Demystifying Energy Consumption in Grids and Clouds

Anne-Cécile Orgerie*, Laurent Lefèvre* and Jean-Patrick Gelas*
**INRIA - ENS Lyon - University of Lyon - LIP (UMR CNRS, INRIA, ENS, UCB)*
46, allée d'Italie 69364 LYON Cedex 07, FRANCE
Email: annececile.orgerie@ens-lyon.fr, laurent.lefevre@inria.fr, jean-patrick.gelas@ens-lyon.fr

Abstract—**Energy efficiency in large-scale distributed systems has recently emerged as a hot topic. This paper addresses some theoretical and experimental aspects of energy efficiency by putting in perspective some assumptions made in this domain and some observations and analyses. Based on some experimental results and measurements, we revisit and focus on some "truths" commonly assumed concerning the energy usage of servers, the links between resource load and consumed energy, the impact of ON/OFF models, and some wrong assumptions linking energy and virtualization.**

Keywords-**Energy efficiency; large-scale distributed systems; energy awareness; green computing;**

I. INTRODUCTION

Green IT has recently emerged as a new research domain [20]. Numerous papers focus on how to reduce the energy consumption of large-scale infrastructures (e.g. datacenters, Grids, Clouds) and thus improve these systems' energy efficiency. As new research challenges, the energy-efficient approaches must be built on serious and proved assumptions. In the state of the art, we observe some basic and rapid hypotheses regarding energy consumption and efficiency which risk being accepted de facto. This paper lists and clarifies some of these "myths" by confronting them with realistic measurements and analyses we made. This paper focuses on some received ideas associated to energy efficiency in large-scale distributed systems and proposes some measured and concrete facts which could moderate some current approaches[1].

After a quick overview of energy efficiency in large-scale distributed systems in section II, this paper addresses three main aspects in energy efficiency:

- the understanding of the energy usage of Grids and Clouds nodes, and the links between resource usage and energy consumption (section III) ;
- the large-scale deployment of ON/OFF models and their impact on energy reduction and on infrastructures (section IV) ;
- the choice of virtualization as the ultimate approach for energy efficiency (section V).

II. RELATED WORKS

Energy becomes a key challenge in large-scale distributed systems such as Grids [10], [27], [36] and Clouds [31], [24]. These infrastructures require more and more power. Several works are conducted to address this issue at different levels. Some studies are focused on specific node components, e.g., network interface cards [12], storage disks [1], CPUs [30], [9]. In [10], the authors propose a model of energy consumption based on CPU activity. Another approach consists in deducing energy consumption by using event-monitoring counters [3], [21]. Other studies are more general and deal with e.g., ON/OFF algorithms [4], load balancing [21], task scheduling [16], [4], [40] or thermal management [3], [10]. Some works prove that temperature issues are really close to energy issues and show that they belong to the same loop [27], [21]. Indeed, if the nodes' heat production decreases, then so will energy consumption, since the fans and the cooling system will be used less. Thus, in order to reduce energy consumption, the nodes should be shut down or run in slower mode. Finally, virtualization seems to be another promising technique to decrease energy consumption [35], [23], [34] and can be combined with consolidation algorithms [31] and migration options [37], [7].

III. IS IT EASY TO UNDERSTAND AND ANALYSE THE ENERGY CONSUMPTION OF DISTRIBUTED RESOURCES?

A. Does homogeneous servers have the same energy consumption?

Many works assume that homogeneous nodes have the same power consumption [32], [18]. In practice, we observe a different reality. We have dynamically collected the consumption in Watts of 3 sets of homogeneous servers: two IBM eServer 326 (2.0 GHz, 2 CPUs per node), two Sun Fire v20z (2.4 GHz, 2 CPUs per node) and two HP Proliant 385 G2 (2.2 GHz, 2 dual core CPUs per node). To measure the real consumption of some machines, we use external watt-meters. With this infrastructure, we can collect one measure per second per machine. Figure 1 shows that homogeneous servers have different consumptions for different kind of activities which can be representative of the life of server nodes: nodes switched off (but plugged in the wall socket), booting, having intensive disks accesses

---

[1] Some experiments of this article were performed on the Grid5000 platform, an initiative from the French Ministry of Research through the ACI GRID incentive action, INRIA, CNRS and RENATER and other contributing partners (http://www.grid5000.fr). This research is supported by the INRIA ARC GREEN-NET project (http://www.ens-lyon.fr/LIP/RESO/Projects/GREEN-NET/).

