Setting Energy Efficiency Goals in Data Centers: the GAMES Approach

B. Pernici, C. Cappiello, M.G. Fugini, P. Plebani, M. Vitali[1]
I. Salomie, T. Cioara, I. Anghel[2] E. Henis, R. Kat, D. Chen, G. Goldberg[3], M. Vor dem Berge, W. Christmann[4], A. Kipp, T. Jiang, J. Liu[5], M. Bertoncini, D. Arnone, and A. Rossi[6]

[1] Politecnico di Milano, Italy
{pernici,cappiell,fugini,plebani,vitali}@elet.polimi.it
[2] Technical University of Cluj-Napoca, Romania
{ioan.salomie,tudor.cioara,ionut.anghel}@cs.utcluj.ro
[3] IBM Israel Science and Technology Limited, Israel
{ealan,ronenkat,cdoron,georgeg}@il.ibm.com
[4] Christmann Informationstechnik, Germany
{micha.vordemberge,wolfgang.christmann}@christmann.info
[5] University of Stuttgart, HLRS, Germany
{kipp,jiang,liu}@hlrs.de
[6] Engineering Ingegneria Informatica, Italy
{massimo.bertoncini,diego.arnone,alessandro.rossi}@eng.it

Abstract. Energy-aware service centers take into account energy consumption of infrastructures, machines, applications, storage systems, and their distributed computing architecture. The approach to energy efficiency in data centers in the GAMES (Green Active Management of Energy in IT Service centers) project is presented: Green Performance Indicators (GPIs), i.e., properties that, continuously monitored, evidence the level of consumed energy by the center's IT resources, can be the basis of a systematic approach to increase energy efficiency. The GPIs are the basis for improving energy efficiency with adaptive actions and to achieve a higher level of green maturity, as prescribed, for instance, in the GreenGrid Data Center Maturity Model (DCMM), based on a usage-centric perspective in GPIs. The paper briefly describes monitoring of GPIs and the adaptation actions adopted to reach the green goals. Preliminary experimental results are discussed.

Keywords: data centers, energy-efficiency, adaptivity, Green Performance Indicators, monitoring, control

1 Introduction

Service centers aware of the energy they consume in terms of resources both at the software and hardware level are gaining more and more attention with the promise of developing systems that can be tuned to consume less energy for IT resources, cooling, consumables, and lower CO_2 emissions. Researches focusing on metrics and measurements for green IT and service centers are in progress with the vision of achieving economical, environmental, and technological sustainability [1, 2]. To this aim, several sets of metrics have been proposed to measure service centers efficiency, e.g., as proposed by Green Grid, Uptime Institute, Transaction Performance processing Council (TPC), to mention a few. Several techniques for an efficient resource allocation and utilization in data centers based on adaptivity have been proposed, using for instance CPU frequency as an important parameter to improve energy [3][4] or proposing an efficient usage of the storage devices [5].

To sustain this attention, GAMES is centered on designing, developing, and executing applications along the perspectives of energy awareness [6], namely of services characterized by indicators regarding which IT resources, as well as what effort (e.g., in terms of costs) are required during development, execution, and maintenance [7].

The goal of this paper is to illustrate how *Green Performance Indicators* (GPIs) can be used as a comprehensive approach to monitor service centers on energy-efficiency related parameters. The GPIs are mainly based on a usage-centric perspective, to enable the assessment of the level of use of all resources in the service centers, also in relation to the services being provided by the center and their requirements and characteristics. The GAMES approach provides a systematic and scalable approach to GPI definition, monitoring, and management. On the basis of GPIs, several control actions can be performed in order to guarantee a high level of usage of resources. Given an application and its IT resources configuration (virtualized or real), a good level of utilization of resources can be achieved through monitoring and adaptation of the system configuration to avoid waste of resources. In the paper, we discuss the GAMES achievements with respect to some of the goals set by The GreenGrid[7] [8], which defines green maturity levels for data centers.

