Energy: A new criteria for performances in large scale distributed systems

Jean-Marc Pierson

IRIT
University of Toulouse
pierson@irit.fr

Abstract. In the framework of the workshop ”Performance Evaluation of Computer and Communication Systems: Milestones and Future Challenges”, this paper proposes to address the emerging criteria of energy. Used since a long time in embedded systems where battery operated devices needed a careful handling, the energy metric is taking a large momentum in the last years on large scale systems where thousands of nodes collaborate to serve high end infrastructures like web servers, clouds and grids.

1 Introduction and motivation

Since the last 5 years, we witness the raise in interest for energy aware infrastructures and computing in large scale systems. What appeared at its beginning as a hype is slowly taking more importance in the everyday life when operating large scale systems. Beside the ecological view coming from the carbon-related global warning concern, attraction is also garnered by several other actors: CEOs and system administrators handling large IT infrastructures caring for their electrical budget or their electricity cap, electricity providers who need to serve optimally a growing demand, and finally computer and mathematical scientists who see an opportunity to explore a new scientific field.

The demand in research in energy-efficiency in large scale systems is supported by several incentives [29, 6, 33], including financial incentives by government or institutions for energy efficient industries / companies [28]. Indeed, studies like [4] report that the IT consumption accounts between 5 to 10% of the global growing electricity demand, and for a mere 2% of the energy. Data centers hosting web services or cloud computing gather thousands of nodes and every single Watt saved on each machine every second is making a huge difference at the end on the energy bill (either money ROI or ecologically speaking).

While some investigate how to reduce the ecological impact [30], most of the works are driving research for reducing electricity demands in terms of Watts. This paper is focusing on this latter and puts in the perspective the new challenge of energy consumption as a criteria for performances in large scale distributed systems. It is intended to serve as an introduction to the relevant metrics and appropriate methods when one address energy as a criteria to performance evaluation.

This paper is organized as follows. Section 2 reviews some of the standard metrics for performance evaluation in large scale systems. Section 3 is concerned with energy metrics and energy benchmarks before investigating the integration of energy concerns in section 4. Section 5 concludes the paper.

2 Performance metrics in large scale systems

Since the early ages of computer science, performance evaluation is related to the number of operations being done during a certain duration. Manufacturers and computer scientists have investigated ways to augment the performances of silicon chips and their usage.

Since the conjecture of Moore in 1965 [26] and refined in 1975 [27] stating that the number of transistors in a chip will double every two years, valid for over 40 years (now we can see its limit), the emphasis is put on the number of operations that can be done during one time interval. While MIPS (Million Instructions per Seconds) was a common measure in first ages to measure the performances of processors it is not anymore that important: Indeed, this measure does not take into account many aspects of a computer not related to CPU, like memory hierarchies or input/output bandwidth, or real application workflows. The preference is now to measure on benchmarks the performances of a computer, from single mono-core node to multicore many-nodes architectures. Specific benchmarks exist to stress the architectures on their CPUs, their memory, their network, their disks. For CPU, we measure performances in terms of number of operations per second (particularly floating point operations per second, or Flops), on well known benchmarks. For instance, SPEC (Standard Performance Evaluation Corporation) [9] and TPC (Transaction Processing Performance Council) [11] both have a collection of benchmarks to evaluate different architectures and applications. Energy concerns are present in these benchmarks as we will detail later in section 3.

The Linpack benchmark [13] is the building brick of the Top500 list [14] that enlists every six months the most powerful operational (super-)computers. Even if not perfect since it does not encompass all the applications characteristics (it is a matrix computation benchmark made for scientific and simulation codes), this benchmark has been accepted by the high performance computing community to compare machines. As for an example, the first in the list is reaching 1.7 Petaflops ($1.7 * 10^{15}$ flops). Another benchmark proposed by J. Dongarra is the HPCC (High Performance Computing Challenge) [24], which encompasses more aspects than just Linpack (but still includes it). Unfortunately, as we will see in section 3 it does not take into account the energy spent for the calculation.

