Job allocation strategies for energy-aware and efficient Grid infrastructures

Antonella Galizia, Alfonso Quarati*

IMATI-CNR, Via De Marini 6, 16149 Genova, Italy
{antonella.galizia, alfonso.quarati}@ge.imati.cnr.it

Abstract - Complex distributed architectures, like Grid, supply effective platforms to solve computations on huge datasets, often at the cost of increased power consumption. This energy issue affects the sustainability of the infrastructures and increases their environmental impact. On the other hand, due to Grid heterogeneity and scalability, possible power savings could be achieved if effective energy-aware allocation policies were adopted. These policies are meant to implement a better coupling between application requirements and the Grid resources, also taking energy parameters into account. In this paper, we discuss different allocation strategies which address jobs submitted to Grid resources, subject to efficiency and energy constraints. Our aim is to analyse the potential benefits that can be obtained from the adoption of a metric able to capture both performance and energy-savings. Based on an experimental study, we simulated two alternative scenarios aimed at comparing the behaviour of different strategies for allocating jobs to resources. Moreover we introduced the Performance/Energy Trade-off function as a useful means to evaluate the tendency of an allocation strategy towards efficiency or power consumption. Our conclusion seems to suggest that performance and energy-savings are not always enemies, and these objectives may be combined if suitable energy metrics are adopted.

Keywords: Grid, Allocation policies, Energy metrics, Performance evaluation, Energy/efficiency trade-off

1. Introduction

The increased availability of HPC systems has enlarged computational capability of e-science infrastructures. This allows researchers to deploy new experiments and solve more complex problems characterized by large computations involving great amounts of data (Michalakes et al., 2008; LHC, 2011).

Along with potentialities, these architectures bring forth a relevant Green issue: the cost associated to increased power consumption. It directly impacts on the economic balance of research centres as well as on the carbon footprint as a whole (Murugesan, 2008). As noted by Kaplan et al., 2009, a medium data centre consumes as much energy as 25,000 households. The U.S Environmental Protection Agency reported in 2007 that data centres accounted for 1.5 % (61 billion Kw/h) of the total U.S. electricity expenditure at a cost of $4.5 billion (EPA, 2007). Moreover, energy costs related to the cooling of server rooms are now comparable to hardware costs. According to an estimate provided by Gartner Inc., (Gartner, 2011), in 2007 the global ICT industry produced approximately 2% of the global CO₂ (carbon dioxide) emissions. This is a figure equivalent to the aviation industry’s emissions. In a 2008 exascale report (Kogge, 2008), DARPA foresees at least a 100MW budget for powering an exaflops machine.

Motivated by strict environmental government regulations, such as those issued in the U.S. and EU (UNFCCC, 2011; EU, 2011), scientific organizations have to make a series of strategic decisions to make their IT infrastructures adequate and thus reduce their energy balances (GreenGrid, 2009; Greenpeace, 2010). Several investigations are going on to tackle this issue from different perspectives: 1) the development of new technologies, such as lower-voltage equipment and integrated systems by green industries such as SiCortex or Tensilica Inc. (SiCortex, 2011;
Tensilica, 2011), or the design of power-aware computing practices at both hardware and software levels, e.g. the Green Flash project at Berkeley (Donofrio et al., 2009); 2) the improvement of the energy-to-efficiency ratio, by using advanced cooling technology and power generation equipment (IBM, 2011; SuperMUC, 2011), or by mixing the electricity produced from dirty or dangerous sources with renewable sources, as Yahoo and Google are doing (Greenpeace, 2011); 3) a smarter utilization of ICT/e-science infrastructure from an energy saving point of view, by taking power optimization criteria into account (ERCIM, 2009).

Grids represent both an opportunity and a challenge to e-scientists as a particular case of HPC infrastructure (Foster, 2002). Due to the heterogeneity of the involved sites, the different ownerships and access policies, the multiplicity of tools and underlying middleware, Grids are often not exploited to their full capacity, thus frequently leading to poor performance and waste of power (Iosup et al., 2006). As a result, efforts are being made worldwide to improve the management of Grid resources and to design and deploy more efficient and energy-aware scheduling algorithms (Garg and Buyya, 2009; Subrata et al., 2009).

Driven by these reasons, this paper studies the relationship between resource efficiency in executing user applications and the corresponding ecological footprint. In our prior work (Clematis et al., 2010) we presented a benchmark characterization of Grid resources aimed at supporting the job allocation task. Through simulation, we observed that a congruent allocation strategy, defined by that characterization, achieved improvements in Grid resource usage. However, in that work we only considered performance issues. In this paper we concentrate on the investigation of how the job allocation to congruent resources affects energy consumption.

