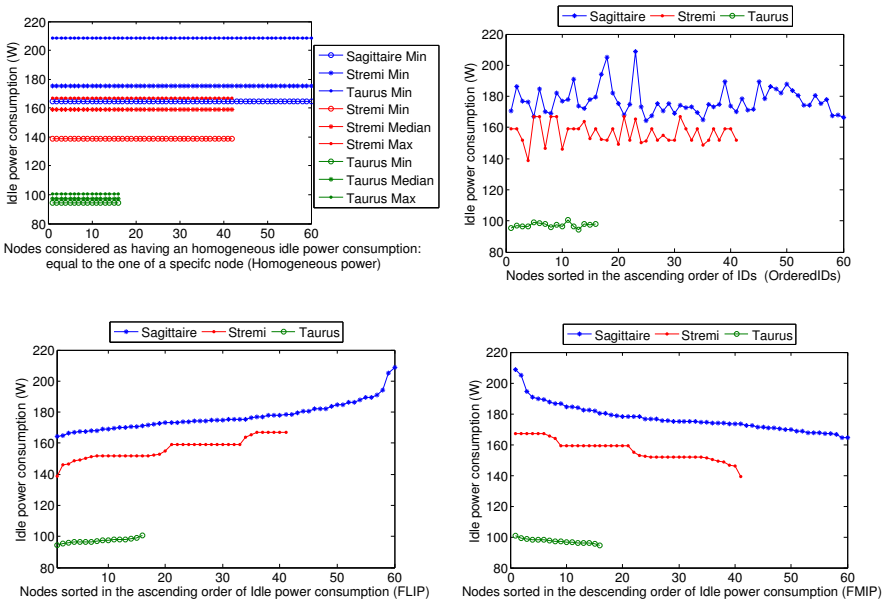

Fig. 8: Ordered lists of nodes obtained with the different scheduling policies (Homogeneous, OrderedIDs, FLIP, FMIP)

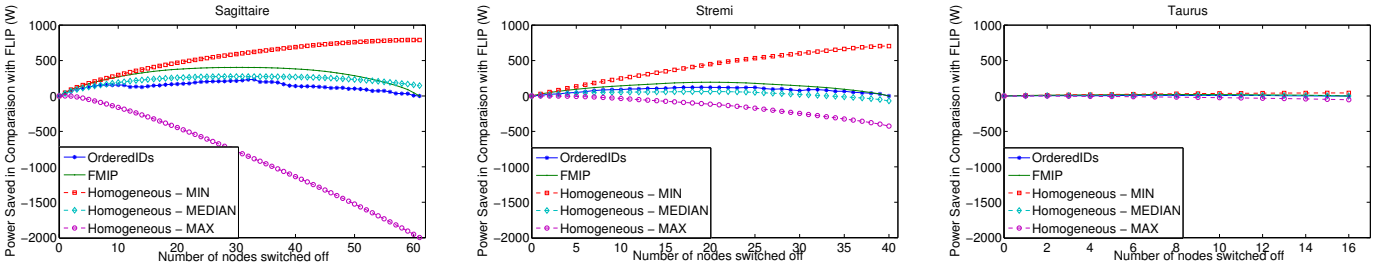

Fig. 9: Power Saved (in Watts) with *FLIP* in comparison with the other scheduling policies on the three clusters

Besides, we observe that for the realistic scheduling policies, we do not save power when no node is switched off ($x = 0$) or when all the nodes are switched off. Compared to *OrderedIDs* and *FMIP*, with *FLIP*, the power saved increases with an increasing number of nodes that remains under 50% of the cluster size. Indeed, until half of the cluster size, there is no overlapping (i.e. not the same nodes) between the ordered list obtained with *FLIP* and the one obtained with *FMIP*. Then, from 50% of the cluster size, the power saved decreases with an increasing number of nodes. In fact, from 50%, we start having more and more overlapping between the ordered lists of nodes obtained with the realistic scheduling policies.

Concerning *OrdererdIDs*, we observe another reality. With an increasing number of nodes, sometimes the power saved increases and sometimes it decreases. This is due to the fact that the ordered list obtained with *OrderedIDs* may overlap with the one obtained with *FLIP* in a random way.

We also notice that the only scenario where we observe a negative power saving is with the *Homogeneous_MAX*. This means that we wrongly save more power by applying *Homogeneous_MAX* instead of *FLIP*. *Homogeneous_MAX* is evaluated as saving more energy than *FLIP* because the nodes it switches off are all considered as consuming as much energy as the the most consuming node of the cluster We also observe that with an increasing number of switched off nodes, the power saved is more and more important when we compare the *FLIP* policy to *Homogeneous_MIN*. Indeed, *Homogeneous_MIN* would save more and more power if, each time there is a new node to switch off, we could switch off the one consuming as much as the less consuming node.