For some time largely ignored by the growing capacities of monocore CPUs, parallelism is coming back on stage due to problems related to physical limits in the chips design and the programmed end of Moore's law. Indeed, smaller integrated chips using CMOS technology induce a heat dispersion problem, due to the existing relations between speed, frequency, voltage and energy (thus heat). Multicore CPUs are now the norm, with trends to build on hundred cores (already produced) or even thousand core CPUs. All the knowledge gained in the late 80s from parallel computing garners a new interest with these new highly integrated architectures.

Many metrics have been built in order to check different aspects of parallel computing and distributed systems.

In the field of parallel programming, the classical metric is the speed-up, which gives the acceleration in terms of time to solution (TTS) of a parallel program using P processors against the same program using 1 processor. It is simply the sequential time over the parallel time. As a first approach, we can already see that the faster a program completes, obviously the less energy it uses.

Another metric is the scale-up. This metric helps to figure out the impact of processing bigger problems with more processors. In the best case, P processors should be able to process a problem of size $P * N$ in the same time than 1 processor is processing a problem of size N. In terms of energy concerns, the scale-up translates to the fact that obviously more processors will consume more energy than one.

Fairness is insuring that the different jobs are processed equally and if there is no starvation. Most of the time a job is processed when allocated all its required resources. Another mean is to allocate only part of the resources. The yield is characterizing how much resources are allocated to a job in comparison with what was asked by this job. Reaching a high minimum yield assures some fairness in the system. We will come back to this notion later in section 3.1 since it allows to trade resources (thus time to solution) for energy. A problem resides in the fact that the resource consumption have to be monitored and may cause overhead when done dynamically.

Other criteria of performances include throughput, latency, user satisfaction, security. Service Level Agreement (SLA) have appeared lately to handle the required level of processing and resources that meets specified needs. In terms of energy awareness, we will see how SLA translates in most common works.

3 Energy performance of large scale systems

While a major topic since several years in embedded systems with battery operated devices, energy-awareness raised interest since only a few years for large scale systems like super-computers, clusters, grids and clouds. For a long time, energy consumption has simply been ignored in the performance evaluation in parallel architectures, parallel programming, and lately grid computing.

For the sake of comparison, the first UNIVAC I (UNIVersal Automatic Computer) machine in the 1950's was consuming 125 kW for 1905 operations per second[1]. Today's best as mentioned before is reaching 1.7 Petaflops at a cost of 6950 kW.

In 2008, a survey looking back on the TOP500 list elaborates a classification for today's best [25]. The author proposes a new class of Power Efficient Systems with the rise of architectures (and vendors) taking energy concerns seriously.

A common office computer consumes between 120 and 200 Watts while high end servers consumes between 80 to 300 Watts. Several studies [16, 23] split the share of the energy consumption in a computer as this: CPU accounts for 37% of energy consumption, memory is 17%, PCI slots are 23%, motherboard is 12%, while disk at 6% and fans with 5% are closing the list. Note here that this does not include the power

[1] source: Wikipedia

supply (which is accounting for more or less 20% loss), the networking infrastructure and all the cooling infrastructure. In data-centers for instance the cooling can consume as much electricity than the computers themselves.

Energy consumption can be considered in mainly two directions: The energy efficiency identifies the performance of the system with respect to the energy costs (Section 3.1 on metrics, Section 3.2 on benchmarks). Another direction is to put either energy savings or performances metrics as optimization goals and others as constraints, or to check for Pareto solutions taking into account both (for instance the power yield in Section 3.1 or in placement and scheduling techniques, see Section 4.3), knowing hardware possibilities (Section 4.1).

3.1 Energy metrics

The first immediate metric that has been used coming from Flops is Flops/Watt. The idea is to measure the number of flops that can be achieved using one Watt. Simplistic enough, this metric has the merit to be easily understandable and related to its ancestor in Top500. It is used by the Green500 listing (see section 3.2). A problem mentioned with this metric is that it measures the power, but not the energy spent. The power view is instantaneous while energy E relates to power P over a period of time t: $E = P * t$.

Hence, two obvious way can be used to reduce energy consumption: Either by reducing power consumption of the computers, or by reducing time to produce the result. When an infrastructure is always on with the same power consumption factor in average, the time is not an issue. In this situations the Flops/Watt metric makes sense. Less related to number of Flops, a metric considers the number of operations (not only flops) per Watt. Conversely, another approach is to measure the average power the infrastructure needs to achieve a given operation. The Spec-power and TPC-Energy benchmarks (see section 3.2) are using these metrics, respectively.