978-1-4244-7614-5/10/$26.00 ©2010 IEEE

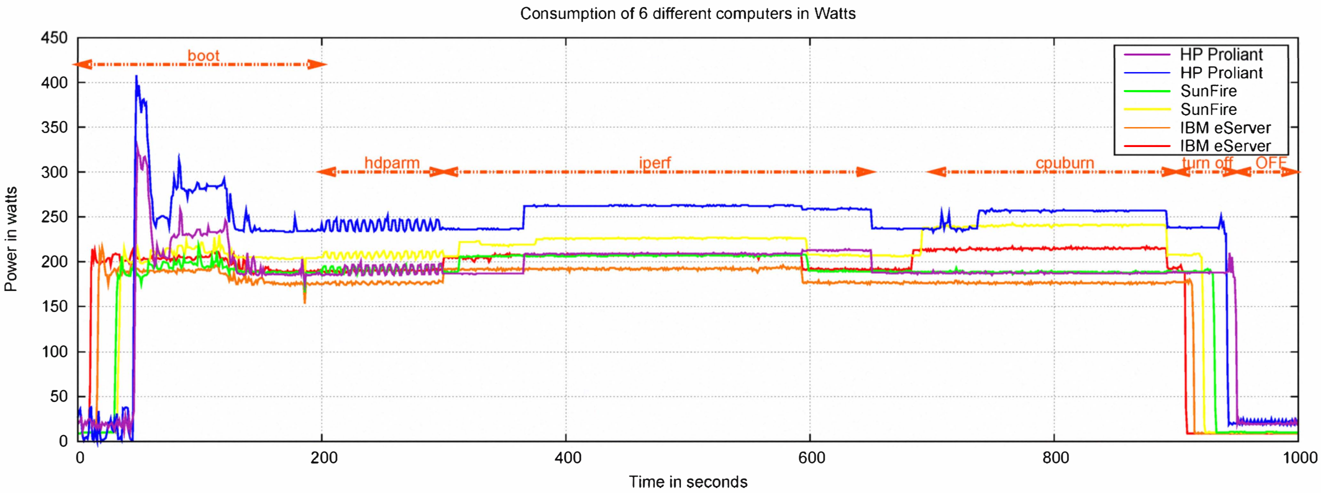

Figure 1. Consumption of 6 servers running typical applications

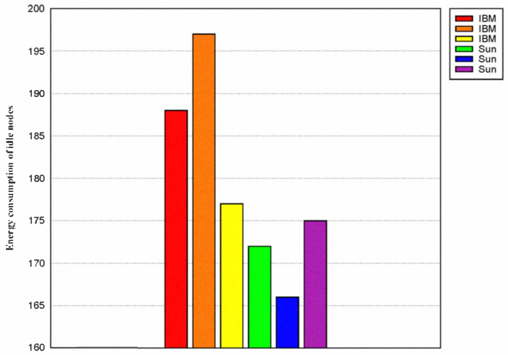

Figure 2. Energy consumption of 6 nodes when they are idle

(`hdparm`), experimenting intensive high-performance network communication (`iperf`), or having intensive CPU usage (`cpuburn`). Figure 2 shows the servers' consumption when they are idle (doing no job but switched on) since a long period of time. In the remainder of this paper, we will call this consumption the *idle consumption* of a node. Let us point out that this consumption can be important compared to the consumption when the node performs a task. The consumption of a node does not depend only on its architecture and on the application it is running. It also depends on e.g., its position on the rack and its temperature. A cooler node will consume less energy, since it will start its fans less often. According to [10], fans can represent 5% of the consumption of a typical server. We conducted another experiment to show that the consumption of a node can be influenced by many factors. We measured the consumption of a typical server (IBM eServer 326) in a rack full of running servers. Its idle consumption was 200 Watts. We then switched off all the nodes in the rack but this one, and we waited one hour to let the nodes cool down. The idle consumption after that was 189 Watts. This represents a 5,5% difference, which is not negligible. A node's consumption can be influenced by various and numerous factors which should not be neglected, since they lead to consequent variations. The identification of these factors is hard and more work is required before we can propose mathematical models of the link between those factors and the variations in energy consumption.

B. *Does the OS have no impact on energy consumption?*

Computing system manufacturers provide more and more opportunity to manage the power for many of the system's components (*e.g.*, CPUs, hard drives, fans, SATA links, Ethernet adapters). However, taking advantage of these new features requires two things: the right BIOS version and configuration, and a modern operating system. Without them, power saving is compromised [38]. For example, a monolithic kernel like Linux has full control over the system components allowed to be managed. It can regulate its activity and energy consumption to meet thermal or energy constraints. This is commonly done through the standard ACPI interface. A simple experience has been conducted by some members of the Lesswatts project. They measured the overall consumption of a server running the same application using 5 different versions of the Linux kernel (from 2.6.22 to 2.6.26). Thanks to the efforts made by Linux's core developers, a noticeable power consumption reduction is shown (cf. http://www.lesswatts.org/results/server). For a few years now, processors have been able to change their working frequency and voltage on-demand in order to reduce power consumption. Today, this technology[2] is commonly available on modern server nodes, although rarely exploited by default

---

[2] Technologies *Powernow!* and *Cool'n'Quiet* (from AMD), and *Speedstep* (from INTEL), allow reducing tension depending on the frequency, and deactivating unused processor parts.

by the operating systems. First, the *P-states* (processor performance states) define different frequencies supported by the processor. In Linux, the *CPUfreq* infrastructure allows control of the P-states thanks to governors, which decide which available frequency between the minimum and maximum frequencies must be chosen. Second, the *C-states* (processor idle states) propose several CPU idle states. C0 is the operational state, the others are idle states. The higher the number, the less energy consumed by the processor and the more time it takes to become active again. Keeping a processor idle for a long period can allow power savings, but this requires reducing CPU wake ups when the processor is idle by disabling useless services and processes. Since version 2.6.24 of the Linux kernel, there is an option available called *Dynamic ticks* or tickless kernel (`NO_HZ`) which allows waking up the processor only when required. In brief, OS design definitely has an impact on the system's energy consumption [2]. This can be accentuated if OS developers do not take advantage of the features manufacturers provide. In order to save energy, the CPU-cores must be awoken only when required.

