

# Leveraging Heterogeneity for Energy Minimization in Data Centers

Marina Zapater<sup>\*</sup> José L. Ayala José M. Moya

**Abstract**—Energy consumption in data centers is nowadays a critical objective because of its dramatic environmental and economic impact. Over the last years, several approaches have been proposed to tackle the energy/cost optimization problem, but most of them have failed on providing an analytical model to target both the static and dynamic optimization domains for complex heterogeneous data centers. This paper proposes and solves an optimization problem for the energy-driven configuration of a heterogeneous data center. It also advances in the proposition of a new mechanism for task allocation and distribution of workload. The combination of both approaches outperforms previous published results in the field of energy minimization in heterogeneous data centers and scopes a promising area of research.

**Index Terms**—heterogeneous data centers, energy-aware optimization, green computing

## I. INTRODUCTION

Current data centers provide the required infrastructure for the execution of a wide range of applications and services including social and business networking, Webmail, Web search, electronic banking, Internet marketing, distributed storage, High Performance Computing (HPC), etc. Large data centers are composed of tens of thousands of servers with tens of peta bytes of storage, and multiple hundreds of giga bit bandwidth to the Internet. They typically serve millions of users globally and 24-7.

A drawback to this capacity growth has been a rapid and dramatic increase of the energy consumption and power density of data centers. The electric bill of the data centers (including the electricity needed for cooling and air conditioning in the data center) was projected to pass 7 billion US dollars in the US alone, while the power density reached  $60 \text{ KW}/m^2$  for data centers by 2010. The Environmental Protection Agency (EPA), in its August 2007 report to the US Congress, affirmed that data centers consumed about 61 billion kilowatt-hours (kWh) in 2006, roughly 1.5 percent of total U.S. electricity consumption, for a total electricity cost of about \$4.5 billion [1].

The work proposed in this paper advances in the technology of energy-efficient data centers and the mechanisms to place them on a more scalable and sustainable energy-efficiency curve.

There are a number of different techniques to reduce the energy cost and power density in data centers at different levels of granularity: chip-level, server level, rack level, data center level, etc. Over the last years, several authors have addressed this problem by the well-known technique of Dynamic Voltage and Frequency Scaling (DVFS) [2] or by the introduction of heuristics to minimize power [3]. The former can effectively reduce the dynamic power of the system, while the latter minimizes the total power of the data center. Both, however, fail on exploiting the heterogeneity of the data center and they cannot minimize the system leakage power with an appropriate assignment of workloads.

On the other hand, recent works like [4] do consider heterogeneous data centers and formulate a detailed mathematical model to perform an efficient energy-aware task scheduling. However, the authors do not propose a methodology to solve this optimization problem in an efficient way to drive the selection of processors in a heterogeneous data center under different workloads.

The work proposed in this paper addresses the existing issues in the field of energy-aware task assignment for green data centers by targeting the following goals:

- The usage of heterogeneity to minimize the energy consumption in specific-purpose data centers, by using a mixed static/dynamic approach to the problem and by characterizing real periodic workloads in terms of energy.
- The static optimization aims to find the best configuration of the data center given a set of heterogeneous machines. We will prove that the best combination is a heterogeneous data center.
- The dynamic optimization shows that the energy can be reduced significantly by optimizing the task allocation and distribution algorithm of the resource manager. We will prove that the best results are obtained in heterogeneous data centers like the ones proposed in our configurations.

This paper is organized as follows: Section II gives further information on the motivation and the related work on this topic. Section III gives an overview of the proposed solution. The energy characterization is presented in Section IV, while

Section V details the algorithm used for the static/dynamic optimization. Results and evaluation are shown in Section VI. Finally, the conclusions of the paper are drawn in Section VII.

## II. RELATED WORK

During the last years, several approaches have targeted the problem of energy efficiency in data centers, proposing different techniques to optimize the energy-aware metrics. In [5], the authors have tried to identify workload time series to dynamically regulate CPU power and frequency to optimize power consumption. Other works that apply voltage scaling, like [6] and [7], also manage the concepts of monitoring and estimation of the workload and the idea of having a pool of generic and interchangeable resources. These works have proposed interesting approaches for the run-time management of the workload in homogeneous data centers; however, our research presents how the heterogeneity in the selection of the processors that compose the data center leads to higher energy savings.

