

Applying Green Metrics to optimise the energy-consumption footprint of IT service centres

Alexander Kipp¹, Tao Jiang¹, Jia Liu¹,
Mariagrazia Fugini², Monica Vitali², Barbara
Pernici², Ioan Salomie³

¹ {kipp | jiang | liu} @hirs.de

High Performance Computing Center Stuttgart, Nobelstr. 19, 70569
Stuttgart, Germany

² {fugini | vitali | pernici} @elet.polimi.it

Politecnico di Milano, piazza Leonardo da Vinci 32, 20133 Milano,
Italy

³ Ioan.Salomie@cs.utcluj.ro

Technical University of Cluj-Napoca, 15 C Daicoviciu Str., 3400 Cluj-
Napoca, Romania

Abstract— This paper presents a novel approach to characterise applications with respect to their energy consumption by using a set of energy-related metrics, called Green Metrics. These indicators are based on energy consumption measurements, such as indices of computing resource usage, of environmental impact, of development costs required to (re)design an application in order to optimise its energy consumption footprint, and of organizational factors related to application management. Our approach is framed in the GAMES (Green Active Management of Energy in IT Service centres) EU Project, with a particular focus on green IT. In this paper, we define Green Metrics enabling to characterise an application in terms of the energy it consumes at run time. Such metrics are the basis for measuring the “greenness” of an application and to detect where it consumes and wastes energy. Hints are provided to improve applications design and execution. Using two application scenarios, we show how monitoring and evaluation of Green Metrics enables to improve energy efficiency.

Keywords - energy efficiency measurement; energy-aware IT systems; performance measurement; Green Metrics

Introduction

Applications made aware of the energy they consume in terms of resources both at the software and hardware level are gaining attention in academia and industry through techniques allowing networked applications to be designed and executed to consume less energy. To sustain this attention, we are working towards methods and techniques supporting the design, development, and execution of applications along the perspectives of energy awareness [16][5]. Our work is framed in the EU project GAMES [29], aimed at developing methodologies, models, and tools to reduce the environmental impact of IT service centres. By *energy-awareness* we refer to applications being characterized by indicators, called Green Performance Indicators, tracing which resources and of what type (e.g. processor, memory, but also Quality of Service - QoS, or human resources) the application requires to run, as well as how much effort (e.g., in terms of costs) is required during development, execution, and maintenance along its life cycle. This paper focuses on such indicators and on their related metrics. In particular, the paper concentrates on IT resource usage and energy consumption. Research on metrics and measurements for green methodologies of IT and data centres is

Author

continuously in progress to achieve economical, environmental, and technological sustainability. Several sets of metrics have been proposed to measure data centre efficiency, e.g., by Green Grid [32], Uptime Institute [35], Transaction performance processing council (TPC) [33] and other initiatives. However, there is no widely accepted metric set [39], allowing for an easy measuring and monitoring of energy consumed and potentially wasted by a data centre.

In our previous work [5], we presented a new metric related to energy efficiency for a single service, which maps the relationship between energy consumption and execution time. In this way, we can compute both *quality* and *energy metrics* for each service, and can design a constraint-based quality and energy-aware service composition. In the approach presented in this paper, we consider Green Metrics for service applications from the perspective of *hardware and virtual machine usage*. We propose a comprehensive approach to Green Metrics giving an overall ‘green’ characterization of applications, through the definition of four *clusters* of metrics, allowing for a measurement of the energy consumption impact of an application in a given (logical) context. In order to highlight the impact of our metrics, we apply some of them in High Performance Computing (HPC) environment, where scientific applications are executed [40], as well as to cloud computing environment executing a business process (BP) application. Within these scenarios, energy-awareness can be evaluated and improved. Our approach starts from the observation to determine a potential for contributing to energy efficiency both at the machine (CPU, disk, I/O, Virtual Machine configuration) level and at the level of applications. For example, considering applications as business processes made as a composition of services, an application could require less resources or adapt its behaviour to its current context by running in energy-saving mode for instance, using less energy-consuming services, or low data redundancy or by renegotiating the QoS level.

We also introduce actions that can be set in place to improve the application “greenness”. Namely, several runs of the application on different configurations eventually enable to measure the values of Green Metrics (e.g., CPU utilization rate or job durations) and to identify where and for which cause energy waste occurs in the system. This can be achieved by examining various execution patterns along time through continuous monitoring of the infrastructure in GAMES. Consequently, it is possible to identify how energy waste can be avoided by setting in place a set of corrective actions, called Green Actions implementing adaptation mechanisms for applications, which makes them energy-aware. Adaptivity is a central characteristic of GAMES. By adjusting IT resource utilization, used services and data, and/or the expected QoS, applications can be executed in a more energy efficient way. The paper is organized as follows. First we discuss related work. Then, the GAMES project and the scenarios are introduced. Green Metrics are then illustrated, in particular focusing on those which have been used in the test bed presented in this paper, namely (IT resource usage and quality Green Metrics). Evaluations and examples within Business Processes (BPs) and HPC environments use are given. Finally, we present some evaluation results deriving from the GAMES framework prototype being developed in the project.

Related Work

Over the past decade, along with the rapid growth in the size, number and performance of service centres, the energy cost and the energy consumption by these IT service centres have also dramatically increased [57]. Therefore energy efficiency is a critical consideration for evaluation of computing systems, in particular with respect to high performance computing environments.

However traditional benchmarks are focused on the determination of the highest achievable computing capability with no, or partial, consideration of energy efficiency. On the contrary, it would be useful to include energy efficiency measurement into supercomputing performance benchmarking or in business applications, with the purpose of both improving

Title

the system's efficiency and of maximizing the work/applications completed over a given time slot with respect to energy consumption.

The well-known Green500 list [27] was launched in 2006 to rank the most energy-efficient supercomputers by using the high performance LINPACK (HPL) benchmark along with associated power measurement methods to measure floating point operation per watt. In order to test the major components of a HPC system including the processor, memory subsystem and interconnection to predict the real-world performance, the High Productivity Computing Systems (HPCS) [26] program at the Defence Advanced Research Projects Agency (DARPA) released the High Performance Computing Challenge (HPCC) suite of Benchmarks in 2003. HPCC examine the performance of HPC architecture using kernels with more challenging memory access patterns and provide benchmarks that bind the performance of many real applications as a function of memory access pattern characterized by the memory access spatial and temporal localities [36]. Furthermore, beyond the performance measurement, some studies have shown insight to where and how power is consumed within HPC systems. For instance, the supercomputer manufacturer SiCortex [25] has released the *Green Computing Performance Index* (GCPI) based on the HPCC suite of benchmarks for measuring system-level server energy efficiency on a broad range of performance metrics. As described in [50], HPCC benchmarks can be used for identifying application power profiles by functions and components in a high performance cluster, correlating applications memory access patterns to power consumption for these benchmarks, and exploring how energy consumption scales with system size and workload. The Standard Performance Evaluation Corporation (SPEC), a non-profit corporation which has been formed to establish, maintain and endorse a standardized set of relevant benchmarks that can be applied to the newest generation of high-performance computers [30]. California Green Business Program [28] lists a set of metrics for measuring environmental benefits of Green business practices. However, the scope of these metrics is beyond IT and data centre and considers almost all elements of business. Moreover the Transaction Processing Performance Council (TPC) [34] and the Storage Performance Council (SPC) [31] have also developed industry standard performance benchmark for different IT resources. These benchmark approaches are mostly developed by adding energy efficiency measurement to existing performance benchmarks and aiming at indicating the energy efficiency of a HPC system and its components for exploring the energy scalability. They differ from each other and focus on various dimensions: hardware components, workload and type of application, metric attributes and accuracy and calibration requirements. A detailed analysis and comparison of these energy and performance benchmarking can be found in [45] and [1].