C. Is the relation between CPU load and energy consumption linear?

To design and build new data centers and new distributed systems, power-provisioning strategies are required. Those strategies are hard to elaborate, even in the current large-scale distributed systems, due to the lack of power-usage data. Indeed, most facilities lack energy sensors and on-line power-monitoring tools. Studying power usage in such infrastructures is a difficult task. That is why we need power models to estimate the energy consumption. nodes in such large-scale distributed systems. We have seen in Section III-A that a node's energy consumption does not only depend on the node's architecture and on the application it is running. Other factors must be considered, and a more detailed analysis is required. This can be done by studying the energy usage of the node's main components: e.g., CPU, disk, network card. In [10], the authors present a typical server's peak power, per component (CPU, memory, disk, PCI slots, motherboard, fan). These results show that CPUs are the most consuming components. However, each component does not have a fixed energy consumption. It depends on the load experienced by the component. To model the node's overall consumption, we need to understand the link between CPU load and energy consumed. Several papers deal with modeling the CPU's energy consumption based on its load [33], [3], [10]. In [33], authors present an energy model at the instruction level. In [10], authors modelize energy consumption according to CPU activity. Another approach consists in deducing it by using event-monitoring counters [3]. The common idea is often that the CPU's energy consumption is directly proportional to its load [10], [5], [32]. We have done several experiments showing that it is wrong. The experiments have been done on three IBM eServer 326 (2.0 GHz, 2 CPUs per node) and three Sun Fire V20z (2.4 GHz, 2 CPUs per node). All six nodes have two CPUs, thus a 200% usage means that the two CPUs are fully loaded. The CPU load is given by the system (we use the `htop`[3] and `sar`[4] commands). On each of these six nodes, we apply successively three different types of load and each of these loads is going to fully load the nodes. So, the CPU usage is the same for those three loads on the six nodes. We then compare the energy consumed by the six nodes during these three experiments. First experiment shows the electric consumption of the 6 nodes while a `stress`[5] tool is running. We find that the consumption increases by 12% to more than 17% compared to the idle consumption for each node. During this experiment, the load of the CPUs reaches 200% for each node. Second experiement is similar, but with two `cpuburns` (one for each CPU). We see that the energy consumption increases more than in the previous experiment: from 17% to 21%. For the last experiment, we launch 2 `cpuburns` simultaneously and an `iperf`. This puts the load of the CPUs at 200%. However, we see that it only increases the electric consumption by 16% to 23% compared to the idle consumption. These values are smaller than in the previous experiment although we added another task (the `iperf` application) which used the network card interface. We have compared the results of the last three experiments on Figure 3. For three different experiments that fully load the CPUs, the electric consumption reaches widely different values for the two types of node architectures. The difference can reach 14% (compared to the idle consumption of the node), which is not negligible. Although CPU utilization has an impact on power consumption, the impact is not linear as stated in several works [10], [5],

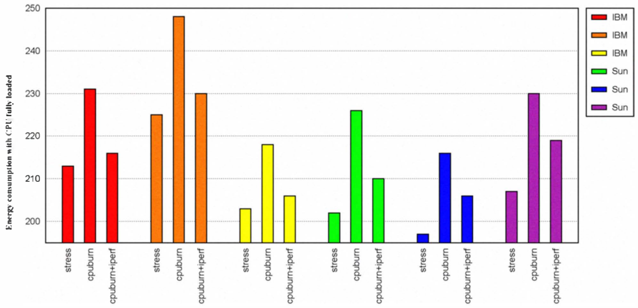

Figure 3. Comparison of the energy consumption of three applications that fully load the CPU.

---

[3] *htop* is a system-monitor tool that produces a frequently-updated list of processes with their CPU-usage percentage.

[4] *sar* is a Solaris-derived system-monitor command used to display CPU activity.

[5] *stress* is a tool used to evaluate the perceived performance under heavy load.

[32]. It is indeed not possible to have a linear function which has three values with the same abscissa (we have three different power consumption values for the same CPU load)! In [39], the authors observe the same phenomenon: the energy consumption model of each application is different. In fact, the relation can be linear under two really restrictive (and not realistic) conditions: (a) the type of workload should be the same (similar utilization of the CPU not like our three kinds of workload); (b) the external environment must be the same (workload and temperature of the neighbouring nodes, external temperature, location in the rack as seen in Section III-A). Moreover, even though tools like `top` shows a CPU usage of 100%, CPU power consumption is not necessary maximal. It may depends of the instructions set in use or the quantity of RAM size accessed by the running application. Then, relation between CPU load and energy consumption can not be linear.