Load balancing [8], and particularly dynamic resource provisioning [9], can be used to distribute the total workload of the data center among different servers in order to balance the per-server workload. As opposed to our work, the previous run-time techniques do not consider an off-line analysis for the energy-efficient design of the data center.

Virtualization technology has provided a promising way to manage application performance by dynamically reallocating resources to VMs. Several management algorithms have been proposed to control the application performance for virtualized servers [10] and to solve the VM-server mapping problem for power savings [11].

The closest works to ours are [4] and [12]. While the first one is focused on cloud servers and the second one applies control theory, both of them lack a detailed and accurate energy model that supports the proposed optimizations. Moreover, as opposed to us, they tend to consider data centers with a low workload or the absence of any initial workload before the management takes place.

In this paper, we present a mixed static/dynamic approach for the energy-efficient management and configuration of a high-performance data center. While previous techniques have not considered the impact of the proper selection of processors in the design of the data center, our research work will firstly propose an heterogeneous design of the system based on an accurate energy model. After that, our optimization problem solved in run-time, will also exploit this heterogeneity to further reduce the energy consumption of the system, outperforming previous approaches.

## III. PROPOSED SOLUTION

Our proposed approach will mainly target specific-purpose data centers, with periodic workloads that can be easily characterized. In these systems, the values of occupancy are also constantly high (i.e: data centers for financial applications, graphic rendering, etc.). Therefore, we can assume with high level of accuracy that the workload exhibited during a period

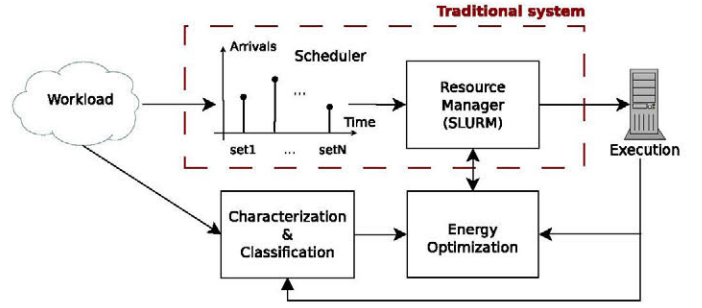


Fig. 1. Energy Optimization System

of time (i.e: one day) is representative of the workload that the data center will have in any other period of time (i.e: the next day).

The traditional functional system found in today's data centers comprises: a workload, which is a set of different tasks entering the system; a task scheduler, which queues the tasks in time, deciding their priority of execution; and a resource manager, which has the knowledge of the available system resources and decides where each task is going to be executed. The data center model that we propose is a variation of the traditional one and can be understood as the system described by Figure 1. In our case, we assume that the workload entering the system has already been scheduled by a commercial scheduler (such as Moab), and we implement our solution in the resource manager (Slurm [13] in our case), which is fed with that workload. This workload can be understood as a collection of job sets randomly distributed in time. Each job set has a random number of tasks; however, the number of different tasks is fixed for all workloads. Each task will be characterized for every machine (resources) of the system in order to obtain its profiling information (see Section IV for details). This information will be used to make an efficient optimization of the data center in two different ways: static and dynamic.

The first optimization approach will perform an off-line configuration of the data center to obtain the most suitable energy-efficient set-up. This approach will find the most appropriate combination of resources (server architectures) from a set of available ones, that will be needed to provide the required energy efficiency and still satisfy the performance constraints. We will prove that the result of the optimization is a heterogeneous data center, which performs better than any of the possible homogeneous configurations.

The second optimization will work after the first one has been applied and will perform a run-time optimization to allocate and distribute the tasks of the workload between the computing resources. This approach will prove that different allocations of tasks lead to different energy consumption values. This means that we can create run-time optimizations in terms of energy that feed information back to the resource manager and exploit the availability of heterogeneous resources.

In the next two sections we will present the application

characterization and the optimized task allocation phases.

#### IV. ENERGY PROFILING

State-of-the-art literature on this topic [3] proposes the following formula to calculate the power consumption for every server in a data center:

$$P_i = \rho \cdot f_i \cdot C_P \cdot (T_{out}^i - T_{in}^i) \quad (1)$$

where  $\rho$  is the air density and depends on ambient temperature,  $f_i$  is the airflow,  $C_P$  is the specific heat capacity of dry air at ambient pressure, and  $T_{out}$  and  $T_{in}$  are the server outlet and inlet temperature respectively.