The major contribution of this paper is the definition of a new metric, the Energy Congruent Index (ECI), capable of classifying resources with a combination of performance and power consumption. Based on this metric, we introduced the Energy Congruent strategy (EC) that, in our opinion, may assist in coupling efficiency with energy awareness when job allocation decisions have to be taken. To verify the soundness of our proposal, we compared EC with other alternative allocation strategies, by means of a queuing system-based simulator (Bertoli et al., 2009). As an additional contribution, to further that comparison, we introduced the Performance/Energy Trade-off function (PET) which helps analyse performance and energy savings achieved by each strategy. By examining the simulation outcomes of the different job allocation strategies, we noticed that to a certain extent efficiency and energy savings were not always in competition and that pursuing the first might help to reach the second.

The paper is organized as follows: Section 2 presents related work. Section 3 introduces our congruent approach and the ECI metric. Section 4 describes the allocation policies, the simulation setup and introduces the PET function. Section 5 exemplifies the characterization of Grid resources on a real testbed. Section 6 provides an evaluation of the proposal through two simulated scenarios. Section 7 presents conclusions and future directions.

2. Related work
Several studies have been proposed over the past decade, with the purpose of adequately addressing the environmental issues that result from the management of IT infrastructures. In the next two subsections we give an overview of two research directions that tackle those issues. Firstly we discuss different approaches adopted to save energy in Grid and distributed systems, from dynamic scaling of processor voltage/frequency to energy-aware job allocation/scheduling policies. Secondly we present the major "green indexes" proposed by the scientific community to evaluate and classify resources from an energy point of view.

2.1. Energy-aware resource management
Moving towards an energy efficient exploitation of Grid and distributed systems, the scientific community investigated this topic on different levels, considering both hardware and software possibilities. In many cases when facing energy reduction the main issue is to maintain acceptable system performance and response time.
A basic way to minimize power consumption in a cooperative system or in a parallel resource is to switch resources/nodes on and off depending on their utilization. This idea is at the base of the GREEN-NET project (Da Costa et al., 2009). The authors propose energy-aware software frameworks for large scale Grids aimed at better exploitation of computational resources with a twofold approach. On the one hand, nodes are switched off according to the estimate of their usage or reservation, and on the other the maximization of resources workload is managed with the migration of jobs from underutilized resources that can be switched off. Resource reservation techniques are adopted to aggregate multiple requests thus to avoid frequent on/off cycles. Mistral (Jung et al, 2010) is a framework designed for the dynamic management of Cloud environments. It balances power consumption, application performance, and the related costs by applying adaptation actions. Before the switching-off task, potential and cost of each possible adaptation action are estimated, as well as their impact in terms of utility values, thus the current workloads (i.e., application characteristics) and their changes are properly taken into account.

Another class of energy saving solutions leverages on the Dynamic Frequency and Voltage Scaling (DFVS) technique. As it is known, the power consumption of a computational resource is directly proportional to the processor frequency and voltage. Many efforts have been made to explore the use of DFVS policies, generally trying to compensate for the degradation in resource response time caused by the frequency reduction. In Elnozahy et al. (2002), the authors propose the use of DFVS during job execution, when resources are underutilized. Their paper discusses different strategies to manage server clusters considering the combination of DFVS and node switch-off during periods of reduced workload. The DFVS approach is also actually utilized in real time systems, i.e. systems where an efficient resource response time is essential. In Xian et al. (2007) and Chen et al. (2005), energy-aware scheduling policies for real time tasks in multiprocessor systems are defined to support DFVS. The approach is appealing also for data centres, as proposed in Quan et al. (2011). Authors evaluate the use of DFVS techniques combined with a heuristic aimed to consolidate and rearrange the allocation of jobs. This policy requires some sort of human interaction with the system. Other authors have studied energy speed and temperature interconnection. In Bansal et al. (2007), policies to set processor speed present the double objective of minimizing the energy used and the maximum temperature attained. Similarly, thermal-aware scheduling and allocation policies have been proposed in literature based on DFVS, see for example Bao et al. (2009), where the approach combines an offline temperature-aware optimization step and on-line voltage/frequency settings employing temperature sensor readings.