Thus, these unrealistic scenarios give the impression that they are sometimes better or sometimes worse than *FLIP* since with them we save more and more power (*Homogeneous_MAX*) or less and less power (*Homogeneous_MIN*). As we have seen in Section IV, this is not realistic since nodes have not the same power consumption even if they belong to the same cluster.

2) Running HPC applications with FLIP: We consider four HPC applications: four NAS[8] (LU, CG, EP, MG) in Class C running over 8, 16, 32 and 64 processes for *Sagittaire*; over 32, 64, 128 and 256 processes for *Stremi*; and over 8, 16, 32 and 64 processes for *Taurus*. We launch two processes per node for *Sagittaire*; sixteen processes per node for *Stremi*; and eight processes per node for *Taurus*. These execution settings have been chosen in order to fill up more than 33% but less than 100% of the nodes of each cluster. We feel up only a part of

[8]NAS: http://www.nas.nasa.gov/publications/npb.html

each cluster because if we fill up all the nodes of a cluster, applying a scheduling policy will not have sense since all the nodes will be assigned and the global power consumption will be the same for the three realistic scheduling policies.

We compare the energy consumption during the four HPC applications scheduled on the *FLIP* assigned nodes (First Less Idle Power) to the other scheduling policies. Each energy measurement is done 25 times and we compute the average value. Figure 10 shows for the three considered clusters the proportion of energy saved with *FLIP* in comparison with the other scheduling policies. The three subfigures corresponding to the three clusters are on the same y scale. Each dot represents the average relative energy saving over the four HPC applications. The error bars represent the range of deviation from the average value when considering each of the four HPC applications.

In Figure 10, the curves are not overlaid for the three clusters. It confirms that power heterogeneity exists in these three clusters. This power heterogeneity is more or less important depending on the considered cluster. The more the power heterogeneity is important, the more we save energy thanks to the FLIP scheduling policy. Indeed, as we can see in Figure 10, compared to *FMIP*, *FLIP* can save up to 17% on *Sagittaire*, up to 11% on *Stremi* and less than 5% on *Taurus*. From the user point of view, his application will consume less energy if he runs an application using the nodes obtained with *FLIP*. If he is charged based on the amount of electricity, *FLIP* may help him save energy and therefore money. But if he is charged based on the number of nodes, without taking care of their energy consumption, *FLIP* will not impact the user's bill. But the owner will save money since he will charge the user for the nodes that cost him less. Thus, depending on the pricing policy, the *FLIP* approach is gainful either for the user or for the owner.

Compared to *OrderedIDs*, with *FLIP*, we save up to 5% on *Sagittaire*, up to 8% on *Stremi* and less than 2% on *Taurus*. This is noticeable when the proportion of nodes is 4/60 for *Sagittaire*, 2/42 for *Stremi* and 1/16 for *Taurus*. Even if *Sagittaire* cluster is more power heterogeneous than *Stremi*, the proportion of energy saved seems to be more important with *Stremi* in comparison with the *OrderedIDs* policy. This is mainly due to the fact that the two first nodes of the ordered list obtained for *Stremi* are among the most consuming nodes while the four first nodes of the one obtained for *Sagittaire* are among the median consuming nodes.

Furthermore, Figure 10 shows that the curves are decreasing while the proportion of used nodes is increasing. In fact, the more we use nodes, the less is the proportion of energy saved. It is due to the fact that the more we used nodes, the more the power heterogeneity decreases in average. To put it into perspective, if we use all the nodes of a cluster, no matter what the scheduling policy is, we should have the same energy consumption.

Besides, we observe that the deviations from the average values over the four HPC applications are not significant especially for *Sagittaire* and *Taurus*. Even if the execution time is different from one HPC application to another, this observation confirms that the proportion of energy saved does not depend on the workload running on the used nodes. These deviations seem to be more important for *Stremi* certainly due to the lack of accuracy in the measurements taken by the PDUs monitoring the *Stremi* cluster.

In addition, we observe that the curves representing the *Homogeneous_MEDIAN* and *OrderedIDs* are very close and are quite overlaid. This shows that these two scheduling policies are comparable in terms of energy consumption. Indeed, even if the *Homogeneous_MEDIAN* policy is not realitic, this policy provides a better approximation of the energy measurements in comparison with *Homogeneous_MIN* and *Homogeneous_MAX*. Nevertheless, as we may measure a "wrong" node (one among the less or the more consuming nodes), the *Homogeneous* policies are not suitable especially when the power heterogeneity is as significant as observed on *Sagittaire* and *Stremi* clusters. Therefore, in order to measure the energy consumption of an application running on a set of nodes, it is really necessary to measure the energy consumption of each node.