Equation 1 is useful in the case of an experimental set-up where the utilization of the data center capacity is high. In this particular case, heat recirculation is a very important factor and the inlet temperature of the machines under study is increased due to the air room temperature and the outlet air of the other machines in the room. In our case, for the task characterization of specific-purpose data centers, we can perform the task characterization off-line in an isolated environment where recirculation is negligible.

Our target with this characterization phase is to obtain a value of the energy variation that occurs when we allocate a certain task  $t$  in a specific processor  $p$ :  $e_{tp}$ . Therefore, for a constant ambient temperature, constant values of the fan speed in the servers, and for data taken in constant time intervals, the outlet temperature follows the CPU temperature [14], and the energy calculation can be reconsidered as follows.

Let  $\rho$ ,  $f_m$  and  $C_P$  be the parameters mentioned in Equation 1, being  $f_m$  different for each server machine  $m$  and obtained from the manufacturer specification.  $\tau_{tp}^{total}$  is the total duration of task  $t$  in processor  $p$ ,  $T_{tp}^{cpuAVG}$  is the average CPU temperature and  $T_p^{cpu}(\tau = 0)$  is the idle temperature of each CPU. Taking into account the aforementioned considerations, the energy consumed per task in each processor is:

$$e_{tp} = \rho f_m C_P (T_{tp}^{cpuAVG} - T_p^{cpu}(\tau = 0)) \tau_{tp}^{total} \quad (2)$$

This approximation has proved to be realistic, as shown in the results section, since for our tests (performed with commercial machines in which real measures were acquired) the ambient temperature did not raise during the executions of the workload. Also, the fan speed remained approximately constant during characterization, which made the air flow constant<sup>1</sup>.

In order to obtain the total energy for the allocation of a task in a certain server, we only have to add the energy variation due to this process to the idle energy consumed by the machine.

<sup>1</sup>A constant fan speed is coherent for the particular case of specific-purpose data centers executing cpu-intensive applications, as it is in this case. If the workload is far from 100% most of the time, then the fan speed will no be constant nor the formula in 2 will represent a realistic scenario.

#### V. ENERGY OPTIMIZATION ALGORITHMS

Once the characterization is performed, we will use the obtained information to generate a set of optimizations that improve the energy efficiency of a data center. In this sense, we can think of three different scenarios:

- The definition of a data center which has the optimum number of machines, given a limited budget and a limited room space, to run the workload.
- The upgrade of an existing data center with new machines that are selected based on a criteria for energy-optimality.
- The execution of the workloads in an already-existing data center. This refers to applying different and new resource managing techniques that exploit the machine heterogeneity.

These three case studies can really be combined in two: (i) a static and off-line approach, which includes the creation and extension of a data center; and (ii) a dynamic run-time approach, that tackles the resource management of the tasks to be executed.

The following subsections will explain further these approaches.

##### A. Static off-line data center optimization

The static off-line approach tries to find the optimum number and combination of server machines from a set of architectures, given a certain workload. The maximum number of different architectures to be used is given by the user. Each task of the workload has a different energy profile per processor, obtained during the characterization phase.

The optimization phase is defined as follows. Let us denote by  $M$  a set of machines, by  $P$  a set of processors and by  $T$  a set of tasks that must be executed. Each machine  $m$  has a price of  $\psi_m$ , consumes power in idle state  $\pi_m$  and occupies a certain space  $\alpha_m$  (understood as the number of U's in a rack). Each processor  $p$  belongs to one machine  $m$ , denoted as  $p_m$ . Every task  $t$  has a duration and consumes a certain amount of energy depending on the target processor,  $\sigma_{tp}$  and  $e_{tp}$  respectively. The problem consists on finding a subset of  $M$  that is able to execute the required tasks  $T$  minimizing the energy consumption:

$$\text{Minimize} \left\{ \sum_{t \in T, p \in P} k_{tp} \cdot e_{tp} + \sum_{m \in M} \pi_m \cdot \tau^{max} \right\} \quad (3)$$

where  $k_{tp}$  is a binary variable that is set to 1 if the task  $t$  is executed in processor  $p$ .  $\tau^{max}$  is the time instant at which all the tasks have been executed. The constraints that the proposed model must fulfill are the following:

$$\sum_{t \in T} k_{tp_m} \cdot \sigma_{tp_m} \leq \tau_m^{max}, \quad \begin{cases} m = 1, \dots, M, \\ p_m = 1, \dots, P_m \end{cases} \quad (4)$$

$$\tau_m^{max} \leq \tau^{max}, \quad m = 1, \dots, M \quad (5)$$

$$\sum_{p \in P} k_{tp} = 1, \quad t = 1, \dots, T \quad (6)$$



$$\sum_{m \in M} k_m \cdot \psi_m \leq \Psi, \quad \sum_{m \in M} k_m \cdot \alpha_m \leq A \quad (7)$$

These constraints ensure that all the tasks are executed within a maximum time 4–5, that all the tasks will be executed once and just once in a processor 6, and that both the price of the data center and its room space will not exceed certain values 7.  $k_m$  is a binary variable that is set to 1 if the machine  $m$  is used.

As a result, we will obtain the optimum number of machines of each type needed to build our data center with energy-performance constraints.

### B. Dynamic run-time allocation

In this case, we suppose that we already have a heterogeneous data center with a fixed number  $M$  of machines (i.e.: the heterogeneous data center found with the static optimization). The dynamic run-time allocation of the tasks, performed by the resource manager, aims at minimizing the energy consumption of the assignment by placing each task where it wastes the minimum energy.

The minimization function is the same than the previous one 3. However, instead of constraints 4, 5 and 7, we use the following one to express the needs of the dynamic task allocation 8:

$$\sum_{t \in T} k_{tp_m} \cdot \sigma_{tp_m} + \gamma_{pm} \leq \tau_m^{max} \quad \begin{cases} m = 1, \dots, M \\ p_m = 1, \dots, P_m \end{cases} \quad (8)$$

This time, the problem consists on finding the most appropriate allocation of tasks  $t$  in processors  $p$ , that is, finding the optimum  $k_{tp}$  that minimizes the energy consumption.  $k_{tp_m}$ ,  $\sigma_{tp_m}$  and  $\tau_m^{max}$  have the same meaning than before. The new factor  $\gamma_{pm}$  is a time offset that represents the amount of time that a processor  $p = 1, \dots, P$  is occupied (executing previous tasks) when the new job set arrives. In this way the system can take into account the initial usage of processors.

The aforementioned constraints ensure that each task will only be allocated in one processor, try to find a compromise between energy minimization and execution time, and allow to distribute tasks in an already-occupied system by setting  $\gamma_{pm}$  to an appropriate value.

As a result, the algorithm will give the allocation of each of the tasks in the workload to the specific machine. Because the workload is divided in job sets, the algorithm will be executed each time that a new set of tasks arrives.

## VI. RESULTS

In this section we present the results obtained after exercising the task characterization and optimization techniques described in this paper.

The data used in order to characterize the workload have been acquired experimentally from the commercial servers showed in table I after the execution of the different tasks. The parameters needed to obtain the energy values have been obtained via snmp, ipmitool and the proprietary tools

Server Model	Processor	#Cores	Idle Power	Airflow ( $m^3/h$ )	Price	Size
RX-220	AMD 64 @2GHz	2	122W	118	\$2000	2U
PrimePower 450	Sparc 64V @1.1GHz	4	200W	200	\$1000	4U
RX-300 S6	Intel Xeon @2.4GHz	8	137W	130	\$3500	2U