Further approaches have focused on the design of algorithms and methodologies for energy-efficient job scheduling. In Zue et al. (2011), the cooperation of multi-clusters is proposed through the design of an algorithm for automating the cooperation process, scheduling and resource allocation policies based on job priority analysis and resource utilization. Authors claim significant advantages in the system both on performance and energy points of views, when comparing cooperating resources with isolated ones. In Subrata et al. (2009), the scheduling of Grid jobs is addressed with a cooperative, power-aware theoretical game approach, in particular the Nash Bargaining Solution. The model takes into account different service providers and brokers, and heterogeneous connections between them. The objective is the maximization of power savings amongst all providers, while maintaining a specified QoS level. The work shows that achievable energy improvements are strictly dependent on the target QoS level and system loads. In Borgetto et al. (2011), energy-aware resource allocation is analyzed for sequential jobs on clusters. The model specifically focuses on job computational requirements, i.e. CPU and memory utilization. Authors formalize the resource allocation problem as a multi-objective optimization problem, with both performance and power consumption objectives. They propose several polynomial heuristics and algorithms to solve these problems by using both previous and novel approaches. They conclude that the choice of a heuristic depends on the specific allocation objective. An example of power reduction in Cloud environments is given by Lee and Zomaya (2010). Their proposal suggests task consolidation to increase resource utilization and thereby reduce energy consumption. They also suggest putting resources into sleep/power-saving mode. In particular, two energy consolidation heuristics are analysed: one to discriminate jobs running
alone, the other to maximize resource utilization. Both methods explicitly take into account active and idle energy consumptions with promising energy figures, according to the authors. Zikos and Karatza (2011) propose several job scheduling policies to investigate energy consumption and performance outcomes in heterogeneous clusters. These policies allocate jobs on the basis of performance or resource power profiles. By adopting a short-queue model, they simulate the scheduling behaviour of each policy and discussed each policy’s ability to pursue performance and energy goals. They conclude that each policy has its own advantages and disadvantages; the choice of one or another depends on the given objectives.

Our proposal followed the scheduling approach and presents an index-based allocation policy, namely Energy Congruent (EC). The definition of the ECI index merges performance and energy aspects characterizing the resources of a Grid system. Our policy does not consider the use of techniques such as DFVS or resource switch-off that may lead to possible variations in system configuration and even require specific privileges to manage resources. EC differentiates from other scheduling strategies as it leverages on an index which synthesizes both performance and energy in a unique parameter. In fact, the other scheduling proposals usually maximize some energy objective function and/or allocate jobs on the basis of some resource characterisation; generally energy and performance issues are treated separately. Before presenting the innovative aspects of our index, in the following, we give an overview of the main green indexes proposed by the scientific and ICT communities in the last years.

2.2. You Cannot Manage What You Do Not Measure

To effectively compare various available architectures in distributed environments, the benchmark of resources at different levels has always been considered a reasonable and efficient solution (Hockney, 1996). When energy consumption is considered, this necessity draws particular attention, due to its business implications, e.g. the cost and sustainability of IT infrastructures and data centres. For these reasons, in the last years some “green index” terminology started to appear in order to evaluate the energy appetite of IT equipment and derive useful ecological rankings (Ranganathan, 2009). However, up to now there is not a wide consensus on a common and general purpose set of energy metrics.

In the industrial ICT community, the measure most used to evaluate data centre efficiency is the Power Use Efficiency (PUE) or its reciprocal, the Data Centre Infrastructure Efficiency (DCIE). Both have been defined by the Green Grid, an industry group focused on data centre energy efficiency (GreenGrid, 2007), and can be considered as the first defined energy-related metrics. The PUE is calculated as the ratio between the total amount of power entering a data centre and the power actually used to run the computer. Thus it expresses efficiency as a measure of the percentage of power that actually gets to the resources with the rest being lost in the infrastructure (e.g. power distribution, cooling systems,...). According to a 2007 U.S. Environmental Protection Agency report (EPA, 2007), in 2006 the typical data centre had a PUE of 2.0 or higher: for every watt of IT power, (at least) an additional watt was consumed to cool and distribute power to the IT equipment. The same report depicts some energy-aware scenarios for 2011; the best estimate for a "state-of-the-art" data centre could reach a PUE of 1.2. Google indeed claims to exceed this optimistic scenario, achieving a PUE of 1.16 for their data centres with an IT load of at least 5MW and time-in-operation of at least 6 months (Google, 2011).

Alternative proposals introduced extensions to these metrics, arguing that just analyzing IT equipment power is not sufficient to evaluate data centre energy efficiency. In this direction, the Compute Power Efficiency (CPE) has been defined (Malone and Belady, 2007); it scales the DCIE by IT equipment utilization at a value between 0 and 1. The aim is to penalize the power consumed by idle servers by counting it as overhead rather than as productively used power. Similarly, the Green Grid has introduced the Data Centre Energy Productivity metric (DCEP) expressed as the useful work divided by the total facility power (GreenGrid, 2008). Kipp et al. (2011) define the Green Performance Indicators (GPi), as a set of indexes that provide a whole view of service centres, from resource usage to the costs required to develop applications. GPi define four layers of indexes aimed at a comprehensive description of energy consumption at
3. Towards More Efficient and Energy-wise Grids

Grids are characterized by their ability to provide scalable, high-performance mechanisms for discovering and accessing remote resources owned by different organizations (Foster, 2002). Because of these capabilities, wider pools of computational resources become available to scientists – besides those owned by their institutions – thus enabling more flexible, dynamic and less expensive experimental scenarios.