Still a difficulty appears: Do the maximum power or the average power have to be taken into account? Nowadays, many components have internal or software means to reduce the power consumption (see section 4.1) hence the course of power consumption over time can have big variations. Some infrastructures even rely on unused nodes switch-off to zero power consumption of a set of nodes (see section 4.3).

Another metric used is the energy itself when accounting for the energy of finite applications. The idea is to measure the Time To Solution and the consumed (max, average) power. The result of the multiplication is expressed in Joules (Watts.s) or Watt.hour.

Metrics can be elaborated from other traditional metrics from parallel systems. We can imagine speedup per Watt, scale-up per Watt or any combination of these.

Example of a metric: The power yield In [5] we extend for instance the yield defined in [35] so that the most energy efficient machines for hosting a set of jobs are chosen. The consumption reduction problem translates to multi-objective optimization problem, where energy adds to CPU, memory (for this work), network bandwidth, and others as constraints of the problem.

Our approach is to rely on the demands a job has on the infrastructure (in terms of CPU and memory bounds), to satisfy at best these demands while minimizing the energy. More formally, [35] defined the yield Y_i of a job i as $Y_i = \sum_{j=1}^{H}(\frac{\alpha_{ij}}{\alpha_i})$ where H is the number of hosts, α_i is the CPU demand of job i and α_{ij} is the allocated part of the CPU of host j for job i. Hence, when a job is allocated what it demands, $Y_i = 1$. Y_i reflects the satisfaction of job i. The memory of a job is considered a rigid need: No allocation can be achieved if a memory need for a job can't be satisfied because of lack of memory.

We extended this definition using a power factor E_{ij}, reflecting the power cost of job i when running on host j. We defined $E_{ij} = \lambda(C_j^{max} - C_j^{min}) \times \alpha_{ij} + (1 - \lambda)\left[A_j(1 - \sum_{i'=1,i'\neq i}^{J}(\alpha_{i'j}))\right]$. The first term $(C_j^{max} - C_j^{min}) \times \alpha_{ij}$ reflects the extra power cost involved by the presence of job i on host j (with C_j^{max} and C_j^{min} being the maximum and minimum power of host j at full charge and when idle). The second term $A_j(1 - \sum_{i'=1,i'\neq i}^{J}(\alpha_{i'j}))$ reflects the attraction of host j for job i at a given time, when other jobs are already placed on it (with A_j an attraction factor that favors the consolidation of the jobs to the (already used) hosts that are consuming less power, for instance we took $A_j = C_j^{min}$). The parameter λ balances the effect of placement (first term, favoring the most energy efficient hosts) and the effect of consolidation (second term, favoring a minimal number of hosts switched-on).

Finally we defined YE_{ij} (the power yield) as $YE_{ij} = \frac{Y_i^{(1-k)}}{E_{ij}^k}$. The k parameter balances between performance (satisfaction) and power E_{ij}. This metric is used in an optimization heuristic to minimize the number of hosts while guaranteeing a given quality of service and/or energy reduction, taking into account additionally the memory used by one job.

As we can see from this example, to capture the effects of the placement of jobs on a set of hosts while still guaranteeing a level of performances is a tedious task: The process must take into account the specificities of the jobs (their demands in CPU and memory), the specificity of the infrastructure (the hosts characteristics, possibly the interconnect), the interactions between jobs (possibly the communications). As for now, an unique metric taking into account all the parameters for task allocation is not existing.

Other metrics The GreenGrid alliance proposes to use the PUE for data centres infrastructures [34, 18]. The PUE (Power Usage Effectiveness) is computed by dividing the amount of power entering a data center by the power used to run the computer infrastructure within it. It encompasses all the surrounding of the infrastructure, including power supplies, chillers, air conditioning. For instance best practices data centres can reach a PUE down to 1.1 while the average data-centers have a PUE of about 1.9.

Finally, one can either optimize the total energy used, or the energy cap. This latter is related to the maximum power consumption over a small period of time. In all infrastructures, the electricity provided by the energy providers is limited, due to physical constraints of the power (electricity) distribution network. A metric is to measure the maximum electricity that can be used in the infrastructure, in case of high workload and extreme situations (for instance when cooling is used at its maximum during hot periods).