TABLE I  
SERVERS PARAMETERS

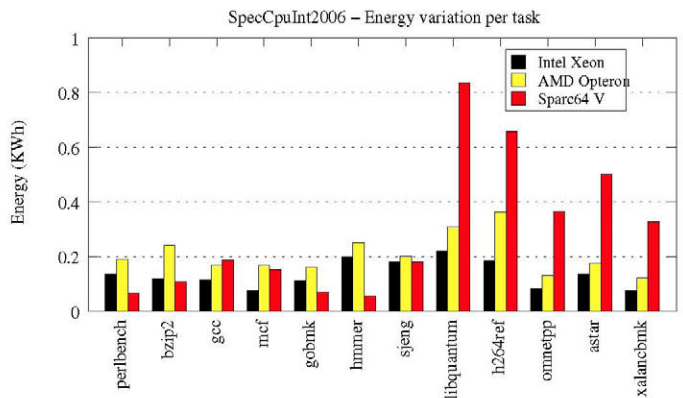


Fig. 2. Energy Characterization of the tasks

of Fujitsu. These parameters are the ambient, CPU and motherboard temperature, the fan speed and processor frequency. The only data that has not been directly characterized nor modeled with equations is the idle power consumption of the servers and the air flow, which have been obtained from the manufacturers specifications. As can be seen in the table, the selected machines present a high level of heterogeneity: they have different processor architectures, different number of cores and while the Sparc and AMD machines are quite old (year 2004-2005), the Intel machine is pretty new (year 2010) and, thus, is expected to outperform the other two.

### A. Workload characterization

The generated workload is composed of 12 different tasks from the SpecCpuInt 2006 benchmarks. We have chosen this benchmark because: (i) it is one of the most common benchmarks used for performance tests in specific-purpose data centers, (ii) it is a cpu-intensive benchmark<sup>2</sup>; and (iii) we can ignore the effects of network delays and disk transfers, and still be working with accurate models.

Finally, a random workload of 2000 tasks randomly split in different job sets of 150, 200, 250 or 300 tasks, and with random arrival times of 10, 20 or 30 minutes has been generated.

The complete benchmark has been executed for all the three servers. For these executions, both the ambient temperature and the fan speed have been approximately constant. With

<sup>2</sup>This has major advantages, as it allows to characterize the tasks taking the power dissipation of the CPU as the maximum contribution to the energy consumption (which is a real hypothesis for our case)

these data, we have used equation 2 to calculate the energy variation of each task per processor. These results are shown in figure 2. We have added this variation to the idle power (per processor) of each machine to get the total energy consumption.

At a first glance, we can see that there is margin for improvement. For example, even though the Intel server should be better than the others, we find that there are some tasks in which the Sparc server outperforms the Intel. On the other hand, the Sparc server behaves very bad with some specific tasks.

Intuitively, this experiment lets us see that with the proper usage of heterogeneity and an efficient optimization algorithm we could obtain really good results in terms of energy.

### B. Scenario 1: Optimum heterogeneous data center configuration

This experiment consists on the definition of the optimum data center. All the algorithms presented for both this scenario and the next one have been coded using ILOG CPLEX optimization suite. The reason for choosing CPLEX is that this tool does not only provide the optimization libraries to solve MILP problems, but also a very complete API for Java, C++, Python and .NET. This will be very useful in the run-time optimization step of next subsection VI-C.

As our experimental work has been performed with three machines that have completely different architectures, we split the data center configuration problem in two cases. First we try to find the optimum data center configuration with the selection of cores from the group of the two oldest machines (the Sparc and the AMD architecture); next, we look for the optimum data center configuration with the selection of cores from the group of the three machine architectures. To do so, we feed the optimizer with a pool 100 machines of each type, and a job set of 200 tasks to be allocated. In this way, we can optimize the data center configuration for an occupancy similar to the worksets of the workload. This allows to minimize the total computation time (and thus, the total idle power).

The implementation of the algorithms will run for a limited amount of time (20 minutes) to find an optimal or near-optimal solution with a low-enough error. The amount of time spent to find a good solution is not an important parameter because this is an off-line optimization that will be carried out just once, whenever the data center is going to be planned or extended.

We compare the results of the heterogeneous data center with the corresponding homogeneous data center, assuring that: (i) the total budget and space constraints are the same, or (ii) that the number of total cores (that is, the data center computing capacity) is the same. We use Slurm resource manager default configuration to simulate the workload allocation and distribution. Slurm can be run on multiple-slurmd mode to emulate a cluster of machines, setting the task affinity to core. This means that Slurm will bind each task to one and only one core, and a machine will execute as many tasks in parallel as available cores.

Configuration	Budget(\$)	Cores	Time (h)	Energy (kWh)
Heterogeneous (AMD & Sparc)	94000	214	118	2761
AMD only (budget limit)	94000	188	276	3472
Sparc only (budget limit)	94000	188	269	2913
AMD only (comp. limit)	214000	214	249	3540
Sparc only (comp. limit)	54000	214	243	2942

TABLE II  
SCENARIO 1: DATA CENTER COMPARISON

Table II shows the results for the AMD Opteron vs Sparc 64V case. As can be seen, the heterogeneous data center outperforms the homogeneous ones both when homogeneous distributions with the same budget limitation are chosen or when the same computational capacity is selected. The savings in terms of energy range from 5% to 22%. In terms of performance, the heterogeneous solution is also faster than the homogeneous one, with speed-ups in the range of 27-35%.