Research towards energy efficient software began with estimating the energy consumption of the processor by instruction level power models [56]. Earlier research focused on energy efficiency of software addressed methodologies targeting on memory related power consumption [17] and apply code transformation [21]. However, in all these research works, the energy consumption of a given application is estimated upon the specific hardware architecture. Software methodologies, designs, and software development tools that can enhance the energy efficiency of application software in mobile systems are presented in [8]. According to [8], computational efficiency, data efficiency, and context awareness can contribute to create energy-aware applications. However, this paper does not focus on the energy consumption for design and management of software applications, thus missing a complete picture of energy consumption of software and service applications. The work in [49] regards a set of server energy efficiency measures and metrics. It envisions the requirements for new metrics considering Green IT as a technology to be harmonized as a hardware, software, architecture, and QoS solution. However, this approach does not take into account the entire lifecycle of an IT service centre as a whole.

Energy-related issues have been addressed also in data management and software systems groups [24]. However, energy-awareness still needs to find methods and tools in the area of

Author

BP management. In fact, various initiatives are active in the area of energy-aware computing, such as GREEN-NET [15] which describes its framework based on three components: an ON/OFF model based on an Energy Aware Resource Infrastructure (EARI), an adapted Resource Management System (OAR) for energy efficiency and a trust delegation component to assume network presence of sleeping nodes. In [58], it is discussed how organizations have recognized environmental sustainability as an urgent problem; however, still the information systems community is slow in acknowledging and tackling the problem from the software and applications viewpoint. Considering a more practical approach, the organizational supply chain is described in [51], where a sustainable IS management framework is proposed, outlining some steps based on a resource identification view. This framework suggests how to identify measures in order to control and minimize systems impacts considering ecological and social dimensions over three different types of IT products (hardware, software, and services). A green IT framework is proposed in [47], which provides clearly defined concepts related to green information technology and devises a practical implementation for sustainability-based feedback mechanisms. Our approach is built on top of existing mechanisms that provide relevant information about the infrastructure layer, in particular, information about power consumption and resource allocation/usage. Such approaches tackle the problem at a lower level, like the heterogeneous resource identification proposed by [59]. In our previous work [5], we have proposed a novel energy-aware resource allocation mechanisms and policies for business processes. These mechanisms were intended to minimize the energy consumption of the process layer, the infrastructure layer, and the control layer of a data centre. By being able to compute both quality and energy metrics for each service, we have designed a service-based process by executing a novel constraint-based quality and energy-aware service composition algorithm.

Current approaches also aim at optimising the application code by analysing and adapting it manually [9]. However, this manual processing can be optimised by using green metrics in order to determine potential energy wastes in advance. In particular with respect to HPC environments, it is essential to refer to historical energy consumption profiles, being reflected by corresponding metrics [23].

Within the GAMES project, energy benchmarks will be used to measure Green Metrics, named Green Performance Indicators (GPIs). These are an extension to the general concept of Key Performance Indicators for measuring characteristics of processes on different levels in information systems. The selection of adequate benchmarks depends on the GPIs definition and the energy consumption that should be measured on certain levels. Therefore, besides the mentioned benchmarks, other existing performance benchmarks could be used by simply adding energy measurements to measure the required metrics. Some key considerations to be taken in account when developing benchmarks for energy efficiency measurement are described in the initial Server Energy Measurement Protocol [38] released by EPA in 2006 and in the ENERGY STAR Test Procedure for Determining the Power Use of Computer Servers at Idle and Full Load (Appendix A to the ENERGY STAR specification for Computer servers 9 released by EPA in 2009) and by the GreenGrid Consortium.

Applications and their Green Metrics

Energy consumption is an intrinsic element of applications with respect to both, development and execution viewpoints. For instance, applications can be designed to save energy, introducing indicators and annotations indicating the probability of executing some branches in the program workflow, or indicating the quantities, types, and localization of used data. From the execution viewpoint, indicators related to energy consumption are indices of resource (CPU, I/O, and disk) usage on a given execution or virtualisation configuration. In this section, we investigate which indicators can be used for energy-related inspection of applications, by describing two kinds of applications which we are examining in GAMES to

Title

define such Green Metrics and to evaluate our approach: HPC and BP applications. These two use cases allow for the inspection and analysis of differing resource consumption, and thus energy consumption, profiles. First, we introduce the main features of our test applications, and then we present the Green Metrics, focusing on those that we will use in our tests.

HPC applications

HPC applications are scientific workflows which make an extensive use of IT resources, such as CPU, data storage, and parallel computing resources. Our metrics are associated to these applications to express the green properties of HPC applications at run time. In fact, HPC jobs are provided as workflows of services where the relevant issues are resource usage at run time based on different machine configurations. In our previous work ([5] and [10]), we discussed the notion of energy awareness and application annotation with respect to different levels of processor usage, data storage usage and environment, as well as user-related requirements. The rationale of examining energy is backed by the claim that “most application platforms are based on a best-effort model, which treats all requests uniformly, without any type of service differentiation or prioritization” [46]. Scientific workflows [40] allow building complex applications to manage and process large data sets, and to execute scientific experiments on distributed resources. Such application scenarios require energy-awareness since they massively use computing resources and incorporate scheduling algorithms which can be made strongly dependent on QoS vs. energy metrics. On the other side, HPC is attaining great benefits from software engineering approaches giving such applications methods and metrics for designing and maintaining them in a modular style [55]. Hence, in future work, also design time related factors are going to be included in tests of HPC applications.

BP applications

Business Processes (BPs) constitute another category of applications which provide a useful scenario where energy-awareness can be evaluated. BP applications are intended as an activity or set of activities aimed at accomplishing an organizational goal. Currently, BP engineering focuses on defining processes as composite executions of *services* (in particular Web-Services based processes). Basically, BPs are service workflows where the composition of services is the ground for creating new added-value processes, and component services are invoked dynamically. Considering BPs both at design and run time, the increased complexity of service-based software requires to be able to guarantee that the system includes up-to-date and accurate information, that the QoS agreed with services consumers is respected, and that BP management occurs constantly by continuously rethinking of its sub-processes and tasks structure to achieve efficiency and compliance to the organization requirements.

In such perspective, the analysis of energy consumption (and waste) is a relevant aspect in modern BP applications. In fact, different BP can have different behaviours about energy consumption, depending on their inherent characteristics, such as use of resources, amount of used data, or human resources/costs employed in BP development and reengineering.

In this paper, the benchmark chosen to analyse BPs under the energy viewpoint is TPC-C [53], an online transaction processing benchmark where the BP is described as a mix of concurrent transactions. TPC-C simulates a complete computing environment where users execute transactions against a database. As a working example, the benchmark describes an on-line selling service centred on five main activities (transactions): 1) entering orders; 2) delivering orders; 3) recording payments; 4) checking the status of orders, and 5) monitoring the level of stock at the warehouses.

Assumptions

In the definition of Green Metrics, we assume that the functional part (namely, the structure of the HPC/BP application) has been decided (design time), and that Green Metrics depend both on the application structure (namely, data flow, control flow, type of employed volatile object or database requirements) and on the specific hardware and software infrastructure

Author

configuration. In other words, the application is seen as a package with both inherent and IT configuration-dependent green properties.

As presented in detail in [4], our Green Metrics are structured in four clusters:

- IT resource usage metrics
- Lifecycle metrics
- Energy impact metrics
- Organizational metrics

In addition, in [4] we describe as well how these metric clusters can be applied to the logical structure of an IT service centre. The complete set of Green Metrics according to these clusters is listed in Table 1.