D. Is the relation between network load and energy consumption linear?

A lot of electricity (and money) is wasted to keep network hosts fully powered on, just to maintain their network presence. Proxying techniques are the good approach to solve this issue in an energy-efficient way [17], [8]. Several works aim to reduce the electric cost of networking equipments, from network cards to switches and routers [13], [12]. Indeed, network cards are always powered on (when the node is on) even when they have nothing to do. It represents an important waste of energy with the scaling effects [13]. In [11], the authors describe their algorithm called Adaptive Link Rate (ALR). It changes the link's data rate based on an output-buffer threshold policy. This algorithm does not affect the mean packet delay. This paper shows that the energy used with the different Ethernet link's data rates are not proportional to utilization. In [14], authors conclude their experiments by saying that the energy consumption's dependency on bandwidth is quite small and not linear. They even show that for some network devices, moving data consumes less energy than doing nothing. We have observed similar results with our office's switch. So, some previous works stating that switch power consumption depends on the number of sent bits [41] are no more up-to-date to model the energy consumed by switches. As for CPU, the link between load and energy consumption is not linear in networking devices.

E. Is the relation between disk load and energy consumption linear?

Storage represents a significant percentage of datacenter power [43]. Thus, managing and reducing disk power consumption is a necessary target to decrease the total power consumption of large-scale distributed systems. Some work has been conducted to model and link disk performance and disk power usage [42], [1]. In [15] the authors show that the power used for the standby and idle modes of disks can be reduced significantly. The power consumption of disks is composed of a fixed portion (the idle state which includes the spindle motor) and a dynamic portion (I/O workload, data transfers, moving of the disk head during a seek operation) which represent about one third of the disk's total consumption [1]. Disks offer three type of functions: seeking, reading, and writing. Those three operations have different consumptions [42] since they affect the dynamic portion. In [1], the authors show that the total power consumption of an enterprise disk drive is not a linear function of the number of I/Os per second. This means that less power is consumed relatively per each I/O (seek operation). This phenomenon is due to the fact that a larger number of concurrent I/O requests to the disk increase the internal disk queue and, therefore, I/Os can be reordered so as to shorten the seek distance and thus reduce power consumption. We have seen through the examples of the three main components (CPU, network interface, disk) of the nodes in large-scale distributed systems, that the link between what we use (load and performance) and what we pay (electric consumption) is not linear. This complicates the problem and makes it harder to find an optimal trade-off between performance and electric cost.

IV. SHOULD WE USE LARGE-SCALE ON/OFF AS MUCH AS WE CAN?

A. Is Intensive and systematic ON/OFF the perfect approach for saving energy?

On/Off algorithms are among the first investigated approaches to reduce energy consumption in large-scale distributed systems [4]. The idea is to switch off the unused nodes because the idle consumption is really high (see Figure 2). Moreover, we have seen at the end of Section III-A that switching off some nodes can also reduce the consumption of the neighbour nodes which are still running. For these reasons, it seems a good idea to switch off the nodes as soon as they are not used. However, as shown on Figure 1, node boot can induce significant consumption peaks. Consequently, the infrastructure's global power supply must support all these peaks simultaneously. If this is not the case, alternative solution must be chosen like sequentially perform ON/OFF operations or by tuning server configuration. For example, the BIOS of some servers proposes an option to add a random delay (between 0 and 50 seconds) before switching on the node. Another problem that occurs with intensive ON/OFF algorithms [4], [28] is the energy cost of switching on and off. If the idle time between two jobs is too short, it will consume more energy to switch off and on again than if the node is left idling between the jobs. We call that critical time T_s [26]. In [25], we present EARI (Energy-Aware Reservation Infrastructure) our framework to manage green large-scale distributed systems. EARI embeds some prediction algorithms to predict the next jobs/reservations of resources. They are used at the end of each job to see if these job's nodes should be switched off (if the next predicted job is in more than T_s seconds) or if they should be let on (if the next predicted job is in less than T_s seconds). In [25], we have compared EARI with a different policy: always switching off the nodes at the end of a job if there is no job after (so no prediction). To compare these policies, we have used traces from Grid5000, a French experimental Grid consisting of about 5000 cores geographically distributed over 9 sites in France. The results of this experiment are shown on Figure 4. Idle consumption (P_{idle}) is around 190 Watts on average. But we also show the results for two smaller idle consumption since we hope that in a near future, server constructors will be able to reduce this value. 100% represents the ideal consumption: it is the case where no prediction error is made (i.e., the future is known, which is never the case!). These experiments with different T_s show that in all the cases, energy results are better with prediction. Thus, systematic intensive ON/OFF approach is not always the greener approach. Coordinated energy-consumption models and prediction algorithms could greatly improve ON/OFF models in terms of both energy and performance.