In the following of the paper, Section 2 illustrates the main GPIs considered in this paper and their relation to the GreenGrid Maturity Model goals. Section 3 describes the monitoring infrastructure to collect and memorize sensor and system data and to analyze them. In Section 4, the controllers that, based on GPIs, select adaptivity actions for improving energy efficiency are illustrated. Section 5 presents the experimental settings and, finally, Section 6 discusses how monitoring data can be used to improve the system configurations.

2 Green Performance Indicators

Starting from the whole set of GAMES GPIs (which include technical, as well as organizational, environmental, and development-related indicators), in this paper we consider only the *IT-related* and the *QoS-related* GPIs, which have

[7] `http://www.thegreengrid.org/`

Table 1. Games Contribution to GreenGrid Maturity Model

Parameter	**Best Practice**	**Level 5**	**GPIs**	**Adapt. Actions**
Compute: Utilization	CPU 20%	CPU 60%	CPU Usage	Resource Allocation
Compute: Workload	Rationalize Workload (virtualization/consolidation)	Shift all of the workload across many data centers taking into account business priorities, external drivers, availability of resource and TCO - "Follow the Moon" strategy	CPU Usage, Memory Usage, Application performance, IOPS, Energy	Resource Allocation, reconfiguration
Compute: Operations	Understand performance through the use of standard benchmarks	Improve application use of processor, memory and major power consumption components	Storage Usage	Resource Allocation
Compute: Power Management	Power Monitoring	Optimization of power with no impact over Performance	Application Performance, Energy	CPU frequency scaling, Power mgmt at server level
Storage: Operation	Storage Consolidation	Operational media choice based on TCO model. energy usage, embedded carbon footprint and business need	Storage Usage, Energy	Storage Migration
Storage: Technology	Low Power Consuming Technology	Use/enablement of low power states for storage	Storage Usage, IOPS, Energy	Acoustic disk modes

been used in the experimental setting[8]. The interested reader is referred to [9] for a complete overview on the GAMES GPIs.

In Table 1, a subset of parameters of the GreenGrid maturity model (related to IT factors) is mapped to GAMES general GPIs and a general reference to GAMES adaptation actions is given.

For instance, considering the Compute Utilization parameter, the GreenGrid Maturity Model indicates CPU usage as a parameter, set at level 20% for Level 2 (best practices) and should reach the target goal of a value greater than 60% (Level 5) by 2015. A similar approach is taken considering other parameters of the maturity model, defining appropriate GPIs, illustrated in general terms in Table 1, and refined in Table 2.

In the first column of Table 2, the main indicators used in GAMES experimentation are reported, while in the second column details of measured values are given. All indicators can be evaluated at different abstraction levels: from system level, to a cluster of servers, to application level. *CPU Usage* relates to the processor utilization percentage. *Memory Usage* refers to the usage of main memory (RAM). This indicator characterizes the percentage of occupation of RAM. *Storage Usage* denotes to which extent storage is used. This GPI denotes the storage utilization percentages for running applications in a given

[8] The terms GPI/indicator and application/service are used in an interleaved manner.

configuration counting files and/or device (depending on the desired granularity for the measures) accesses. Throughput and capacity usage based energy efficiency indicators are also calculated by modeling used power per device. *IOPS* characterizes the efficiency of the communication with the storage/devices accounting the number of IPOS operations that can be executed in a time unit using a single Watt for a device or for storage. *Application Performance* measures energy consumption per computed unit, e.g., number of Transactions/kWh or GFLOPS/kWh. This indicator is given in $ComputationUnit/KWh$. *Response Time (RT)* is a performance GPI, related to QoS, measuring the time taken by a service S to handle user requests (measuring the delay between the moment a service request is sent and the moment the service is rendered). It can be measured also at the infrastructure level, e.g., on storage access as disk response time.