Table 1. Overview of Green Metrics and their Units (*costs units are related to energy consumption)

Metric	Unit
<i>IT Resource Usage Metrics</i>	
CPU Usage	<i>Percentage</i>
Storage Usage	<i>Percentage</i>
I/O Usage	<i>Percentage</i>
Memory Usage	<i>Percentage</i>
<i>Lifecycle Metrics</i>	
Lifecycle Costs	<i>Cost*</i>
Process Engineering	<i>Index</i>
Quality of Service	<i>QoS Levels</i>
<i>Energy Impact Metrics</i>	
Consumables	<i>Cost</i>
System Energy Usage	<i>KWH</i>
Supply Chain	<i>CO₂ units</i>
<i>Organizational Factors Metrics</i>	
Human Resource Usage	<i>Cost*</i>
Compliance	<i>Cost*</i>

In this paper, we will illustrate only the metrics being applied in the presented use cases and experimental results by providing a short description why and how the corresponding metric has been used. A detailed overview and definition of all referred metrics in Table 1 is given in [4].

In the category of *IT Resource Usage Metrics* the energy consumption of an application relates both to the nature of the application and to the system configuration of the run time environment. Resource consumption varies with the type of applications, depending on the processor intensiveness or data intensiveness or hybrid nature. Thus, energy consumption of an application (measured over a given time period, say between t_i and t_k) is characterized as a *function F of its resource utilization*. Resources are CPUs in computing nodes and application servers, storage devices, main memory, and I/O resources. Mapping F into resource utilization provides a link to energy consumption of the infrastructure executing the given application. All metrics in this category are provided as percentages.

CPU Usage Metric

This metric relates to the relative CPU utilization of specific applications. Therefore, the CPU Usage Metric describes the processor utilization percentage for an application, when performing (specific) computing operations.

Storage Usage Metric

Storage devices (local or remote hard disk drives (HDD)) are used when reading or writing data to permanent storage. The Storage Usage Metric denotes the entire storage utilization

Title

percentages for data read and write operations on the corresponding storage device for an application in a given configuration.

I/O Usage Metric

The I/O operations allow the application to communicate to system devices. The I/O Usage Metric describes the percentage of occupation of the corresponding I/O device for communications and the number of messages transferred by an application over a set of system components (e.g. a grid of Network Devices and Switches).

Memory Usage Metric

This metric refers to the usage of main memory (RAM). Memory units execute read and write operations regarding volatile objects in an application. The Memory Usage Metric characterizes the percentage of RAM occupation for specific application operations.

Given an application and its IT resources configuration (virtualized or real), it is expected that a good level of utilization of resources is achieved, to avoid waste of resources or to allow for an adaptation of the corresponding system configuration. An aggregate value for evaluating the greenness of IT resources can be defined using a threshold T_{res} for each of the considered resources.

In the category of *Lifecycle Metrics*, in this paper we employ the *Quality of Service Metric*.

- This metric evaluates the quality measures of an application [43][6][7][42]. We mention the two following indicators (the term service and job are used to denote a part of the workflow, such as an activity or a job portion). *Response Time*: The time taken by a service S to handle user requests (measuring the expected delay between the moment when a service request is sent and the moment when the service is rendered). Response time S is the sum of the processing time ($T_{process}$) and the transmission time (T_{trans}) as follows: $T_{response}(S) = T_{process}(S) + T_{trans}(S)$
- *Application Performance Metric*: This metric allows for the measurement of energy consumption per computing unit, e.g. #Transactions / kWh or FLOPS / kWh for a specific application type. This metric allows in particular comparing different IT service centres in a transparent fashion. This metric is given in Computation Unit / kWh.

As for the *Energy Impact Metrics*, in the present paper, we adopt the *System Energy Usage Metric*, evaluating the total energy consumed in the system running a given application.

Adaptive Applications

We refer *energy-aware adaptive* applications, when we use APIs built out of Green Metrics to characterise the energy consumption of an application and to adapt the runtime environment of this application correspondingly in order to optimise the energy consumption. Adaptive applications are difficult to design since they require monitoring APIs and applying criteria for deciding that an application is wasting energy and can run in a different mode (e.g. providing a reduced yet acceptable level of QoS). APIs group the Green Metrics, from low (system-) level to high (application- and business-) level. APIs can relate either to whole service centres or to a specific application, including its services and data, its environment, development platform, and human resources, costs, and development time.

The data related to our Green Metrics can be collected directly by physical sensors by the GAMES monitoring tools [2], and then some metrics are aggregated from low level metrics in the form of APIs in the GAMES assessment tool. This proceeding allows reflecting energy efficiency at various logical levels within an IT service centre. A detailed overview about the hierarchical aggregation and layering of APIs is given in [4]. In particular, metrics are aggregated in a hierarchical approach, allowing for the aggregation of information being collected at a low level (CPU, memory, network, etc.) and at a more abstract view of the IT infrastructure, e.g. computing nodes, HPC clusters, up to the whole IT service centre. Such a comprehensive overview allows the assessment of “greenness” of applications at different (hierarchical) levels. For instance, in order to measure the *IT Resource Usage Metrics* at the application or workflow level, the CPU, memory, storage usage rate on each computing node that is used to run the application or tasks of a workflow is measured and aggregated as a API

Author

element. Moreover, the QoS requirements of an application in terms of performance and energy efficiency need to be collected from users at a high level (e.g. in terms of "expected response time") and then to be mapped onto APIs that specify how user requirements can be fulfilled, out of the measured system level metrics.

Using Green Metrics

Green Metrics can be used to monitor applications during their whole lifecycle with a manifold aim:

- 1) to *adaptively configure the application platform*, so that the application can be tuned to be executed in a more energy-efficient way;
- 2) to capture the *energy losses* that can be associated to application code, so that these can be redesigned in a more energy-efficient way, e.g., by using evolved tools able to include application annotations, such as branching probabilities of workflow parts, or failure rates of some activities;
- 3) to certify an application at a given class of energy efficiency as a *long term objective*.

Therefore, Green Metrics have to be used according to a design and evolution methodology that:

- 1) generates APIs computed using Green Metrics,
- 2) considers Green Metrics inclusion in the application,
- 3) analysis monitored energy consumption and IT resource consumption patterns,
- 4) suggests green actions that can be applied to adapt the run-time environment as well as the application to operate at the correct quality of service level whilst taking into account the according energy consumption.

The computation of APIs based on our Green Metrics introduced in [4] and applied to concrete use cases within this paper is detailed in [12]. In the following we outline the methodological steps with focus on *IT Resource Usage Metrics* and *Quality Metrics* within our scenarios.

Green Actions for IT Resource Usage

Green Actions are a way of achieving adaptive applications for energy saving, thus also decreasing the corresponding costs. These actions aim at improving energy consumption by intervening on the metrics as follows (we list the metric and a set of possible actions directly impacting the energy consumption, which can be applied by according controlling mechanisms, e.g. [14]):

- *CPU Usage Metric*: augment CPU usage percentage, de-allocate unused CPUs, allocate CPUs to larger code portions, pre-allocate CPU to workflow activities with a higher probability of being executed (higher branching probability in workflow), balance workload of each single CPU, change CPU frequency mode.
- *Storage Usage Metric*: reduce number of records to be stored, improve read-write operations performances, extend transfer channel from-to the database system, cache larger quantities of data on the basis of last-used policies, accumulate data transfer operation at a given time moment to avoid continuous input/output, improve quality of indexed for data retrieval.
- *I/O Usage Metric*: minimize data transfer among different CPUs, reduce number of messages to devices, avoid unnecessary write operations, concentrate write operations at a given time instant altogether, expand the I/O channel data transfer rate.
- *Memory Usage Metric*: pre-allocate memory bars to large application code portions, minimize memory partitioning in pages, de-allocate libraries when unused according to a least-used policy, minimize read operations in the code, switch-off unused memory bars, accept allocation only of code portions larger than N lines (N to be determined through

Title

several runs), lower uncertainty about estimated application variable that contribute to determine memory overflow, select appropriate occupation factors for each memory component in the memory bars.