3.2 Energy benchmarks

The Green500 [8] initiative is challenging the most powerful machines in terms of Flops/Watts. In the same manner than the Top500, Linpack is used to compute the performance. The first 3 (numbers from October 2010) in the ranking achieve a 773 MFlops/Watt, with a total consumption of the corresponding machines of 57.54 kW. Notably, 8 out of the 10 first such machines use accelerators (PowerXCell from IBM, GPU from AMD ATI and Nvidia). The Green500 is exploring at the moment a new list, based on the HPCC benchmark.

SpecPower [32] is an industry-standard benchmark that evaluates the power and performance of servers and multi-node computers. The initial benchmark addresses only the performance of server side Java. It exercises the CPUs, caches, memory hierarchy and the scalability of shared memory processors (SMPs) as well as the implementations of the JVM (Java Virtual Machine), JIT (Just-In-Time) compiler, garbage collection, threads and some aspects of the operating system. It computes the overall server-side java operations per Watt, including the idle time on specific workloads. The comparison list includes 172 servers. Among these, a Fujitsu server with 76 quadcores (304 cores) reach a maximum value of 2927 ssj_ops/Watt.

The TPC (Transaction Processing Performance Council) proposes the TPC-Energy benchmark [15] for transactional applications: Web/Application Services, Decision Support, On Line Transaction Processing. TPC measures Watts/operations on the TPC benchmarks (for instance transaction per seconds). Only few servers from HP have now been evaluated. As an example, 5.84 Watts/transaction per seconds is given for an typical online transaction processing workload.

For more details and comparative studies on these energy benchmarks, the reader can refer to [31].

The EEMBC (Embedded Microprocessor Benchmark Consortium) has a similar approach. It is providing a benchmark for energy consumption of processors [2]. It is mainly dedicated for embedded systems and computes number of operations per joule linked to the over performance benchmarks of the consortium, measured on different standard applications for embedded systems.

Manufacturers provide information about the consumption of their components, using average loads. For instance, AMD describes the ACP (Average CPU Power) [1] that characterizes power consumption under average loads (including floating point, integer, java, Web, memory bandwidth, and transactional workloads, subset of TPC and SPEC benchmarks). Interestingly, this work shows for instance that cores can consume between 61% to 80% of the processor power, and processors consume less than 25-35% when idle.

4 Integrating energy concerns

Energy concerns have been integrated in many works at hardware, network, middleware, and software levels in large scale distributed systems. This section does not intend to provide an exhaustive view of these works, but rather representative trends followed by researchers worldwide. For further reading, eEnergy Conference proceedings [22] or the COST IC0804 proceedings [21] are providing good insights.

4.1 Hardware level

Hardware has long been thought as the main (and often only) place where energy savings can be achieved at large. Manufacturers believed (and still believe) that more energy efficient hardware is the key issue, and that quicker processors will achieve the energy saving goals (working more on the t parameter of the energy formula of section3.1). Much developments have been achieved at the processors, memory, motherboards, network card, etc. levels. In [7], the COST Action IC0804 surveys such hardware leverages for energy. As an example, the Performance (P) and CPU (C) states in ACPI compliant components allow to reduce the power consumption in deep sleep or halt modes on processors almost to nothing (down to 2 Watts). These states can be controlled also by software, opening door to Dynamic Voltage Frequency Scaling (DVFS) or efficient consolidation techniques (see section 4.3).

Another aspect at hardware level is the problem of collecting energy usage. Virtually all the researchers are using different power meters: Some use oscilloscopes to measure directly the amperage and voltage at the hardware component level, while the majority in large scale systems are using external power meters. Nowadays, intelligent Power Distribution Units (PDU) allow to collect electricity demands at the plug, distribute it on several servers or nodes and aggregate values to send back to any interested party (like the middleware or software for instance). For instance the GreenNet infrastructure allowed us to measure over a long period of time the pattern of the usage of the Grid5000 platform and to propose appropriate enhancements at the middleware level [12]. Also, the TPC introduced an EMS (Energy Management Systems) to accompany its benchmark. The main advantage of such power meter is that they are not intrusive: They do not change the behavior of the observed nodes. Their drawback is the data acquisition frequency, (often one second), which is far too little when one wants to tune precisely operating systems for instance (where the quantum is order of magnitude smaller).