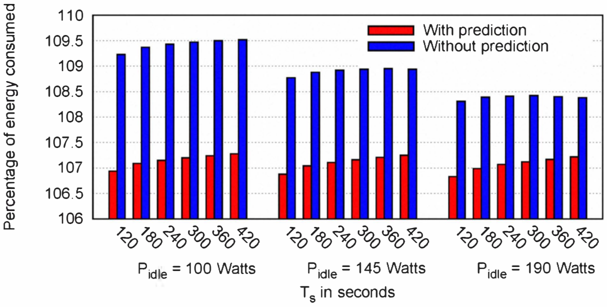

Figure 4. Comparison of energy consumption between the On/Off policy and EARI

B. Does a switched-off node consume energy?

To limit the energy waste, it is important to switch off nodes when they are not used since the idle consumption is still really high even with modern computers. Some works assume that the consumption of switched-off nodes is zero (as stated in [28]). However, we observe that nodes still consume energy when they are off (Figure 5). The consumption of a plugged-node when it is off is called the *off consumption.* For the HP Proliants, the off consumption represents, on average, 10% compared to the idle consumption of these nodes. So, this off consumption is not null, nor negligible, nor stable and depends on the architectures. This can be due to the card controllers embedded in those nodes which are used to wake up the remote nodes. These off consumptions have widely decreased over the server generations, but there is still room for improvements. Indeed, for the two HP Proliant (385 G2), the off consumption represents 15% of the idle consumption. This off consumption must thus be considered when designing algorithms and frameworks to manage resources in an energy-efficient way. Booting and halting a node consume energy by generating peaks of consumption (mostly for the boot startup), as seen on Figure 1. These energy costs may be reduced by using suspend-to-disk or suspend-to-RAM techniques. In [19], some experiments on two notebooks and one desktop machine show that the standby mode (ACPI S1 state, only a few parts of the board are switched off) is not energy efficient compared to the off state. However, it shows that the two notebooks consume as much energy when they are off than when they are in suspend to RAM mode. We plan to study more deeply the energy costs of these techniques. Another mean to reduce the energy costs due to boots and halts, is to reduce the number of such operations by aggregating the jobs in time on the same resources. This is another feature included in EARI [26]: if the user agrees, jobs can be "glued" together (one after the other on the same resources) to save one halt/boot cycle. We demonstrate in [26] on Grid5000 traces that this technique can save significant amounts of energy.

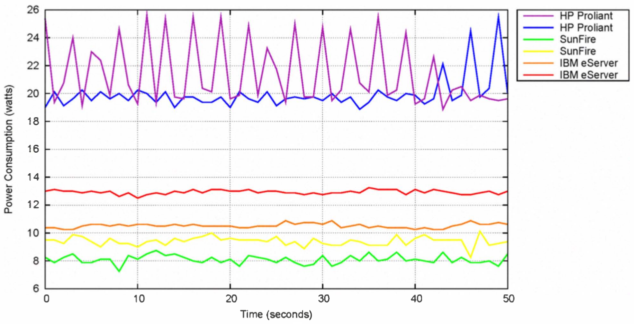

Figure 5. Consumption of 6 nodes when they are off

V. IS VIRTUALIZATION THE PANACEA FOR ENERGY EFFICIENCY?

Virtualization solutions appear as alternative approaches for companies to consolidate their operational services on physical infrastructure, while keeping specific features inside the Cloud perimeter (e.g., security, fault tolerance, reliability).

A. Does virtualization techniques increase energy consumption?

Virtualization presently seems to be the most privileged technique to reduce energy consumption in large-scale distributed systems [35], [23], [31], [34]. Common criticism is that virtualization increases the energy consumption for a given application. We have found no paper in the litterature able to either support or deny this assumption. To clarify this issue, we installed Xen 3.2[6] on a Sun Fire v20z. The idle consumption was identical to that with a GNU/Linux Debian distribution. We then launched a `cpuburn` in a virtual machine (512 MB of RAM and one virtual CPU) and we saw that the energy consumption is the same with or without virtualization (the difference is less than 1 Watt). Our next experiment was to launch two `cpuburn` on the same machine without virtualization and two `cpuburn` in two different virtual machines. We also obtained the same consumptions (the difference is less than 1 Watt). Finally, we did the same experiment with an `iperf` and we observed that, inside a virtual machine, this intensive networking application consumes 7 Watts more on average (this represents 4% of the idle consumption). This effect is due to the poor management of the network allocation in Xen 3.2. The next versions are expected to solve this issue. We conducted the same experiment with a HP Proliant 385 G2 and with Xen Server 5.0 (which is more recent than Xen 3.2) and a Debian. The idle consumptions with the Debian and with a virtual machine on Xen Server 5.0 were still identical. We then did the experiment with a `cpuburn` in a virtual machine (512 MB and one virtual CPU) and a `cpuburn` in a Debian. There was a 2% difference between the two consumptions (the virtual machine experiment was the less consuming one). These experiments lead to conclude that virtualization with a recent hypervisor does not imply any energy overhead for computing tasks. However, we have not studied the time overhead which should be induced by the CPU overhead in the device-driver domain [6] (a virtual machine cannot always run on an entire physical CPU, the resources are shared). As pointed out in [36], virtualization still leads to a waste of resources since the hypervisor needs resources and this reduces the possibility to put several virtual machines on the same node. This issue is not often taken into account in the design of energy-efficient frameworks for virtualized distributed systems.