Green Actions for Quality Metrics

Considering the intrinsic heterogeneity of resources in IT service centres and their varying required capabilities of computational power and thus energy consumption characteristics, a service request for heterogeneous resources is normally allocated to available resources based on some scheduling policies. Imposing QoS constraints over service requests, scheduling policies are typically revised to achieve for optimum performance. However, in some cases it might be sensible to adjust the QoS settings of the runtime environment, including the hardware- and software environment, in order to achieve a better energy-efficiency whilst providing the correct results in time. Applying our metrics allows to determine the energy-efficiency of applications and IT infrastructures at various levels, and thus also to determine the optimal QoS settings from an energy-efficiency related viewpoint. In this way, activities of the workflow with less critical execution time constraints could be executed in slower servers, hence consuming less electrical energy and thereby reducing their overall cost/price.

Generally, consumers expect high QoS that may require higher performance, resulting in higher energy consumption. However, the expected QoS can be delivered by efficient resource utilization. Negotiating and establishing a service level agreement between consumers and providers can be seen as a way of achieving greenness in QoS. Contributions to energy can be made by trade-offs of shared processing power and lower performance.

Supporting different levels of QoS for a single activity of the workflow can make the activity run in different IT resource configurations. This kind of distribution based on differentiated QoS can eventually result in energy saving.

Service providers can pro-actively provide high QoS to the service requesters, by using different familiar approaches like caching and load balancing of service requests.

Another case regards aborting an activity execution due to a failure. Repair may be a time and energy consuming task. Thus, making an application recoverable (Recoverability above) is a desired QoS property at design time for which various repair strategies have to be defined (see for instance [22]). Sample strategies are substituting the faulty activity (more energy consuming) or redoing it (less energy consuming although another activity has to wait until the re-done activity is finalised).

In both cases, a service can be made energy efficient: if an activity has high failure rate (observed from logs over various runs), then availability should be guaranteed by a green action that adapts the execution mode for example, executing redundant activities. Conversely, if an activity has a low failure rate, substitution can be suitable at a given energy cost, while the redo/retry action can be considered as adoptable when the failure rate is close to zero.

BP Case Study

In order to test BP applications, we used an open source implementation of the TPC-C benchmark, namely TPCC-UVa [18]. This implementation reflects all the characteristics of the benchmark.

The application, together with its database, has been installed on a virtual machine. This is an important decision in setting the saving energy policy. In fact, several studies (i.e., [44] and [52]) discuss how virtualization is an important feature for saving energy, in that it allows for a guided efficient allocation of resources among all applications which are concurrently running on the system at a given time. Virtualization enables also a dynamic environment where resource allocation can change as the load of an application increases/decreases responding to different request rates. The BP application has been installed and executed on an Ubuntu10.10 virtual machine hosted in a Windows XP machine. The used hypervisor is VMWare.

Author

In detail, a set of indicators has been evaluated with several configurations for the virtual machine. The set of indicators chosen to monitor the application are related to resource usage metrics (CPU, memory, storage and I/O usage) and to QoS metrics (application performance and response time). The application performance indicator in this case is computed as the ratio between the number of transactions executed and the energy cost of the application. Data about resource usage has been collected directly from the operating system of the virtual machine running the TPC-C application. Data about QoS metrics have been retrieved from the log of the benchmark while information about energy consumption of the application has been estimated using PowerInformer [37], an open-source tool developed by Intel, able to compute power consumption by a single application.

The tested system configurations are reported in Table 2. In each test, different parameters about the virtual machine configuration have been changed in order to compare the efficiency of different configurations. The parameters that have been changed running the tests are the number of CPU cores allocated to the virtual machine running the application, the memory size and the amount of disk allocated.

Table 2. Different configurations of the virtual machine for each test

	Test1	Test2	Test3	Test4	Test5	Test6	Test7
CPU	4 cores	3 cores	2 cores	1 core	2 cores	2 cores	2 cores
Memory	512 MB	512 MB	512 MB	512 MB	1 GB	2 GB	512 MB
Disk	10 GB	10 GB	10 GB	10 GB	10 GB	10 GB	5 GB

For each of these configurations, the application was run and the previously introduced indicators have been computed. Results are reported in Table 3.

Table 3. Test results

	Application Performance (trans/kWh)	CPU usage	Storage Usage	I/O usage	Memory Usage	Response Time
Test1	13822	47.15%	38%	96.15 B/s	97.96%	0.034s
Test2	15080	67.42%	32%	191.00 B/s	93.30%	0.063s
Test3	16250	99.67%	38%	171.11 B/s	97.91%	0.048s
Test4	17354	99.67%	38%	336.61 B/s	97.96%	0.054s
Test5	14188	94.21%	38%	90.98 B/s	54.61%	0.078s
Test6	19419	91.08%	48%	252.14B/s	47.25%	0.035s
Test7	16300	67.23%	76%	103.00B/s	96.98%	0.050s

Taking into account values in Table 3 it is possible to understand the role of Green Metrics in this scenario. Virtual machines can be reconfigured, so the value of the indicators can be used to assert if the system can be improved and if resources can be better allocated to avoid wastes. As an example the application can be started in a given configuration, the indicators for the machine running the application can be computed and the state of the system can be evaluated. After that it is possible to decide if the system can be improved or not. If the initial configuration is the Test1 configuration, the according indicator shows that CPU usage is low, so too much processing power has been reserved for that machine, and this power is underemployed. The machine can be reconfigured by reducing the amount of cores allocated step by step, until a good configuration is reached (Test3 or Test4 configurations). The same

Title

reasoning can involve memory and disk usage: starting from Test6 configuration we can reduce the amount of memory allocated (Test5) until we reach an efficient memory allocation (Test3) or we can reduce the amount of disk allocated (Test7) increasing the Storage Usage indicator.

The application tested here is quite static, so the load rate is almost constant. In this case, Green Metrics can be used to find the optimal initial configuration of the system. Once this optimal configuration is achieved, nothing else can be done at runtime. In real world applications, however, the load rate changes periodically depending on time, season, etc. hence, the proposed approach can result even more useful because the system can keep on monitoring the application behaviour through measurement of the indicators, and the resource allocation can be increased or decreased depending on the load. This can ensure the optimal configuration at each moment of application running time. In order to do that, the setting of some thresholds is needed to decide in which interval the GPI values have to be improved. When the value is outside the predefined thresholds, a Green Action can be applied to improve it. For instance, in case of resource usage indicators, there can be a minimum (underemployed resource) and a maximum (over-employed resource) usage value. Similar thresholds can also be applied to the QoS indicators to ensure a given service level to the user. So, Green Metrics can result really important in understanding and improving the energy efficiency of a dynamic scenario even trying to keep a good quality for the service provided. In the business process scenario it was possible to save about the 8% of energy between Test1 and Test2 configuration.

HPC Case Study

We now illustrate the HPC case study on a cancellous bone simulation [48]. Such process can be run as a scientific workflow on different machine cluster configurations, and its greenness can be analysed. This kind of simulation enforces a high memory I/O bandwidth and CPU usage. Thus, the simulation requires special hardware architectures, e.g. a vector-based system in order to be executed with the best energy consumption per computing operation ratio. To allow a classification of the according compute job, the user would tag this computation job before submission according to its characteristics, like expected CPU Usage, Memory IO bandwidth, Disk/Network IO bandwidth and estimated job duration.

The system behaviour is analyzed according to our Green Metrics and is stored in a database. This allows for future deployments to optimize the execution of jobs and system settings, for better energy performance. Here, we take into account a virtual scenario in order to illustrate how the Green Metrics allow for an optimization of resource/service provider selection, as well as for an optimization of job deployments whilst taking into account the entire energy consumption. Obviously, the simulation job has to be run several times. In particular, depending on the available resources of the HPC Centre, the jobs are deployed on different system configurations, which perform differently from the energy consumption viewpoint, as will be shown in the sequel of the paper.

In particular, we assume that the HPC job is deployed on two different machine cluster configurations, and analyse the greenness of the system.

Evaluating energy-related metrics

The simulation job is executed several times on the available resources of the HPC-Centre. The jobs are deployed on 2 different Systems (A and B in the following). According to the mentioned metrics the values reported in Table 4 have been monitored (where Application Server Metrics stands for CPU Usage).