4.2 Network level

At the network level, initial works investigated the energy consumption at the hardware level, trying to correlate between network activity and energy spent [20]. Results show that the traffic has little influence on the power consumption compared to the actual switch on of modules or plugs in the switches and routers. The metrics measured the number of bytes per Watt or the achievable bandwidth per Watt. Other works optimize routes so as to prefer less energy demanding technologies (like optical networks) or change dynamically the characteristics of these networks. In Ethernet networks, Adaptive Link Rate [19] is a solution where energy savings can be obtained by quickly changing the speed of network links in response to the amount of data transmitted.

4.3 Middleware level

At the middleware level, two main complimentary solutions are used: Dynamic hardware adaptation and consolidation. Dynamic hardware adaptation mainly relies on the DVFS capacities. Dynamic Voltage and Frequency Scaling tries to gain energy on inactivity phases. The idea is to find the right clock for the right task. Since $power =$

$voltage^2 * frequency * \alpha$, with α as a hardware and conditions related parameter, reducing frequency and/or voltage allows for a large spare of energy. As we saw in section 4.1, different combinations of frequency/voltage can be tuned by software (P-C states). The metric is then evaluated against the frequency/voltage combination during the course of a particular application or a lifetime of an infrastructure.

Most of the works are doing consolidation using virtual machines to embed the jobs: In this approach, the main issue is to switch off as much hosts as possible to save energy while still guarantying quality of services. Techniques vary in the choice of the hosts since it depends on each application and on possible links between jobs distributed among the platform. The jobs do not have the same energetic behavior on every node (depending on their CPU, memory, I/O accesses). Moreover if communications occur between jobs, the techniques tend to collocate them on the same hosts. To evaluate the different techniques, the metric is related to the dynamic number and consumption of hosts for the application.

Example with a job allocation formalization and methodology We proposed in [5] to model the problem of task allocation as a mixed integer linear problem, putting in equation the different constraints of the task allocation process. For instance, using the same notations than in section 3.1, one equation is $\forall i \sum_h \alpha_{ih} \leq \alpha_i$, with α_i is the CPU demand of job i and α_{ih} is the allocated part of the CPU of host h for job i. Afterwards, we defined several problems to be solved, given these constraints: To maximize the minimum yield of the jobs (definition of yield Y as in section 3.1), to minimize the energy (defined as the sum of the dynamic power of the switched-on hosts), and to find a tradeoff between them.

We will not detail here the model itself (due to limitation of space) but we will focus on the methodology and results. Once modeled, the problems can be solved using a mixed integer linear program, finding an optimal: minimal set of efficient hosts to run the jobs so as to guarantee the constraints. The resolution of this NP-hard problem proved to be much too long for real life instances (over 4 hosts and 12 jobs). Therefore we studied different heuristics (greedy-like, binpacking-like, based on the metric described in section 3.1) and compared them with bounds on the optimal. We came to the result that it is possible to find near-optimal placements of jobs in less than 1 second for instances of the problem with 500 hosts and 1500 jobs. ? After these encouraging results, we are now extending this work as to take into account the dynamic of the system, where jobs can change their demands during time and possibly migrate between the hosts of the infrastructure (to enable dynamic optimal consolidation).

4.4 Software level

Since a long time, offline analysis of codes are performed in embedded systems to evaluate the energetic cost of processors within limited electric constraints. In large scale distributed systems, such offline analysis with apriori consumption models do not exist so far. Techniques like [10] start to dynamically relate a set of observed elements on the system to actual power consumption: Performance counters, load average, memory usage, etc. can be mathematically related to power consumption using linear regression techniques for instance. In these works the more metrics are observed and the more accurate the prediction of energy usage will be. Such techniques are difficult to apply on a very large scale, since a large number of elements have to be observed, possibly intrusively. Moreover, with virtualization mechanisms and communications, such regression techniques based on observations still have to be developed.

Other works include Service Level Agreement (SLA) considerations. In these works, the applications state their performance to achieve or the energy cap not to exceed. For instance, [3] uses machine learning techniques to achieve SLA specifications while [17] use autonomic computing for the same.