Moreover, we notice that for the three clusters, the only scenario where we observe negative energy savings is with the *Homogeneous_MIN*. Indeed, this is the idealistic scenario where all the nodes are consuming as much as the less consuming node. We also observe that the proportion of energy saved is always the most important when we compare the *FLIP* policy to *Homogeneous_MAX*. In fact, the *Homogeneous_MAX* corresponds to the worst unrealistic scenario since we arbitrary measure the most consuming node and consider that all the others have the same energy consumption.

VII. Conclusion

This paper first presents a detailed power analysis of the nodes from different homogeneous clusters during different workloads. It shows that even if distinct nodes are made of identical hardware, they have a more or less heterogeneous idle power consumption. We presented an experimental methodology that has enabled us to detect that this power heterogeneity mainly comes from the CPUs and/or the fans cooling the CPUs. Since we observed the power heterogeneity especially in *Sagittaire* and *Stremi* which are older clusters, we assume that this power heterogeneity could be due to the more or less intensive usage of these specific hardware components (CPUs and fans). Indeed, this is related to leakage power that might vary over time, due to electromigration process [21]. This is also related to fluctuations in the manufacturing process. For *Taurus*, we observed less discrepancies, since it is a more recent cluster with much less nodes.

We showed that it is wrong to consider that nodes from homogeneous clusters consume the same power when they execute the same workloads. In order to consume less energy, we suggest to take into account this heterogeneity in terms of power consumption in order to build energy-aware schedulers At this end, we propose an energy-aware scheduling approach called *FLIP* taking into account the heterogeneous idle power consumption of homogeneous nodes. We show that the *FLIP* scheduling policy lets the owner save more power when switching off the remaining unused nodes. When executing applications, it shows that we are able to save energy up to 17 % while exploiting the high power heterogeneity that may exist in some homogeneous clusters (like *Sagittaire*). Depending on

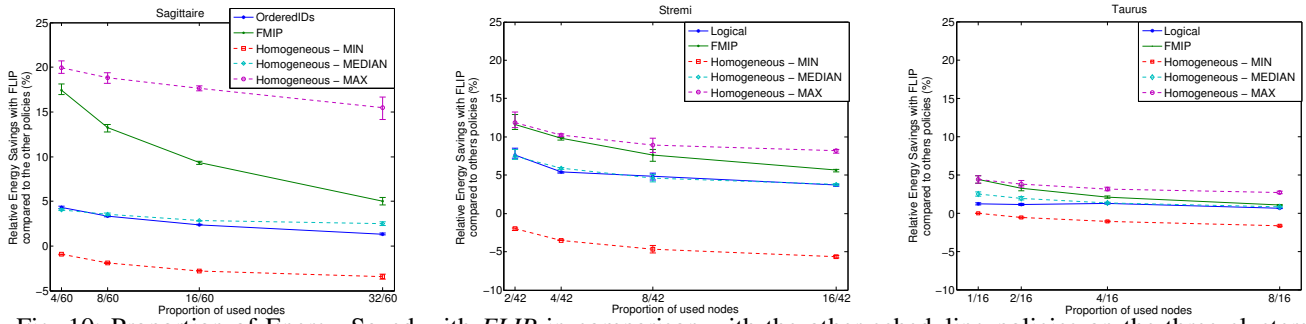

Fig. 10: Proportion of Energy Saved with *FLIP* in comparison with the other scheduling policies on the three clusters

the pricing policy, the *FLIP* approach is financially gainful either for the user or for the owner.

FLIP is focused on the list of nodes that we assign to the user. Our future work consists into taking into account other parameters in a multi-criteria optimization approach. The goal will still be to consume less energy but also to consume it at the lowest financial cost with the lowest pollution impact.

ACKNOWLEDGMENT

Experiments presented in this paper were carried out using the Grid'5000 experimental testbed, being developed under the INRIA ALADDIN development action with support from CNRS, RENATER and several Universities as well as other funding bodies (see https://www.grid5000.fr).

REFERENCES

[1] C.-h. Hsu, W.-c. Feng, and J. S. Archuleta, "Towards efficient supercomputing: A quest for the right metric," in *Proceedings of the High Performance Power-Aware Computing Workshop*, 2005.

[2] J. Dongarra et al, "The international ExaScale software project roadmap," *Int. J. of High Performance Computing & Applications*, vol. 25, no. 1, 2011.

[3] W. Feng, X. Feng, and R. Ge, "Green supercomputing comes of age," *IT Professional*, vol. 10, no. 1, pp. 17 –23, 2008.

[4] P. M. Kogge and et al, "ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems," in *DARPA Information Processing Techniques Office*, Washington, DC, September 28 2008, p. pp. 278.

[5] Y. Hotta, M. Sato, H. Kimura, S. Matsuoka, T. Boku, and D. Takahashi, "Profile-based optimization of power performance by using dynamic voltage scaling on a pc cluster," in *Proceedings of the 20th International in Parallel and Distributed Processing Symposium, IPDPS 2006*, 2006.