Author

Table 4. Estimated Energy Footprint of a Cancellous Bone Simulation System in 2 Configurations

	System		System	
	A		B	
<i>Application Server Usage</i>	(avg) 95% CPU usage		(avg) 83% CPU usage	
<i>I/O Usage Metric</i>	(avg) 8GB/s		(avg) 6 GB/s	
<i>Memory Usage Metric</i>	(avg) 93% Memory Usage		(avg) 60% Memory Usage	
<i>Application Performance Metric</i>	20 GFLOPs/KWh		18 GLOPs/KWh	

The Application Server Usage Metric allows for getting an overview about the CPU utilization and the amount of Disk I/O operations. In particular, this allows for determining if the deployment decision for the job under the current setup has been appropriate. E.g., if the job just produces a medium CPU payload, this might imply a reduction of the clock speed of the CPU without affecting the entire job execution time whilst reducing the entire energy consumption.

The I/O Usage Metric allows determining the average memory operational (I/O) performance, which is a relevant value for HPC jobs. In particular, this allows determining whether a job has a high impact on the memory performance. For example, if a job with a supposed high CPU payload causes a medium CPU payload during execution whilst outbidding the available memory I/O bandwidth, we consider that this would imply that the memory I/O channel might be a performance bottleneck. This means also that deploying this kind of job on a machine with improved memory I/O bandwidth might allow for a smoother execution of the job, and thus for a better energy efficiency of the entire job execution.

The Memory Usage Metric allows for determining the average memory usage, considering also peak usages. In particular, this metric determines if potentially too many memory modules are active in the system. In that case, for future deployments, unnecessary memory bars can be turned off and switched on again on demand, thus allowing for additional energy saving.

The Performance Metric provides a value relevant for comparison of different IT service centres against performance values according to the energy consumption.

According to the measurements reported in Table 4, System A performs better for this kind of job than System B, although System A has a higher CPU payload during the job execution whilst System B has a lower CPU payload. System A provides a better memory interaction, thus a better and more energy efficient processing for this kind of jobs. Thus, System A allows for 20 GFLOPS/KWh whilst System B just allows for 18 GFLOPS/KWh, so for future deployments of such kind of jobs System A should be chosen if possible in order to allow for better energy efficiency.

In summary, the evaluation of these measured values shows a better energy efficiency of System A for this job type due to a better Memory I/O ratio and a better usage of the available memory, whilst System B has to supply memory that is not used by the job/application. A potential solution to improve the Performance Values for system B would be switching off some of the unused memory modules.

Applying these metrics to HPC jobs is a first step toward the consideration of Green Metrics for the entire IT Service centre. This measured metrics will allow deriving high level and more abstract views on the involved system. In particular, this will allow gaining a better overview of the energy consumption at a fine granularity, according to the selected QoS needs. This will also allow the definition of high level SLA that can be mapped to low-level

Title

system metrics, hence obtaining an optimization of the entire energy consumption of the whole IT service centre.

Finally, applying these Green Metrics to the HPC infrastructure would lead to the development of new kinds of business models; in particular this would allow e.g. to offer different tariffs to the customers, depending on their deployed job types and related carbon emissions, or to negotiate service level agreements with customers, whenever lower QoS (e.g., response time) can be accepted in favour of energy savings.

Evaluation

The evaluation of the presented concepts is now given by applying two types of local control loops (LCLs) to the HPC test bed, namely the fuzzy logic and bio-inspired local control loops. Local control loops are used to optimise the energy consumption for a specific server in a service centre according to the relevant green metrics values. The energy consumption optimization process is achieved by identifying the server over provisioned hardware resources and executing Dynamic Frequency Scaling (DFS) actions to put those resources into low-power states. This section is going to concentrate on the evaluation results of applying the control loops to our testbed environment. A detailed description of the control loops is given in [14].

In order to build a baseline to compare the impact of the GAMES framework with respect to the entire energy consumption, we first have to reflect on the evaluation results of HPC use case execution without enabling the GAMES framework.

The explicit description of the HPC usage case, including the theoretical background of this use case and its computing characteristics, is presented in GAMES deliverable D7.2 [19].

For the evaluation of the GAMES local control loops, we applied three different scenarios, namely the execution of the mentioned simulation without any controller, as well as with the local control loops realising different internal reasoning, learning and decision taking mechanisms. In order to assess the differences between the executions with different local control loops, the following aspects of the IT environment have been monitored and evaluated:

- power consumption (using the System Energy Usage Metric),
- CPU Usage (using the CPU Usage Metric) and
- CPU frequency (monitoring the frequency mode of the CPU).

These values allow for a performance assessment of the energy-aware controllers. Furthermore, in order to eliminate the influences of unknown factors and disturbances, the trial has been executed 5 times for each use case with same computational conditions. Afterwards, the measured data for each category has been post-processed by computing the mean values at each sampling time for every case, accordingly. This proceeding enables us to improve the precision of the measurements for verifying and evaluating the performance of the controllers by eliminating the potential measurement errors caused by the corresponding power meters. Furthermore, although always the same computation has been executed, it is not foreseeable when a specific computing node gets occupied by a certain compute job. So, by taking into account the mean values for the entire cluster, this effect is also considered within our measurement, ensuring that the following evaluation results allow for a meaningful comparison of the corresponding execution modes. Figure 1 depicts the CPU behaviour and the related power consumption of the entire cluster by executing the cancellous bone simulation process with no use of local control loops.

Author

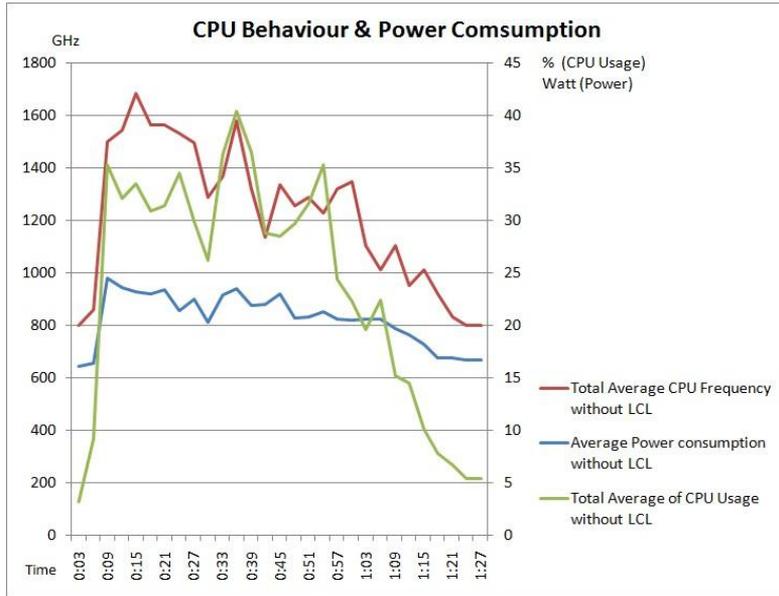


Figure 1. CPU Load and Power-Consumption without LCLs

GAMES-enabled Execution

The evaluation results of the execution of the HPC use case by applying the local control loops are as follows.

Figure 2 and Figure 3 depict the CPU performance and power consumption of the entire cluster for the fuzzy-logic and bio-inspired-learning based local control loops, respectively.