5 Discussion and conclusion

In this paper we propose an overview of performance evaluation under energetic concerns. We describe commonly used metrics and benchmarks, before giving the main trends for energy savings, focusing especially on large scale distributed systems.

We have seen that not a single metric has emerged and that many compete nowadays. Many are useful and complementary and the coming years will tell which ones are used in everyday practices. Finally, as mentioned in the introduction, the ecological case is not under study today. For instance the nature of energy used or the full lifecycle of IT equipment from manufacturing to recycling are most of the time ignored. We believe that the next generation metrics will encompass these ecological parameters as well as the today simple energetic costs.

6 Acknowledgement

This work was partially supported by the COST (European Cooperation in Science and Technology) framework, under Action IC0804 (www.cost804.org). The author wants to thank particularly D. Borgetto, H. Casanova and G. Da Costa for the common work on the energy yield metric and problem resolution developed in this article.

References

1. AMD-ACP. www.amd.com/us/documents/43761c_acp_wp_ee.pdf.
2. Embedded Microprocessor Benchmark Consortium Energy benchmark. www.eembc.org/benchmark/power_sl.php.
3. J. Ll Berral, I. Goiri, R. Nou, F. Julia, J. Guitart, R. Gavalda, and J. Torres. Towards energy-aware scheduling in data center using machine learning. In *ACM/IEEE International Conference on Energy-Efficient Computing and Networking (e-Energy), Passau, Germany, 13/04/2010-15/04/2010*. ACM, April 2010.
4. Paolo Bertoldi and Bogdan Atanasiu. Electricity consumption and efficiency trends in the enlarged european union, available online re.jrc.ec.europa.eu/energyefficiency/pdf/eneff report 2006.pdf, 2006.
5. Damien Borgetto, Georges Da Costa, Jean-Marc Pierson, and Amal Sayah. Energy-Aware Resource Allocation. In *Energy Efficient Grids, Clouds and Clusters Workshop (co-located with Grid 2009) (E2GC2), Banff, 13/10/09-15/10/09*. IEEE, octobre 2009.

6. Kirk W. Cameron, Kirk Pruhs, Sandy Irani, Partha Ranganathan, and David Brooks. Report of the science of power management workshop, available at scipm.cs.vt.edu/scipm-reporttonsf-web.pdf, April 2009.
7. Davide Careglio, Georges Da Costa, Ronen I. Kat, Avi Mendelson, Jean-Marc Pierson, and Yiannakis Sazeides. Hardware leverages for energy reductions in large scale distributed systems. Technical Report IRIT/RT-2010-2-FR, IRIT, University Paul Sabatier, Toulouse, May 2010.
8. Wu chun Feng and Thomas Scogland. The green500 list: Year one. In *23rd IEEE International Parallel and Distributed Processing Symposium (IPDPS) - Workshop on High-Performance, Power-Aware Computing (HP-PAC)*, Rome, Italy, May 2009.
9. Standard Performance Evaluation Corporation. www.spec.org/.
10. Georges Da Costa and Helmut Hlavacs. Methodology of measurement for energy consumption of applications. Technical Report IRIT/RT-2010-4-FR, IRIT, University Paul Sabatier, Toulouse, July 2010.
11. Transaction Processing Performance Council. www.tpc.org/.
12. Georges Da Costa, Marcos Dias De Assuncao, Jean-Patrick Gelas, Yannis Georgiou, Laurent Lefvre, Anne-Ccile Orgerie, Jean-Marc Pierson, Olivier Richard, and Amal Sayah. Multi-Facet Approach to Reduce Energy Consumption in Clouds and Grids: The GREEN-NET Framework. In *ACM/IEEE International Conference on Energy-Efficient Computing and Networking (e-Energy), Passau, Germany, 13/04/2010-15/04/2010*, pages 95–104. ACM, 2010.
13. Jack Dongarra, Piotr Luszczek, and Antoine Petitet. The linpack benchmark: past, present and future. *Concurrency and Computation: Practice and Experience*, 15(9):803–820, 2003.
14. Jack J. Dongarra, Hans W. Meuer, Erich Strohmaier, Jack J. Dongarra, Hans W. Meuer, and Erich Strohmaier. Top500 supercomputer sites. Technical report, Supercomputer, 1997.
15. Transaction Processing Performance Council Energy. http://www.tpc.org/tpc_energy/.
16. Xiaobo Fan, Wolf-Dietrich Weber, and Luiz Andre 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.
17. Aeiman Gadafi, Daniel Hagimont, Laurent Broto, and Jean-Marc Pierson. Autonomic Energy Management of Clustered Applications. In *E2GC2 : Energy Efficient Grids, Clouds and Clusters Workshop (co-located with Grid 2009), Banff, Canada, 13/10/09-15/10/09*. IEEE, octobre 2009.
18. The Green Grid. www.thegreengrid.org/.
19. Chamara Gunaratne and Kenneth J. Christensen. Ethernet adaptive link rate: System design and performance evaluation. In *LCN*, pages 28–35. IEEE Computer Society, 2006.
20. Helmut Hlavacs, Georges Da Costa, and Jean-Marc Pierson. Energy consumption of residential and professional switches. *Computational Science and Engineering, IEEE International Conference on*, 1:240–246, 2009.
21. Pierson Jean-Marc and Helmut Hlavacs, editors. *Proceedings of the COST Action IC804 on Energy Efficiency in Large Scale Distributed Systems - 1st Year, available online on www.cost804.org*. IRIT, July 2010.
22. Randy Katz and David Hutchison, editors. *ACM/IEEE International Conference on Energy-Efficient Computing and Networking (e-Energy), Passau, Germany, 13/04/2010-15/04/2010*. ACM, April 2010.
23. Kevin Lim, Parthasarathy Ranganathan, Jichuan Chang, Chandrakant Patel, Trevor Mudge, and Steven Reinhardt. Understanding and designing new server architectures for emerging warehouse-computing environments. *SIGARCH Comput. Archit. News*, 36(3):315–326, 2008.