In the IntelXeon vs AMD Opteron vs Sparc 64C case, the optimizer chooses as the best data center a heterogeneous configuration comprised of 5 Sparc and 24 Intel machines. In this sense, the optimization system admits that there are some tasks that perform better in Sparc processors, and uses them to minimize energy. The improvement margin is, however, smaller. For the same computational capacity, the heterogeneous solution decreases energy in a 6% and execution time in a 7%.

### C. Scenario 2: Optimum workload distribution and allocation

In this experiment, we generate an optimum distribution and allocation of tasks. The input to our system (which acts as a resource manager) is the same than in the previous scenario. However, the whole workload will now be used instead of a particular job set.

This case is different from the previous one in the sense that the algorithms must be ready to work in run-time, in parallel with the execution of the workload. This can be done with the implementation of a new Slurm plug-in that uses the CPLEX library tools for the task distribution and allocation. The optimization will be run each time that a new job set (defined as a pack of random tasks) arrives to the resource manager, for a limited period of time, in order to assign tasks to processors. This optimization procedure mainly improves the total energy variation (the aforementioned  $e_{tp}$ ). The goal of this optimization is not really to reduce drastically the total execution time (as this time is inherently reduced by the static optimization), but just to ensure that it does not exceed a maximum. Also, as we suppose a stationary state of the data center with all the machines turned on and highly occupied, the idle power consumption of machines is not as important as in the static optimization.

Table III shows the results of the dynamic optimization, performed both for the homogeneous data centers and the

Configuration	Time (h)	Energy variation (kWh)
Heterogeneous AMD & Sparc	112	151
AMD only (budget limit)	248	218
Sparc only (budget limit)	261	290
Heterogeneous Intel & Sparc	90	101
Intel only (comp. limit)	100	134
Sparc only (comp. limit)	103	301

TABLE III  
SCENARIO 2: ENERGY VARIATION MINIMIZATION

heterogeneous ones and after the execution of the optimization algorithms with the whole workload. As can be seen, energy savings are obtained for all the data centers, but specially for the heterogeneous ones, that outperform the homogeneous configurations in all cases. Even though the improvements are again higher for the AMD and Sparc combination, there are also savings in the Intel and Sparc case. This savings range from 24% to 47% without execution time overhead.

Finally, it must be noted that the execution of this run-time optimization is completely feasible. As the algorithm does only have to work with the current job set, the optimization is fast: it only needs from 30s to 1min of time to find results with good error margins. As the execution time of a task ranges from 10 minutes to hours, the time overhead introduced by the optimization is negligible.

## VII. CONCLUSIONS

This paper proposes a mixed static/dynamic energy minimization strategy for data centers. The solution is especially aimed for specific-purpose data centers, which exhibit high occupancies and where the workload can be adequately characterized. The proposed static approach shows how the proper selection of the heterogeneity of the data center design can achieve a notorious energy optimization during the design phase of the system. This energy optimization can be extended to more than a 20% with low execution time overhead when combined with the proposed dynamic load assignment mechanism. Moreover, the run-time optimization can be easily implemented in the resource manager as an Slurm plug-in, which allows integration with actual commercial data centers. The conducted experimental work has tackled with realistic workloads and machine architectures, and the obtained plug-ins will be very soon incorporated in a real environment (Madrid Supercomputing and Visualization Center – CeSViMa). Future work will focus on the development of more accurate energy models that will take real measures of the servers idle power and study the dependency on the power consumption of servers with the room temperature is envisioned, as well as the effect of the cooling equipment of the data room. The objective will be to develop an integrated energy minimization solution that takes into account both the resource manager and the air conditioning control.

## ACKNOWLEDGMENT

This work was funded by the Campus of International Excellence (CEI) of Moncloa, under Research Grant of the

Program for Attracting Talent (PICATA).

The authors would like to thank the staff of Elite Tecnologías Convergentes S.L., who kindly lent us their servers to perform the workload characterization.