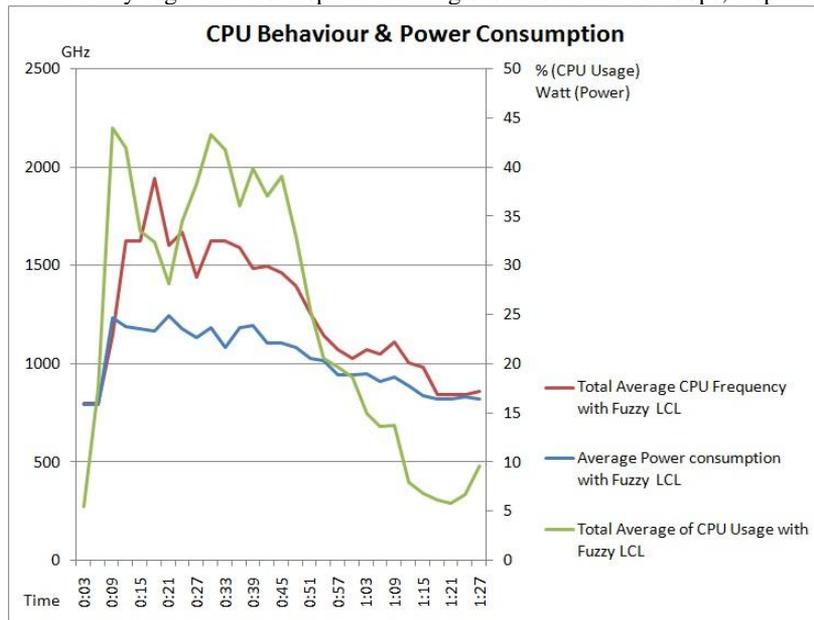


Figure 2. CPU Load and Power-Consumption with Fuzzy-Logic based LCL

Title

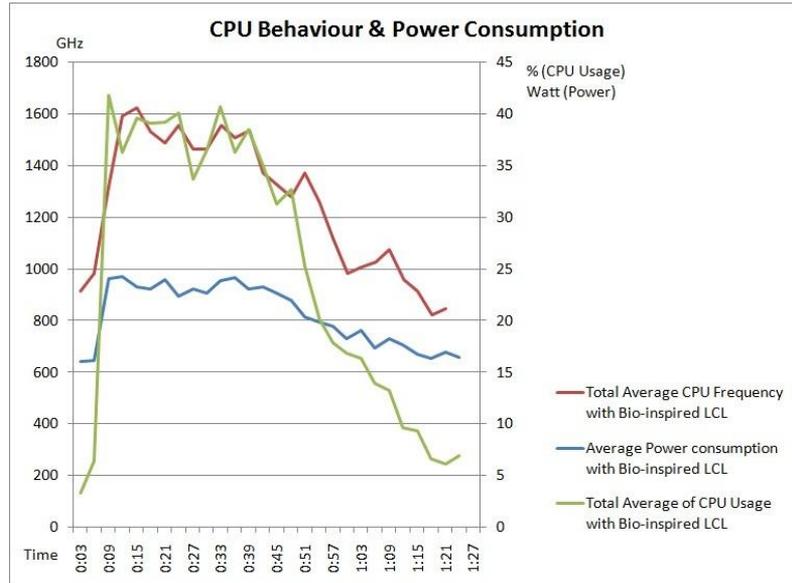


Figure 3. CPU Load and Power-Consumption with Bio-Inspired based LCL

In order to allow for a meaningful comparison of the monitored values of the evaluation runs the potential measurement error has to be taken into account as well. Therefore, we considered for the evaluation the mean values of all test runs with one specific setting by measuring the entire cluster. Since within all evaluation runs exactly the same computation has been executed the potential measurement error is considered within this mean values as well.

Since the data, in particular regarding to power consumption, has been measured in a real time environment by physical sensors and plugins, the measurement deviation has to be taken into account. For this reason, we employed the Standard Error (SE) to estimate the possible measurement error and give a scope of the error for processing data. Perceivably, the influences of both controllers can be evaluated more precisely with consideration of the SE. A detailed overview about the measured values is given in [20].

In summary, it is difficult to determine how the controllers work on each node, since, although altogether exactly the same fragments of the HPC job have been computed, the distribution of the compute jobs on the computing nodes might vary. Therefore, we considered the average values of entire GAMES cluster for several test runs. In the following, Figure 4, Figure 5 and Figure 6 depict the comparisons of the three main factors, namely power consumption, CPU utilisation and CPU frequency, in the cluster scale. As previously explained, we only use the average values to compare for removing the influences of unknown factors. The comparisons regarding CPU Usage/Frequency describe the behaviours of the cluster. In addition, the power consumption is the main criterion to assess the performance of controllers, in order to compare the effectiveness of the different settings. From the comparison of the power consumption, we determined that there are different phases in the simulation process, where the local control loops show a better behaviour in order to achieve better energy efficiency than the normal optimisation done by the operating system. These issues have been investigated and are summarised in the following.

The average power consumption of the entire evaluation with the local control loops is as follows:

- Without local control loops: 20.61 Watt per node. The standard error is 0.86 Watt
- With Fuzzy Logic Controller, 20.45 Watt per node. The standard error is 0.86 Watt
- With Bio-inspired Controller, 20.56 Watt per node. The standard error is 0.83 Watt

Author

Although this evaluation shows just a little improvement of the energy-efficiency, we determined aspects to be enacted as Green Actions, which are described in the following, allowing us to improve the entire energy-efficiency for the second release of the local control loops.