Table 2. Main GAMES GPIs

GPI	**Measured values**
CPU Usage	measured directly (%)
Memory Usage	measured directly (%)
Storage Usage	disk/file access counters, throughput and capacity usage, disk-used-capacity/W
IOPS	IOPS/W, Disk IOP rate, Device IOPS, Disk-throughput/W
Application performance	Transactions/Energy, GFLOPs/KWh
Energy	measured directly
Response Time (RT)	execution time at server (ms), disk response time

3 Monitoring IT Components in Data centers

To be able to compute the GPIs, testbed has been set up focusing on a wide variety of different test options and fine-grained monitoring. The detailed architecture of the GAMES infrastructure is described in [10] and it outside the scope of this paper. In Section 3.1, we provide a general description of the monitoring infrastructure, while in Section 3.2 we describe the system to analyze and store information deriving from monitoring and to make it accessible to the controllers which enforce the adaptation actions.

3.1 Monitoring Infrastructure

For monitoring, the testbed has been set up considering the following main elements:

- Infrastructure Servers, providing all necessary services for a network;

- a gigabit Ethernet switch;
- a Boot Server, enabling to easily switch the service systems execution on the various computing nodes;
- a Storage Server, equipped with different types of hard drives;
- a Power Distribution Unit, with an integrated power meter measuring the power consumption of the storage, switch and infrastructure servers;
- a RECS (Resource Efficient Computing System) Cluster Server, with 18 integrated computing nodes and a dedicated, system-independent monitoring architecture.

The Storage Server and the RECS Cluster Server[9] are the main testbed components. The latter offers computing capabilities of 18 CPUs that can be centrally managed. To avoid influencing the results by bothering the nodes to get sensor data, a new monitoring approach has been designed and a Cluster Server has been built. The core concepts are: i) reduce network load; ii) avoid the dependency of polling each node at the operating system layer; iii) build up a basis for new monitoring and controlling concepts. These requirements have been implemented via a dedicated master-slave architecture of microprocessors that collect the most important sensor values from each single compute node, such as status (on/off), temperature of the main board and power consumption. Every node is connected to a slave micro-controller locally gathering these metrics. Then, the master micro-controller centrally collects all information. This master micro-controller is accessible to the user by a dedicated network port and has to be read out only once to get all information about the installed computing nodes. If a user monitors e.g., 10 metrics on all 18 nodes, he would have to perform 180 pulls. These can now be reduced to one, since the master does a pre-aggregation and processing of the monitoring data. This example shows the immense capabilities of a dedicated monitoring architecture.

For storage usage, file/block level statistics are collected via special purpose monitoring agents to relate them to applications.

To collect all low-level, or directly measurable, metrics, the Nagios monitoring tool is employed. Nagios allows using a wide variety of available plug-ins to gather most typical metrics from servers like memory usage or CPU usage. In addition, some own plug-ins have been developed to read out the RECS Cluster Server micro-controller and the power distribution unit with the integrated power meter. Historically grown, many of the public Nagios plug-ins have different ways of storing their information to the database. To get a common data set, to aggregate and normalize these metrics, a pre-processing is performed (see [11]).

3.2 Providing Information to Controllers

One key aspect is how information flows from the monitoring infrastructure to the controllers which enforce adaptivity actions. The *Energy Sensing and Mon-*

[9] A detailed description of the RECS is at `http://shared.christmann.info/download/project-recs.pdf`

itoring Infrastructure subsystem is in charge of collecting, processing and dispatching data from the service center IT infrastructure and environmental sensors. Its *Infrastructure Access Module* component captures energy, performance and context data at different layers: facilities, single server (compute node), cluster server, storage system and virtual systems (virtual machines, virtual storage and application container).

Once collected, raw monitoring data are analyzed. The *Finalised Context Interface Monitor* component parses, refines and organizes raw data into structured run-time energy/performance data (context data) for the rest of the platform. Context data collected and processed in real time refer to: IT infrastructure energy/performance data, environmental data and the state of the GAMES-enabled applications running on the service center. The GPIs are evaluated starting from context data and the results are collected and stored into the *Energy Practice Knowledge Base* which eventually contains all information about configuration and status of the service center.

Besides the main service-oriented architecture of the communication system, an event-based architecture is provided to support asynchronous notifications by means of a *Provenance Tracking Interface* module. Various classes of events are specified ranging from classes regarding the start-end of executions to actions required to adapt the system.