24. Piotr Luszczek, David H. Bailey, Jack Dongarra, Jeremy Kepner, Robert F. Lucas, Rolf Rabenseifner, and Daisuke Takahashi. S12 - the hpc challenge (hpcc) benchmark suite. In *SC*, page 213. ACM Press, 2006.
25. Hans Werner Meuer. The top500 project: Looking back over 15 years of supercomputing experience. *Informatik Spektrum*, 31(3):203–222, 2008.
26. Gordon E. Moore. Cramming more components onto integrated circuits. *Electronics*, 38(8), apr 1965.
27. Gordon E. Moore. Progress in digital integrated electronics. *IEEE, Technical Digest 1975. International Electron Devices Meeting*, 1975.
28. Bruce Naegel. Energy efficiency: The new sla, available online: datacenterjournal.com/index.php?option=com_content&task=view&id=2352&itemid=43. *The Data Center Journal*, December 2008.
29. COST Action IC0804 on Energy Efficiency in Large Scale Distributed Systems. www.cost804.org.
30. Jean-Marc Pierson. Allocating resources greenly: Reducing energy consumption or reducing ecological impact? In *ACM/IEEE International Conference on Energy-Efficient Computing and Networking (e-Energy), Passau, Germany, 13/04/2010-15/04/2010*, pages 127–130. ACM, avril 2010.
31. M. Poess, R. O. Nambiar, K. Vaid, J. M. Stephens, K. Huppler, and E. Haines. Energy Benchmarks: A Detailed Analysis. In *ACM/IEEE International Conference on Energy-Efficient Computing and Networking (e-Energy), Passau, Germany, 13/04/2010-15/04/2010*, pages 131–140. ACM, 2010.
32. Standard Performance Evaluation Corporation Power and Performance. www.spec.org/power_ssj2008/.
33. U.S. Environmental Protection Agency ENERGY STAR Program. Report to congress on server and data center energy efficiency, available online at www.energystar.gov, August 2007.
34. Andy Rawson, John Pfleuger, and Tahir Cader. Green grid data center power efficiency metrics: Pue and dcie. In *The Green Grid*, December 2008.
35. M. Stillwell, D. Schanzenbach, F. Vivien, and H. Casanova. Resource Allocation using Virtual Clusters. In *Proc. of the 9th IEEE Symp. on Cluster Computing and the Grid*, May 2009.