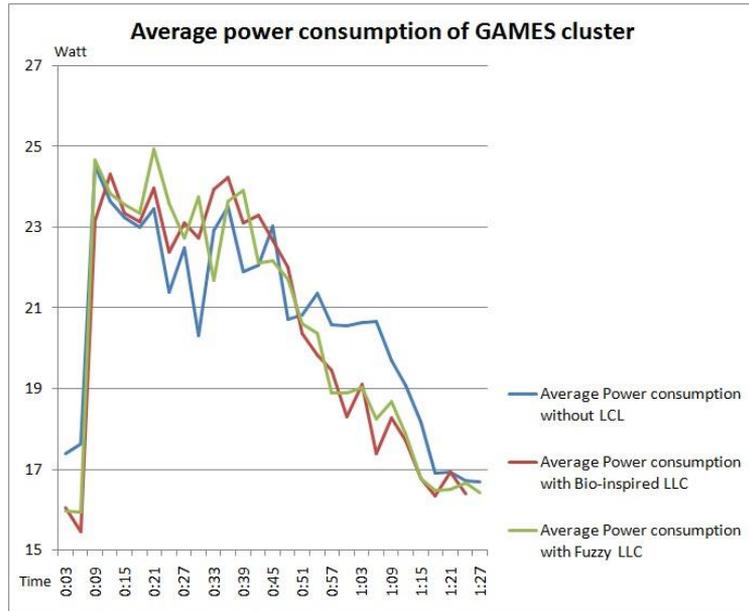


Figure 4. Average Power Consumption of the Usage Scenarios

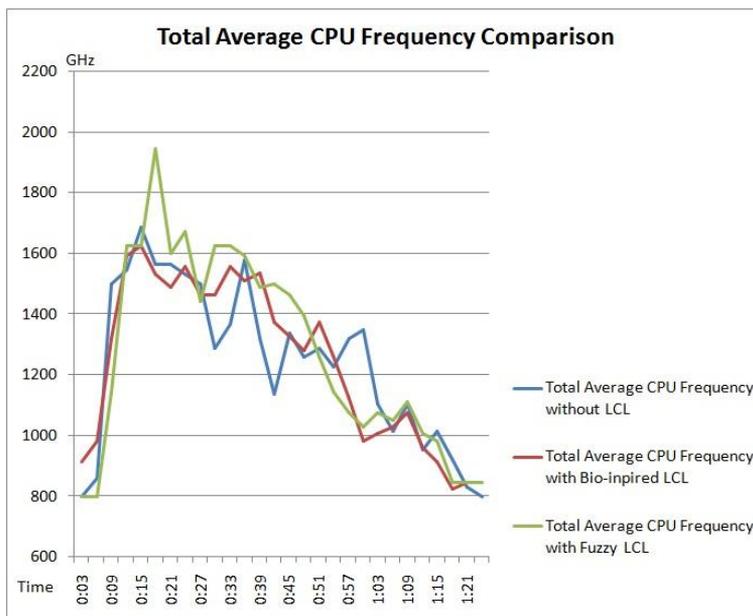


Figure 5. Average CPU Frequency of the entire Cluster per Node

Title

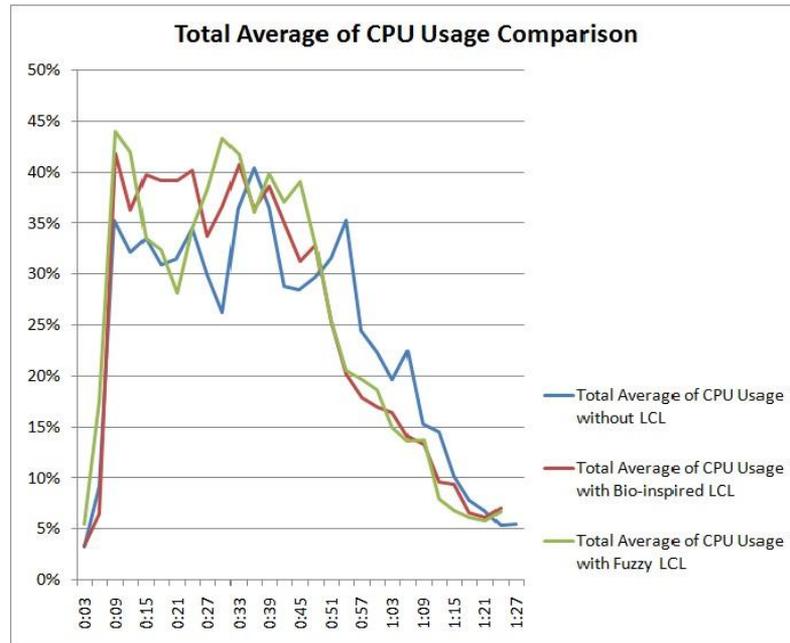


Figure 6. Average CPU utilisation of the entire Cluster per Node

The *Local Control Loops* tackle the problem of service centre servers characterized by a high energy consumption by adjusting the servers' CPU power states (p-states) according to the incoming workload variation. The first HPC 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 enable Local Control Loops with more accurate control on the CPU power consumption and in consequence better energy saving. The current HPC testbed processors Intel P8400, used for the first trials, have only 3 p-states, but for the second trials the HPC test bed servers processors will be improved to the new generation Intel i7 processors having up to 12 p-states, thus enabling for better energy saving results.

Another optimization for improving the 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 on changing processor's p-states, the Local Control Loops will take into consideration the state of the neighbour dependent processors. For example if a server's processor enters the Idle working mode, another server processor's Local Control Loop being 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. For the *Fuzzy Logic Local Control Loop*, better energy savings results can be obtained if its execution parameters will be refined so that the workload history will have a greater impact upon the decision making process. In the current test case Fuzzy Logic Local Control Loop considers only current and near past workload (one step behind) for deciding on the processor p-state changes.

The *Bio-inspired Local Control Loop* dynamic frequency scaling decisions 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

Author

learned. This exactly suits the HPC case, which deals with a relatively small number of applications.

Concluding Remarks

This paper has presented Green Metrics aimed at energy consumption monitoring in applications. In particular a BP application and a HPC simulation test bed have been presented, illustrating a subset of Green Metrics (namely, those regarding IT resource usage) applied to these examples. Furthermore, some measurements and observations about energy consumption in the test beds have been shown. Some Green Actions aimed at running HPC, cloud-based, and in general business process-based applications in a less energy consumption mode have been discussed.

Since this is part of an ongoing work, an outline of the defined metrics has been given. We are going to perform a thorough validation on more extended examples, and to provide a method to design and analyse energy-aware applications. In fact, we have mentioned the basics of the GAMES approach that enables evaluating energy consumption in applications using annotations and APIs for energy-awareness. We are refining the APIs and defining a methodology with project partners for a complete evaluation of green properties of platforms and applications, and are developing a set of GAMES prototypes that perform activities such as process and data mining to derive information about energy consumption and adjust the resource usage modes both at design and run time to achieve energy-aware adaptation.

The presented evaluation results are promising (see also [11] for more details about the BP use case), showing that the GAMES framework can have significant impacts on the energy efficiency of IT service centres. Furthermore, we identified areas of improvement for the GAMES components, allowing for achievement of better results in the second release of the GAMES framework.

Acknowledgements

This work is partially supported by funding from the Information Society Technologies Programme under the 7th Research Framework Programme of the European Union (Call: FP7-ICT-2009-4 Grant agreement no. 248514), within the research project GAMES (Green Active Management of Energy in IT Service centres). This paper 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 paper. The authors want to thank all who contributed to this paper, especially all members of the consortiums.

References

- [1] A. Fanara, E. Haines, and A. Howard. The state of energy and performance benchmarking for enterprise servers. In *Proceedings of the First TPC Technology Conference, TPCTC 2009*, Lyon, 2009.
- [2] A. Kipp, L. Schubert, J. Liu, T. Jiang, W. Christmann, and M. von dem Berge, "Energy Consumption Optimisation in HPC Service Centres," in *Proceedings of the Second International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering*, 2011, pp. 1-16.
- [3] A. Kipp, T. Jiang, and M. Fugini, "Green Metrics for energy-aware IT systems," in *Proceedings of the Fifth International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2011)*, 2011, pp. 241-248.
- [4] A. Kipp, T. Jiang, M. Fugini, and I. Salomie, "Layered Green Performance Indicators," *Future Generation Computer Systems*, vol. 28, no. 2, pp. 478-489, May 2012.
- [5] A. M. Ferreira, K. Kritikos, and B. Pernici. Energy-aware design of service-based applications. In L. Baresi, C.-H. Chi, and J. Suzuki, editors, *ICSOC/ServiceWave, volume 5900 of Lecture Notes in Computer Science*, pp 99–114, 2009.
- [6] A. Mani and A. Nagarajan. Understanding quality of service for web services. <http://www-128.ibm.com/developerworks/library/ws-quality.html>, 2002.
- [7] B. Pernici, editor. *Mobile Information Systems: Infrastructure and Design for Adaptivity and Flexibility*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [8] B. Steigerwald, R. Chabukswar, K. Krishnan, and J. D. Vega. Creating energy efficient software. <http://isdlibrary.intel-dispatch.com/isd/146/creating-energy-efficient-software.pdf>, 2007.