4 Control Actions

A set of controllers are set in place to analyze the GPIs and energy consumption on the various devices and to enable control actions able to change the state/configuration of components to save energy. The controllers are described in the following sections. Table 3 summarizes the features of the Energy Aware Self-* Controller and of the Storage Controllers.

4.1 GAMES Energy Controllers

The Energy Aware Self-* Controllers combine techniques like context- and energy-aware computing, autonomic computing and self-adapting control. Run-time energy efficiency is addressed by two types of control loops: (i) a set of Local Control Loops which take adaptation decisions at each server level and (ii) a Global Control Loop which takes adaptation decisions at the whole service center level.

The *Local Control Loops* exploit the energy saving opportunities given by short-term fluctuations in the performance request levels of the servers' running tasks. The adaptation actions at each server level are: (i) put server CPU to Sleep State (C-3) when no workload is executed and (ii) execute transitions between CPU performance states (P-states) according to the workload characteristics when the CPU is in the Operating State (C-0). When deciding to change the P-State of the CPU, the Local Control Loop computes the trade-off between performance and energy by considering that the higher the P-State,

Table 3. GAMES Energy Aware Self-* Controller

Energy Aware Self-* Controller	**Method**	**Input GPIs**	**Adaptation Actions**
Local Loop Control (LLC)	human immune system inspired	CPU usage, Application Performance	Dynamic Frequency Scaling action
Global Loop Control (GLC)	Self-adaptation control, reasoning-based evaluation, reinforcement learning	CPU usage, Memory usage, Application performance, RT	Dynamic Power Management at server level, Workload consolidation, resource provisioning
Storage Management Controller	**Method**	**Input GPIs**	**Actions**
Disk Acoustic mode Control Loop	Fuzzy Inference System (FIS)	Disk IOP rate Sequential access ratio, Disk response time	Set disk acoustic mode (Normal/Quiet)
Storage placement Control Loop	File splitting into chunks, usage centric energy efficiency ranking of storage devices	Device IOPS, throughput and capacity usage	Split application file into chunks, Initial chunk placement based on device rank, Chunk migration based on device rank and chunk usage statistics

the more the power consumed by the CPU. In a lower P-State, the CPU performance degradation will increase. For the Local Control Loop, an adaptation decision methodology is defined inspired from the human immune system biological structures and processes defending a living organism against diseases [12]. The server is monitored to gather CPU-related power/performance data which is then formally represented using an antigen inspired model.

The *Global Control Loop* minimizes the service center energy consumption by pro-actively detecting the service center over-provisioned hardware computing resources and dynamically adapting, at run time, the allocated computing resources to the workload intensity changes. A self-adaptation function associates to each energy-inefficient state of the service center (i.e., a non-green state where the GPIs levels defined at design-time are not reached) an adaptation action plan [13]. To identify energy-inefficient states, an Energy Aware Context Model and ontology are defined to evaluate the GPIs through reasoning based techniques. The adaptation methodology is based on reinforcement learning and gives the following types of adaptation actions: (i) energy-aware resource provisioning and consolidation in the middleware (e.g. provisioning of hardware resources to virtual tasks, or virtual tasks deployment or migration) and (ii) dynamic power management actions at the hardware level (such as hibernate/wake-up servers).

4.2 Storage Controllers

The GAMES storage management system achieves storage energy efficiency by using two controllers: (i) the Disk Mode Controller, controlling the disk acoustic mode, and (ii) the Chunk Placement Controller, controlling the placement of

application files split into chunks.
The *Disk Acustic Mode Controller* is based on a Fuzzy Inference System that leverages our previous work [14] and translates our understanding of disk power consumption under various load conditions into a set of fuzzy rules that attempt to save energy.
The *Chunk Placement Controller* splits an application file (which can be large, e.g., 200 GB) into smaller chunks (of 1 MB or smaller, for finer grained file placement), and performs two main operations: placement of a new chunk, and chunk migration. Both placement operations aim to place the chunk on a disk such that the energy it consumes will be minimized. Energy minimization is achieved by ranking the storage devices per their usage centric energy efficiency metrics. The adopted metrics for storage are disk used capacity per Watt, disk access IOPS per Watt, and disk throughput per Watt [15].