[6] 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 *Proceedings of the eighteenth ACM symposium on Operating systems principles*, ser. SOSP'01. Banff, Alberta, Canada: ACM, October 2001, pp. 103–116.

[7] F. Hermenier, N. Loriant, and J.-M. Menaud, "Power Management in Grid Computing with Xen," in *Frontiers of High Performance Computing and Networking - ISPA 2006 International Workshops*, vol. 4331, Sorrento, Italy, December 4-7 2006, pp. 407–416.

[8] A.-C. Orgerie and L. Lefèvre, "When clouds become green: the green open cloud architecture," in *Parco2009 : International Conference on Parallel Computing*, Lyon, France, September 2009.

[9] M. Diouri, M. Dolz, O. Gluck, L. Lefevre, P. Alonso, S. Catalan, R. Mayo, and E. Quintana-Orti, "Solving some Mysteries in Power Monitoring of Servers: Take Care of your Wattmeters!" in *Energy Efficiency in Large Scale Distributed Systems conference (EE-LSDS)*, Vienna, Austria, Apr. 2013.

[10] M. Dias de Assunçao, A.-C. Orgerie, and L. Lefevre, "An analysis of power consumption logs from a monitored grid site," in *Green Computing and Communications (GreenCom), 2010 IEEE/ACM Int'l Conference on Cyber, Physical and Social Computing (CPSCom)*, dec. 2010, pp. 61 –68.

[11] F. Cappello et al, "Grid'5000: A large scale, reconfigurable, controlable and monitorable grid platform," in *6th IEEE/ACM International Workshop on Grid Computing, Grid'2005*, Seattle, Washington, USA, Nov. 2005.

[12] G. Da Costa and H. Hlavacs, "Methodology of Measurement for Energy Consumption of Applications," in *Energy Efficient Grids, Clouds and Clusters Workshop (co-located with Grid), E2GC2 2010, October, 25 - 29, 2010, Brussels, Belgium*. IEEE, October 2010.

[13] R. Ge, X. Feng, S. Song, H.-C. Chang, D. Li, and K. W. Cameron, "Powerpack: Energy profiling and analysis of high-performance systems and applications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 21, no. 5, pp. 658–671, 2010.

[14] M. Steinder, I. Whalley, J. E. Hanson, and J. O. Kephart, "Coordinated management of power usage and runtime performance," in *IEEE/IFIP Network Operations and Management Symposium: Pervasive Management for Ubioquitous Networks and Services, NOMS 2008, 7-11 April 2008, Salvador, Bahia, Brazil*, 2008, pp. 387–394.

[15] K. H. Kim, R. Buyya, and J. Kim, "Power aware scheduling of bag-of-tasks applications with deadline constraints on dvs-enabled clusters," in *Seventh IEEE International Symposium on Cluster Computing and the Grid (CCGrid 2007), 14-17 May 2007, Rio de Janeiro, Brazil*, 2007, pp. 541–548.

[16] G. Varsamopoulos, A. Banerjee, and S. K. S. Gupta, "Energy efficiency of thermal-aware job scheduling algorithms under various cooling models," in *Second International Conference on Contemporary Computing, IC3 2009, Noida, India, August 17-19, 2009*, ser. Communications in Computer and Information Science, vol. 40. Springer, pp. 568–580.

[17] Q. Tang, S. K. S. Gupta, and G. Varsamopoulos, "Thermal-aware task scheduling for data centers through minimizing heat recirculation," in *Proceedings of the 2007 IEEE International Conference on Cluster Computing, 17-20 September 2007, Austin, Texas, USA (CLUSTER 2007)*. IEEE, 2007, pp. 129–138.

[18] M. Dias de Assunçao, J.-P. Gelas, L. Lefèvre, and A.-C. Orgerie, "The green grid'5000: Instrumenting a grid with energy sensors," in *5th International Workshop on Distributed Cooperative Laboratories: Instrumenting the Grid, INGRID 2010*, Poznan, Poland, May 2010, pp. 25–42.

[19] M. Diouri, O. Gluck, L. Lefevre, and F. Cappello, "Energy considerations in checkpointing and fault tolerance protocols," in 2^{nd} *Workshop on Fault-Tolerance for HPC at Extreme Scale (FTXS 2012), co-located with the* 42^{th} *conference on Dependable Systems and Networks.*, Boston, USA, Jun. 2012.

[20] D. Bedard, M. Y. Lim, R. Fowler, and A. Porterfield, "PowerMon: Fine-grained and integrated power monitoring for commodity computer systems," in *Proceedings Southeastcon 2010*. Charlotte, NC: IEEE, Mar. 2010.

[21] J. Black, "Electromigration - a brief survey and some recent results," *Electron Devices, IEEE Transactions on*, vol. 16, no. 4, pp. 338 – 347, apr 1969.