Title

- [9] C. Bischof, M. Christian, and C. Iwainsky, "Brainware for green HPC," *Computer Science - Research And Development*, p. 7, 2011.
- [10] C. Cappiello, M. Fugini, G. Gangadharan, A. M. Ferreira, B. Pernici, and P. Plebani. Toward energy-aware adaptive business processes, *COOPIS'10 Conf.*, Crete, Greece, Oct. 2010
- [11] C. Cappiello, M.G. Fugini, A. Mello Ferreira, B. Pernici, P. Plebani, M. Vitali, "Business Process Co-Design for Energy-Aware Adaptation", *IEEE International Conference on Intelligent Computer Communication and Processing (ICCP2011)*, Special Session on Green Computing, Cluj-Napoca, Romania, August 25 - 27, 2011
- [12] C. Cappiello, M.G. Fugini, B. Pernici, P. Plebani. Green information systems for sustainable IT, in *Information Technology and Innovation Trends in Organizations* in A. D'Atri, M. Ferrara, J.F. George, P. Spagnoletti, editors, Springer Verlag, pp. 153-160, 2011.
- [13] C. S. Park. Contemporary Engineering Economics. *Prentice Hall*, March 2010.
- [14] Cioara T, Salomie I, Anghel I, et al. A Dynamic Power Management Controller for Optimizing Servers' Energy Consumption in Service Centers. In: *Proceedings of the First International Workshop on Services, Energy % Ecosystem (SEE 2010 / ISOC2010)*.; 2011:158-168. Available at: <http://www.springerlink.com/content/p5647x53u8xj6029/>.
- [15] Costa, G.D., Gelas, J.P., Georgiou, Y., Lefevre, L., Orgerie, A.C., Pierson, J.M., Richard, O., Sharma, K.: The green-net framework: Energy efficiency in large scale distributed systems. In: *Proceedings of the IEEE Int'l Symposium on Parallel and Distributed Processing*, IEEE (2009) 1-8
- [16] D. Ardagna, C. Cappiello, M. Lovera, B. Pernici, and M. Tanelli. Active energy-aware management of business-process based applications. In P. Mahönen, K. Pohl, and T. Priol, editors, *ServiceWave, volume 5377 of Lecture Notes in Computer Science*, pages 183–195. Springer, 2008.
- [17] D. Brooks, V. Tiwari, and M. Martonosi. Wattch: a framework for architectural-level power analysis and optimizations. In *Proceedings of the 27th annual international symposium on Computer architecture*, pages 83–94, 2000.
- [18] D. Llanos and B. Palop, "TPCC-UVa: an open-source TPC-C implementation for parallel and distributed systems," in Proc. of IPDPS. IEEE, 2006, p. 389.
- [19] D7.2 - Validation Scenarios. GAMES report. Available at: <http://www.green-datacenters.eu>
- [20] D7.4 - First GAMES Trial Results. GAMES report. Available at: <http://www.green-datacenters.eu>
- [21] F. Catthoor, E. d. Greef, and S. Suytack. Custom memory management methodology: Exploration of Memory Organisation for Embedded Multimedia System Design. *Kluwer Academic Publishers*, Norwell, MA, USA, 1998.
- [22] G. Friedrich, M. Fugini, E. Mussi, B. Pernici, and G. Tagli. Exception handling for repair in service-based processes. *IEEE Trans. on Software Engineering, Special Issue on Exception Handling*, 36(2), pp.198-215, March 2010.
- [23] G. Martinovic and Z. Krpic, "Towards Green HPC Blueprints," in *Proceedings of the Second International Conference on Cloud Computing, GRIDs, and Virtualisation*, 2011, no. c, pp. 113-118.
- [24] Harizopoulos, S., Shah, M., Ranganathan, P.: Energy efficiency: The new holy grail of data management systems research. In: Proc. 4th Biennial Conference on Innovative Data Systems Research (CIDR). (2009)
- [25] <http://en.wikipedia.org/wiki/SiCortex>
- [26] <http://icl.cs.utk.edu/hpcc/>
- [27] <http://www.green500.org/>
- [28] <http://www.greenbusinessca.org>
- [29] <http://www.green-datacenters.eu>
- [30] <http://www.spec.org/>
- [31] <http://www.storageperformance.org/results>
- [32] <http://www.thegreengrid.org>
- [33] <http://www.tpc.org>
- [34] <http://www.tpc.org/information/benchmarks.asp>
- [35] <http://www.uptimeinstitute.org>
- [36] I. L. Takahashi, J. J. Dongarra, D. Koester, R. Rabenseifner, B. Lucas, J. Kepner, J. Mccalpin, and D. Bailey. Introduction to the HPC Challenge Benchmark Suite, 2005.
- [37] Intel® PowerInformer: <http://software.intel.com/en-us/articles/intel-powerinformer/>
- [38] J. Koomey, C. Belady, C. Wong, R. Snevely, B. Nordman, E. Hunter, K. Lange, R. Tiple, G. Darnell, M. Accapadi, P. Rumsey, B. Kelley, W. Tschudi, D. Moss, R. Greco, and K. Brill. Server Energy Measurement Protocol, 2006.

Author

- [39] J. Williams and L. Curtis. Green: The New Computing Coat of Arms? *IT Professional*, 10(1):12–16, 2008.
- [40] J. Yu and R. Buyya. A taxonomy of scientific workflow systems for grid computing. *SIGMOD Rec.*, 34(3):44–49, 2005.
- [41] Kipp A, Jiang T, Fugini M. Green Metrics for energy-aware IT systems. In: *Proceedings of the Fifth International Conference on Complex, Intelligent, and Software Intensive Systems (CISIS-2011)*. Seoul; 241-248, 2011.
- [42] L. Zeng, B. Benatallah, M. Dumas, J. Kalagnanam, and Q. Sheng. Quality Driven Web Services Composition. In *Proceedings of the 12th World wide Web Conference (WWW)*, 2003. Reasoning about naming systems. *ACM Trans. Program. Lang. Syst.* 15, 5 (Nov. 1993), 795-825. DOI=<http://doi.acm.org/10.1145/161468.16147>.
- [43] M. Comuzzi and B. Pernici. A framework for QoS-based web service contracting. *ACM Transactions on the Web (TWEB)*, 3(3), June 2009.
- [44] M. Marty and M. Hill, “Virtual hierarchies to support server consolidation,” in *Proceedings of the 34th ISCA*, 2007
- [45] M. Poess, R. O. Nambiar, and K. Vaid. Energy benchmarks: a detailed analysis. In *Proceedings of E-Energy 2010*, Passau, 2010.
- [46] Malay Chatterjee et al. Differential QoS support in web services management, *SOA World Magazine*, September 13, 2005.
- [47] Mann, H., Grant, G., Mann, I.J.S.: Green IT: An Implementation Framework. In: *Proceedings of the AMCIS*. (2009)
- [48] R. Schneider, G. Faust, U. Hindenlang, P. Helwig. Inhomogeneous, orthotropic material model for the cortical structure of long bones modelled on the basis of clinical CT or density data. *Computer Methods in Applied Mechanics and Engineering*, Volume 198, Issues 27-29, 15 May 2009, Pages 2167-217
- [49] S. Sekiguchi, S. Itoh, M. Sato, and H. Nakamura. Service aware metric for energy efficiency in green data centers. <http://www.iea.org/work/2009/standards/Sekiguchi.pdf>, 2009.
- [50] S. Song, R. Ge, X. Feng, and K. W. Cameron. Energy profiling and analysis of the HPC challenge benchmarks.. *International Journal of High Performance Computing Applications*, vol. 23 no. 3, August 2009.
- [51] Schmidt, N.H., Ereik, K., Kolbe, L.M., Zarnekow, R.: Towards a Procedural Model for Sustainable Information Systems Management. In: *HICSS' 09: Proceedings of the 42nd Hawaii Int'l Conference on System Sciences*, Hawaii, USA, IEEE Computer Society (2009) 1-10
- [52] T. Imada, M. Sato and H. Kimura, “Power and QoS performance characteristics of virtualized servers”, in *Proceedings of the 10th IEEE/ACM International Conference on Grid Computing*, 2009
- [53] TPC-C: <http://www.tpc.org/tpcc/>
- [54] U.S. Environmental Protection Agency (EPA). Report to Congress on Server and Data Center Energy Efficiency – Public Law 109-431. Technical report, August 2007.
- [55] V. Basili, D. Cruzes, J. Carver, L. Hochstein, J. Hollingsworth, M. Zelkowitz, and F. Shull. Understanding the high-performance-computing community: A software engineer’s perspective. 2008.
- [56] V. Tiwari, S. Malik, and A. Wolfe. Power analysis of embedded software: A first step towards software power minimization. *IEEE Transactions on VLSI Systems*, 2:437–445, 1994.
- [57] W. Forrest, J. M. Kaplan, and N. Kindler. Data centers: How to cut carbon emissions and costs. McKinsey on Business Technology, 2008. Available at http://www.mckinsey.com/client/service/sustainability/pdf/Data_Centers.pdf
- [58] Watson, R., Boudreau, M.C., Chen, A.: Information systems and environmentally sustainable development: Energy informatics and new directions for the is community. *MIS Quarterly* 34(1) (2009) 23-38
- [59] Zenker, N., Rajub, J.: Resource Measurement for Services in a heterogeneous Environment. In: *ICTTA' 08: Proceedings of the 3rd Int'l Conference on Information and Communication Technologies: From Theory to Applications*, Damascus, Syria, IEEE Communications Society (2008) 1-15