The inputs to the storage management system include application-level information (e.g., application annotations and application to file mapping) as well as file-level and block-level monitored storage data.

5 Tools Environment and First Evaluation of Results

Monitoring is implemented using Nagios and NDOUtils. The Energy Sensing and Monitoring Infrastructure Prototype is coded in Java, EJB v3.0. It is implemented as a SOA (Service Oriented Architecture) and uses Grizzly (HTTP Web Server) and Jersey (REST Service stack). The application server for the enterprise application is Glassfish. GPIs computation uses Octave, an open-source Matlab-like engine. The event system is based on the JMS middleware. The Global and Local Control Loops are implemented in Java. The former is based on an agent-oriented architecture using JADE. For monitoring and action enforcement, the Global Control Loop uses OpenNebula Java OCA API with wakeonlan and OpenSSH tools. To develop the Local Control Loop in-line power management techniques, the cpuinfo tool and Hyperic Sigar library are used for monitoring, while the cpufreq module is used to enforce the CPU P-state changes. The EACM ontology is described in OWL while the low-level GPIs are described in the SWRL language.
As a first evaluation, the simulation of a typical HPC application has been executed on a 18 nodes cluster with enhanced monitoring capabilities, each holding an Intel P8400 CPU (2x 2.26 GHz, 1066 MHz FSB) and 4GB DDR3 Dual Channel RAM. Details of the testbed environment are reflected in [10]. Tests to analyze improvements of energy efficiency in business applications have been performed using the TPC-C benchmark.
For energy control loops, the energy consumption of a testbed server has been measured at different load values by means of a power meter. The optimal energy consumption/performance tradeoff is achieved for a threshold of about 75% CPU and Memory usage, considering a standard deviation of +/− 15%.

Considering the compliance with the Maturity Model goals related to CPU usage values, as a first achievement, we have a better performance both when

addressing the problem in the energy control loops (around 75% +/- 15% in Global Control Loop control) and in VM configuration (threshold set at 80%), while the GreenGrid Maturity Model has a goal of 60% for level 5.
Energy consumption lines up to improvements of 5% in local control loops, 15% in global control loops, and 7% in software configuration adaptation.

6 Mining and Configuration Improvement

The data collected in the monitoring are further used to assess the system and its configuration. A mining system supports statistical investigation over GPIs, while software and hardware configuration improvements are considered for system evolution and redesign.

In Table 4, indicators used to drive the selection of hardware and software configurations and setting of thresholds are shown.

Table 4. GPIs for improvement of software and hardware configurations

	GPIs
Sw config (VM)	resource usage, RT
Hw config	resource usage, IOPS/W, application performance

6.1 Mining GPIs

GPIs are constantly monitored and their historical values are forwarded to the GAMES Data Mining module. All the measures taken on the GAMES testbed at a specific instant (i.e., in a defined time slot) constitute a finalized context instance stored into a stack containing historical values. This stack is the data source; however, mining the stack for time, resource and consequently energy consumption analysis, can require quite an effort, due to the high number of monitored parameters. A pre-processing step is performed and the computation of proper and significant indicators reduce the computational cost of the mining step, by aggregating more parameters. This also permits to interprete the extracted patterns, since indicators and relations among them can be easily analyzed.

Basic descriptive statistics computations are performed, such as correlation, and supervised and unsupervised data mining algorithms are applied such as association rule mining and clustering, respectively.

Correlation between two indicators A and B is a bi-variate descriptive statistics parameter that quantifies how the behavior of A affects B. This computation is performed by the Data Mining module my means of a synchronous invocation of its API. On the contrary, the APRIORI algorithm chosen for extracting *association rules* among indicators, is asynchronously invoked by a human supervisor through the Graphical User Interface of the GAMES Energy Efficiency Tool. Through the GUI, experts can perform a rule validation process by discarding spurious rules and keeping only useful and exploitable relations. Finally, *clustering* is applied to indicators related to the application layer. Discovered clusters, for instance, could split the overall space of the business processes in more subspaces each one characterized by a well defined green performance indicators related behavior, corresponding to the cluster centroid.