B. *Is (Live) Migration of virtual machines free?*

In [24], we present the Green Open Cloud, a dynamic energy-aware cloud framework which uses virtualization to allow high performance, improved manageability, fault tolerance and uses migration to bring the benefit of being able to move workload between virtual machines. Live migration [7] greatly improves the capacities and the features of Cloud environments: it facilitates fault management, load balancing, and low-level system maintenance. Migration operations mean more flexible resource management: when a virtual machine is deployed on a node, it can still be moved to another one. It offers a new stage of virtualization by removing the concept of locality in virtualized environments. However, this technique is complex and more difficult to use over MAN/WAN [37] than in a cluster. IP addressing is a problem since the system should change the address of the migrated virtual machine which does not remain in the same network domain. However, this technique is not free in terms of consumption. In Figure 6, six `cpuburn` in six different virtual machines (Xen Server 5.0) are launched at 10 seconds on Cloud node 1 with a one second interval. Then, all the virtual machines are migrated to Cloud node 2. The apparition of the fifth and the sixth jobs does not increase the consumption. Indeed, as the jobs are CPU intensive (*cpuburn* uses 100% of a CPU-core capacity) and as there are only four cores on the node (2 dual core CPUs), they are fully used with the first four virtual machines. The fifth virtual machine appears as “free” in terms of energy cost because it shares already fully used resources. This phenomenon is a form of competition for the physical resources. Each *cpuburn* job lasts 300 seconds (Figure 6). At $t = 110$, we launch the migration of the 6 virtual machines from Cloud node 1 to Cloud node 2. The migration requires sustained attention from the hypervisor that should copy the memory pages and send them to the new host node. For this reason, it cannot handle 6 migrations at the same time, so they are done one by one. The competition occurs and we see with the power consumption of Cloud node 2 that the virtual machines arrived one by one. The consumption of Cloud node 1 begins to decrease during the migration of the third virtual machine. At that time, only three virtual machines are still running on the node. Each job ends 5 seconds late. The competition on resources that occurs during the migration request does not affect more the jobs running on the last migrated virtual machines since they are still running while they wait for the migration. There is an “expensive” moment in terms of energy during the migration when the two nodes consume energy for the same virtual machine. So, migration energy cost should not be neglected as it is sometimes implicitly the case [22]. Moreover, this first result does not include the network’s energy usage. Our typical application does not require disk usage and data, so the virtual machine is lightweight and thus fast to migrate.

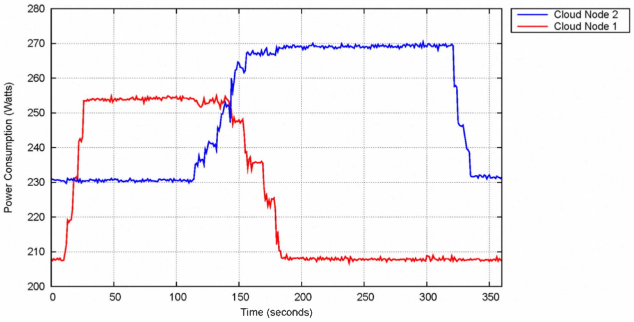

Figure 6. Migration of 7 Virtual Machines

---

[6] Xen is a virtual-machine monitor that allows several guest operating systems to run concurrently on the same computer hardware.

VI. CONCLUSIONS AND FUTURE WORKS

IT energy consumption "only" represents 3 to 5% of CO_2 emission in the world which is similar to the aviation transport. While small in amount, this usage is symbolic because IT can greatly influence by their solutions other industrial and research domains [29]. We have shown that understanding and modelizing the energy consumption of large-scale systems infrastructure is a complex task depending on various contexts (e.g., location, usage...). Using intensive ON/OFF techniques is an important approach to reduce energy consumption. But these solutions must be performed in an intelligent and coordinated way. Virtualization solutions should be carefully taken into account when dealing with energy reduction. This paper does not present an exhaustive list of aspects of energy efficiency. Some open questions and myths remain. The next open questions we plan to address are: (i) Is CO_2 the right metric to measure and expose the energy usage of IT infrastructures? (ii) Do we still need to support the same quality of experiment while reducing energy usage or do we need to enforce energy-aware policies?

REFERENCES

[1] M. Allalouf, Y. Arbitman, M. Factor, R. I. Kat, K. Meth, and D. Naor. Storage modeling for power estimation. In *SYSTOR '09: Proceedings of SYSTOR 2009: The Israeli Experimental Systems Conference*, pages 1–10, New York, NY, USA, 2009. ACM.