## REFERENCES

- [1] EPA, "EPA Report to Congress on Server and Data Center Energy Efficiency," U.S. Environmental Protection Agency, Tech. Rep., 2007. [Online]. Available: [http://www.energystar.gov/ia/partners/prod/\\_development/downloads/EPA\\_Datacenter\\_Report\\_Congress\\_Final1.pdf](http://www.energystar.gov/ia/partners/prod/_development/downloads/EPA_Datacenter_Report_Congress_Final1.pdf)
- [2] "Integrating adaptive components: An emerging challenge in performance-adaptive systems and a server farm case-study," in *Proceedings of the 28th IEEE International Real-Time Systems Symposium*, ser. RTSS '07. Washington, DC, USA: IEEE Computer Society, 2007, pp. 227–238. [Online]. Available: <http://dx.doi.org/10.1109/RTSS.2007.29>
- [3] Q. Tang, S. K. S. Gupta, and G. Varsamopoulos, "Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach," *IEEE Trans. Parallel Distrib. Syst.*, vol. 19, pp. 1458–1472, November 2008. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1441362.1441369>
- [4] L. M. Zhang, K. Li, and Y.-Q. Zhang, "Green task scheduling algorithms with speeds optimization on heterogeneous cloud servers," in *Proceedings of the 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing*, ser. GREENCOM-CPSCOM '10. Washington, DC, USA: IEEE Computer Society, 2010, pp. 76–80. [Online]. Available: <http://dx.doi.org/10.1109/GreenCom-CPSCom.2010.70>
- [5] C. Lefurgy, K. Rajamani, F. Rawson, W. Felter, M. Kistler, and T. W. Keller, "Energy management for commercial servers," *Computer*, vol. 36, pp. 39–48, December 2003. [Online]. Available: <http://dx.doi.org/10.1109/MC.2003.1250880>
- [6] E. Pinheiro, R. Bianchini, E. V. Carrera, and T. Heath, "Load balancing and unbalancing for power and performance in cluster-based systems," in *Workshop on Compilers and Operating Systems for Low Power*, 2001.
- [7] E. N. Elnozahy, M. Kistler, and R. Rajamony, "Energy-efficient server clusters," in *Proceedings of the 2nd international conference on Power-aware computer systems*, ser. PACS'02. Berlin, Heidelberg: Springer-Verlag, 2003, pp. 179–197. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1766991.1767007>
- [8] V. Cardellini, M. Colajanni, and P. S. Yu, "Dynamic load balancing on web-server systems," *IEEE Internet Computing*, vol. 3, pp. 28–39, May 1999. [Online]. Available: <http://dl.acm.org/citation.cfm?id=613336.613401>
- [9] E. Pakbaznia, M. Ghasemazar, and M. Pedram, "Temperature-aware dynamic resource provisioning in a power-optimized datacenter," in *Proceedings of the Conference on Design, Automation and Test in Europe*, ser. DATE '10. 3001 Leuven, Belgium, Belgium: European Design and Automation Association, 2010, pp. 124–129. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1870926.1870959>
- [10] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy, and G. Jiang, "Power and performance management of virtualized computing environments via lookahead control," *Cluster Computing*, vol. 12, pp. 1–15, March 2009. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1507559.1507584>
- [11] Y. Wang and X. Wang, "Power optimization with performance assurance for multi-tier applications in virtualized data centers," *Parallel Processing Workshops, International Conference on*, vol. 0, pp. 512–519, 2010.
- [12] D. Barbagallo, E. Di Nitto, D. J. Dubois, and R. Mirandola, "A bio-inspired algorithm for energy optimization in a self-organizing data center," in *Proceedings of the First international conference on Self-organizing architectures*, ser. SOAR'09. Berlin, Heidelberg: Springer-Verlag, 2010, pp. 127–151. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1880569.1880578>
- [13] M. Yoo, A. B. Jette, M. A. Grondona, "SLURM: Simple Linux Utility for Resource Management," *LECTURE NOTES IN COMPUTER SCIENCE*, 2003.
- [14] I. American Society of Heating, Refrigerating and Air-Conditioning Engineers, "2008 ASHRAE Environmental Guidelines for Datacom Equipment - Expanding the Recommend Environmental Envelope -," 2008.