6.2 Software Configuration Improvement

Using Virtual Machines enables a dynamic allocation of resources to each deployed application. The design of the configuration of Virtual Machines for business transaction running the TPC-C benchmark has been analyzed. An example is shown in Table 5 where, starting from an inefficient configuration, the system continuously detects the violation of a GPI. The violated indicators are highlighted in bold. The violation of the threshold suggests the administrator to change the configuration until reaching efficiency without violating other indicators. The reconfiguration can also be automatic if rules are defined to enact a reconfiguration when a violation occurs.

Table 5. Virtual Machine reconfiguration towards energy efficiency

Step	**Virtual Machine Configuration**	**GPI1 (App. Perf.) > 1000**	**GPI2 (CPU us.) > 80%**	**GPI3 (Storage us.) > 30%**	**GPI4 (IOPS/ Watt) > 100**	**GPI5 (Memory us.) > 75%**	**QoS1 (Resp. time) < 1 sec.**
Step0	4 CPUs, 512 MB RAM, 10 GB HD	13822	**47.15%**	38%	133	97.96%	0.034
Step1	3 CPUs, 512 MB RAM, 10 GB HD	15080	**67.42%**	32%	175.57	93.30%	0.063
Step2	2 CPUs, 512 MB RAM, 10 GB HD	16250	99.67%	38%	107.42	97.91%	0.048

6.3 Hardware Configuration Improvement

Hardware configuration design and improvement have also been analyzed. In the experimentation on the High Performance Computing (HPC) case, the Local Control Loops tackle high energy consumption by adjusting the servers' CPU power states (P-states) according to the incoming workload variation. The HPC first trials show the Local Control Loops energy saving potential. However, the results are directly influenced by the number of P-states of the servers' CPUs. A higher number of power states will provide the Local Control Loops with more accurate control on the CPU power consumption and consequently with higher energy saving. The current HPC testbed processors Intel P8400, used for the first trials have 3 P-states (for next trials the HPC test bed servers processors will be improved to Intel i7 processors with up to 12 P-states, thus achieving better results). Another optimization for improving energy savings results is to make the Local Control Loops aware of the workload dependencies and predictability in the HPC case. In this approach, when deciding to change the processor's P-states, the Local Control Loops will take into consideration the state of the neighbor dependent processors. For example, if a server's processor enters the Idle working mode, another server processor's Local Control Loop aware of this fact (due to the workload dependency knowledge among processors) may also decide to enter the Idle mode in the near future, thus starting early the necessary preparations.

The dynamic frequency scaling decisions taken by the bio-inspired Local Control Loop are taken by learning from previous experience in a similar manner to the human immune system. The energy savings results will be improved when a workload with similar characteristics is frequently encountered in the execution history of the local loop, because the dynamic frequency scaling assignment decisions for that type of workload will have already been learned. This exactly suits the HPC case, which deals with a relatively small number of applications. A detailed analysis about the first GAMES Trial results is available at http://www.green-datacenters.eu/.

7 Concluding Remarks and Future Work

In this paper, we have presented the GAMES approach to monitoring and adapting the use of resources by service centers to save energy. We set the basis to evaluate to what extent and how the maturity model goals can be achieved in GAMES through setting *thresholds* for energy-efficiency goals, based on usage-based GPIs. Monitoring and control of indicators then suggest adaptation actions able to tune energy consumption to the needed level. By analyzing the relevant context information, the results show also that the GAMES platform can improve energy efficiency by adopting different hardware and software configurations.

Ongoing work is aimed to show the effectiveness of the designed GPIs and at evaluating the overall adaptation strategies. The initial results encourage further investigation in the direction of GPIs for energy-awareness and of monitoring and mining techniques. Further integration of modules and complete integration are being developed to enrich the monitoring infrastructure and the relevant context data analysis and mining.