[2] K. Baynes, C. Collins, E. Fiterman, B. Ganesh, P. Kohout, C. Smit, T. Zhang, and B. Jacob. The performance and energy consumption of three embedded real-time operating systems. In *CASES '01: Proceedings of the 2001 international conference on Compilers, architecture, and synthesis for embedded systems*, pages 203–210, New York, NY, USA, 2001. ACM.

[3] F. Bellosa, S. Kellner, M. Waitz, and A. Weissel. Event-Driven Energy Accounting for Dynamic Thermal Management. In *Proceedings of the Workshop on Compilers and Operating Systems for Low Power (COLP'03)*, pages 1–10, Sep. 2003.

[4] 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 *SOSP '01: 18th ACM symposium on Operating systems principles*, pages 103–116, New York, NY, USA, 2001. ACM.

[5] G. Chen, W. He, J. Lie, S. Nath, L. Rigas, L. Xiao, and F. Zhao. Energy-aware server provisioning and load dispatching for connection- intensive internet services. *NSDI'08: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation*, pages 337–350, Berkeley, CA, USA, 2008. USENIX Association.

[6] L. Cherkasova and R. Gardner. Measuring CPU overhead for I/O processing in the Xen virtual machine monitor. In *ATEC '05: Proceedings of the annual conference on USENIX Annual Technical Conference*, pages 24–24, Berkeley, CA, USA, 2005. USENIX Association.

[7] C. Clark, K. Fraser, S. Hand, J. G. Hansen, E. Jul, C. Limpach, I. Pratt, and A. Warfield. Live migration of virtual machines. In *NSDI'05: Proceedings of the 2nd conference on Symposium on Networked Systems Design & Implementation*, pages 273–286, Berkeley, CA, USA, 2005.

[8] G. Da-Costa, J.-P. Gelas, Y. Georgiou, L. Lefèvre, A.-C. Orgerie, J.-M. Pierson, O. Richard, and K. Sharma. The green-net framework: Energy efficiency in large scale distributed systems. In *HPPAC 2009 : High Performance Power Aware Computing Workshop in conjunction with IPDPS 2009*, Rome, Italy, May 2009.

[9] H. Dietz and W. Dieter. Compiler and runtime support for predictive control of power and cooling. *Parallel and Distributed Processing Symposium, International*, 0:345, 2006.

[10] X. Fan, W.-D. Weber, and L. A. Barroso. Power provisioning for a warehouse-sized computer. In *ISCA '07: Proceedings of the 34th annual international symposium on Computer architecture*, pages 13–23, New York, NY, USA, 2007. ACM.

[11] C. Gunaratne and K. Christensen. Ethernet adaptive link rate (alr): Analysis of a buffer threshold policy. In *Proceedings of IEEE GLOBECOM*, pages 1–6. IEEE, Dec. 2006.

[12] C. Gunaratne, K. Christensen, and B. Nordman. Managing energy consumption costs in desktop pcs and lan switches with proxying, split tcp connections, and scaling of link speed. *Int. J. Netw. Manag.*, 15(5):297–310, 2005.

[13] M. Gupta and S. Singh. Greening of the internet. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 19–26, New York, NY, USA, 2003. ACM.

[14] H. Hlavacs, G. D. Costa, and J.-M. Pierson. Energy consumption of residential and professional switches. *Computational Science and Engineering, IEEE International Conference on*, 1:240–246, 2009.

[15] A. Hylick, R. Sohan, A. Rice, and B. Jones. An analysis of hard drive energy consumption. In *Modeling, Analysis and Simulation of Computers and Telecommunication Systems, 2008. MASCOTS 2008. IEEE International Symposium on*, pages 1–10, Sept. 2008.

[16] R. Jejurikar and R. Gupta. Energy aware task scheduling with task synchronization for embedded real-time systems. In *Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on*, pages 1024– 1037. IEEE, June 2006.

[17] M. Jimeno, K. Christensen, and B. Nordman. A Network Connection Proxy to Enable Hosts to Sleep and Save Energy. In *IEEE International Performance, Computing and Communications Conference (IPCCC 2008)*, pages 101–110, Piscataway, NJ, USA, 2008. IEEE Press.

[18] K. H. Kim, R. Buyya, and J. Kim. Power Aware Scheduling of Bag-of-Tasks Applications with Deadline Constraints on DVS-enabled Clusters. In *IEEE International Symposium on Cluster Computing and the Grid (CCGRID 2007)*, pages 541–548, May 2007.

[19] K. Knopper. Got the Power: Reality and myth in the quest for green computing . *Linux Pro Magazine*, pages 31–35, 2008.

[20] L. Lefèvre and J.-M. Pierson. *Special theme ERCIM News : Towards Green ICT*, volume 79. Oct. 2009.

[21] A. Merkel and F. Bellosa. Balancing power consumption in multiprocessor systems. In *SIGOPS Operating Systems Review*, 40(4):403–414, 2006.