Acknowledgments. This work has been partially supported by the GAMES project (http://www.green-datacenters.eu/) and has been partly funded by the European Commissions IST activity of the 7th Framework Program (contract number ICT-248514). This work expresses the opinions of the authors and not necessarily those of the European Commission. The European Commission is not liable for any use that may be made of the information contained in this work.

References

1. Nowak, A., Leymann, F., Schumm, D., Wetzstein, B.: An architecture and methodology for a four-phased approach to green business process reengineering. In: ICT-GLOW. (2011) 150–164
2. Berl, A., Gelenbe, E., Girolamo, M.D., Giuliani, G., de Meer, H., Quan, D.M., Pentikousis, K.: Energy-efficient cloud computing. Comput. J. **53**(7) (2010) 1045–1051
3. von Laszewski, G., Wang, L., Younge, A.J., He, X.: Power-aware scheduling of virtual machines in dvfs-enabled clusters. In: Cluster Computing and Workshops, 2009. CLUSTER'09. IEEE International Conference on. (2009) 1–10
4. Wang, Y., Wang, X., Chen, M., Zhu, X.: Power-efficient response time guarantees for virtualized enterprise servers. In: Real-Time Systems Symposium, 2008. (2008) 303–312
5. Zhu, Q., Chen, Z., Tan, L., Zhou, Y., Keeton, K., Wilkes, J.: Hibernator: helping disk arrays sleep through the winter. In: ACM SIGOPS Operating Systems Review. Volume 39. (2005) 177–190
6. Bertoncini, M., Pernici, B., Salomie, I., Wesner, S.: Games: Green active management of energy in it service centres. In: CAiSE Forum. (2010) 238–252
7. Kipp, A., Jiang, T., Liu, J., Fugini, M.G., Vitali, M., Pernici, B., Salomie, I.: Applying green metrics to optimise the energy-consumption footprint of it service centres. International Journal of Space-Based and Situated Computing, Inderscience Publishers, to appear (2012)
8. Singh, H., et al.: Data center maturity model. Technical report (2011)
9. Kipp, A., Jiang, T., Fugini, M., Salomie, I.: Layered green performance indicators. Future Generation Computer Systems **28**(2) (may 2012) 478–489
10. Kipp, A., Jiang, T., Liu, J., Buchholz, J., Schubert, L., vor dem Berge, M., Christmann, W.: Testbed architecture for generic, energy-aware evaluations and optimizations. In: Proc. of the First International Conference on Advanced Communications and Computation (INFOCOMP 2011). (2011) 103–108
11. Kipp, A., Liu, J., Jiang, T., Khabi, D., Kovalenko, Y., Schubert, L., vor dem Berge, M., Christmann, W.: Approach towards an energy-aware and energy-efficient high performance computing environment. In: Intelligent Computer Communication and Processing (ICCP), 2011 IEEE International Conference on. (aug. 2011) 493 –499
12. Cioara, T., Pop, C.B., Anghel, I., Salomie, I., Dinsoreanu, M., Condor, I., Mihaly, F.: Immune-inspired technique for optimizing servers energy consumption. In: Proc. of the 2010 IEEE International Conference on Intelligent Computer Communication and Processing (ICCP2010), Cluj-Napoca, Romania. (2010) 273 280
13. Cioara, T., Anghel, I., Salomie, I., Copil, G., Moldovan, D., Pernici, B.: A context aware self-adapting algorithm for managing the energy efficiency of it service centres. Ubiquitous Computing And Communication Journal, Special Issue of 9th RoEduNet International Conference (2011)
14. Chen, D., Goldberg, G., Kahn, R., Kat, R., Meth, K.: Leveraging disk drive acoustic modes for power management. In: Proc. of the IEEE 26th Symposium on Mass Storage Systems and Technologies MSST 2010. (2010) 1–9
15. Chen, D., Henis, E., Kat, R.I., Sotnikov, D., Cappiello, C., Ferreira, A.M., Pernici, B., Vitali, M., Jiang, T., Liu, J., Kipp, A.: Usage centric green performance indicators. In: Proc. of GreenMetrics 2011 (SIGMETRICS workshop). (June 2011)