[22] M. Milenkovic, E. Castro-Leon, and J. R. Blakley. Power-Aware Management in Cloud Data Centers In *Lecture Notes in Computer Science (LCNS) - Cloud Computing*, volume 5931/2009, pages 668–673, 2009. Springer.

[23] R. Nathuji and K. Schwan. Virtualpower: coordinated power management in virtualized enterprise systems. In *SOSP '07: Proceedings of twenty-first ACM SIGOPS symposium on Operating systems principles*, pages 265–278, New York, NY, USA, 2007.

[24] A.-C. Orgerie and L. Laurent. When clouds become green: the green open cloud architecture. In *International Conference on Parallel Computing (Parco 2009)*, Lyon, France, Sept. 2009.

[25] A.-C. Orgerie, L. Lefèvre, and J.-P. Gelas. Chasing gaps between bursts : Towards energy efficient large scale experimental grids. In *PDCAT 2008 : The Ninth International Conference on Parallel and Distributed Computing, Applications and Technologies*, Dunedin, New Zealand, Dec. 2008.

[26] 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 *14th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, pages 171–178, Melbourne, Australia, Dec. 2008.

[27] C. Patel, R. Sharma, C. Bash, and S. Graupner. Energy Aware Grid: Global Workload Placement based on Energy Efficiency. Technical report, HP Laboratories, November 2002.

[28] K. Rajamani, and C. Lefurgy. On Evaluating Request-Distribution Schemes for Saving Energy in Server Clusters. *In Proceedings of the IEEE International Symposium on Performance Analysis of Systems and Software*, pages 111-122, 2003.

[29] S. Ruth. Green IT - More Than a Three Percent Solution? *IEEE Internet Computing*, 13(4):74–78, 2009.

[30] D. C. Snowdon, S. Ruocco, and G. Heiser. Power management and dynamic voltage scaling: Myths and facts. In *Proceedings of the 2005 Workshop on Power Aware Real-time Computing*, New Jersey, USA, Sep 2005.

[31] S. Srikantaiah, A. Kansal, and F. Zhao. Energy aware consolidation for cloud computing. In *Proceedings of HotPower '08 Workshop on Power Aware Computing and Systems*, December 2008.

[32] M. Steinder, I. Whalley, J. E. Hanson, and J. O. Kephart. Coordinated management of power usage and runtime performance. In *IEEE Network Operations and Management Symposium (NOMS)*, pages 387 - 394, April 2008.

[33] S. Steinke, M. Knauer, L. Wehmeyer, and P. Marwedel. An accurate and fine grain instruction-level energy model supporting software optimizations. In *in Proc. Int. Wkshp Power and Timing Modeling, Optimization and Simulation (PATMOS)*, 2001.

[34] J. Stoess, C. Lang, and F. Bellosa. Energy management for hypervisor-based virtual machines. In *ATC'07: 2007 USENIX Annual Technical Conference on Proceedings of the USENIX Annual Technical Conference*, pages 1–14, Berkeley, CA, USA, 2007. USENIX Association.

[35] R. Talaber, T. Brey, and L. Lamers. Using Virtualization to Improve Data Center Efficiency. Technical report, The Green Grid, 2009.

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

[37] F. Travostino, P. Daspit, L. Gommans, C. Jog, C. de Laat, J. Mambretti, I. Monga, B. van Oudenaarde, S. Raghunath, and P. Y. Wang. Seamless live migration of virtual machines over the man/wan. *Future Gener. Comput. Syst.*, 22(8):901–907, 2006.

[38] A. Vahdat, A. Lebeck, and C. S. Ellis. Every joule is precious: the case for revisiting operating system design for energy efficiency. In *EW 9: Proceedings of the 9th workshop on ACM SIGOPS European workshop*, pages 31–36, New York, NY, USA, 2000. ACM.

[39] A. Verma, P. Ahuja, and A. Neogi. Power-aware Dynamic Placement of HPC Applications. In *ICS '08: Proceedings of the 22nd annual international conference on Supercomputing*, pages 175–184, 2008. ACM.

[40] G. von Laszewski, L. Wang, A. Younge, and X. He. Power-aware scheduling of virtual machines in dvfs-enabled clusters. In *IEEE International Conference on Cluster Computing and Workshops (CLUSTER '09).* , pages 1–10, Sept. 2009.

[41] T. T. Ye, G. Micheli, and L. Benini. Analysis of power consumption on switch fabrics in network routers. In *DAC '02: Proceedings of the 39th annual Design Automation Conference*, pages 524–529, 2002. ACM.

[42] J. Zedlewski, S. Sobti, N. Garg, F. Zheng, A. Krishnamurthy, and R. Wang. Modeling hard-disk power consumption. In *FAST '03: Proceedings of the 2nd USENIX Conference on File and Storage Technologies*, pages 217–230, Berkeley, CA, USA, 2003. USENIX Association.

[43] Q. Zhu, F. M. David, C. F. Devaraj, Z. Li, Y. Zhou, and P. Cao. Reducing energy consumption of disk storage using power-aware cache management. *High-Performance Computer Architecture, International Symposium on*, 2004.