In order to exploit grid potential, tools such as brokers and schedulers are generally used to address the computational requests of researchers towards available and possibly suitable resources (Bai et al., 2004). The need to also consider power consumption establishes further constraints to the allocation process, asking for more comprehensive solutions. In the next two subsections we introduce our proposal to address this issue, under two apparently divergent and competing points of view: performance efficiency and energy saving.

3.1. Improving Grid Performance

A relevant task in Grid Computing is the discovery and selection of resources aimed at enabling a good match with user application needs. Grid middlewares usually offer only basic services for resource information retrieval, and are inadequate with respect to more detailed and specific user
requirements. To cope with this issue, Grid Resource benchmarking has been a topic of interest for the scientific community (Dikaiakos, 2007).

In Clematis et al. (2010), we presented a benchmark-based methodology aimed at improving the matchmaking process on the basis of information about computational resource performance. To this end we proposed integrating the information available via Grid information and monitoring services by annotating resources with both kernel and application-specific performance metrics. On these bases, we discussed the design of GREEN (GRi d Environment ENabler), a Grid service mainly devoted to discover the set of performance-ranked resources that satisfy users’ requirements. These last are usually expressed by users when submitting a job to the Grid through a job submission language. JDD, JDL, JSDL (JDD, 2011; JDL, 2006; JSDL, 2011) are Grid submission languages that, in addition to stating the application-related attributes (e.g. name and location of source code, input and output files), allow users to specify syntactic requirements (e.g. number of processors, main memory size) and ranking preferences (if any) thus to guide and constrain the matching process on resources.

Essentially, our proposal has introduced an allocation mechanism relying on a mapping between Grid resources and a set of pre-computed benchmarks thus supplying a fine grained resource description that enables users to express their knowledge precisely about the application under submission.

In further detail, if we assume the availability of \( k \) benchmarks, \( B_1, \ldots , B_k \), then for each node \( n \) of a Grid, a performance vector \( P_n = \langle p_1, \ldots , p_k \rangle \) may be assigned, which results from the execution of benchmarks \( B_1, \ldots , B_k \) on node \( n \). Generally each \( p_i \) value is expressed according to the metric associated to the corresponding benchmark \( B_i \) (e.g. MFLOPS, bandwidth in MBps, latency, etc…). As a particular case, in the following, we consider the value of \( p_i \) as the time required to execute benchmark \( B_i \) on node \( n \). This choice implies that nodes with a lower \( p_i \) have a better performance profile and therefore are preferred as target nodes to execute a job \( j \).

According to such resource characterization, when a job is submitted, the user would express its closeness to a specific benchmark \( B_h \). Let us call this job \( j_h \). On the basis of the \( p_h \) values, a scheduler may rank the resources (idle or with the shortest waiting queue) and deliver \( j_h \) to the node \( n \) with the minimum \( p_h \). As each job is precisely allocated to the resource that better performs with respect to the associated benchmark we call this allocation strategy Performance Congruent (PC). Results, gathered from a past simulation, highlighted that the careful use of application benchmark information in resource selection significantly reduces job execution times.

In the present paper, we want to investigate if and how this solution may also improve the energy efficiency of Grid environments. To this end we analysed the impact that efficient job scheduling has in decreasing the overall power consumption, by comparing PC with a ‘general purpose’ and simple strategy such as Round Robin. Furthermore, to give a significant indicator of the actual energy savings achieved by PC, we compared it with other energy-aware allocation strategies. We also focused on a simple energy allocation strategy namely LCR (Less Consuming Resource), and on the more widely used, and previously discussed, floating point operation per watt (MFLOPS/W) strategy - MFW for short in the following. Finally we considered the comparison of PC with the newly designed allocation strategy EC. This strategy inherits the congruent mechanism from PC and extends it by combining energy consumption with a detailed performance description of Grid resources.

### 3.2. The Energy Congruent Index

Our previous experience related to the Performance Congruent strategy outlined the importance of a fine grained profiling of resources to exploit Grid assets to the maximum. For this reason, when considering energy aspects, although we moved in the wake of analogous proposals (Sharma et al., 2006; GCPI, 2011), we avoided the consideration of just a single value (as in the case of the HPL derived MFLOPS/W index) or even a set of mixed kernel benchmarks (as in the case of GCPI). Instead we borrowed the idea of selecting nodes on the basis of proper job/resource matching from the Performance Congruent strategy, and designed a new strategy, EC, which couples the power
consumption of a node with a portfolio of (application) benchmark-driven performance descriptions.

Figure 1 The Energy Congruent policy allocates each job \( j_h \) to the Grid resource(s) with the lowest \( eci_h \).

Similarly to PC, we associated a performance vector \( P_n = \langle p_1, \ldots, p_k \rangle \) with each Grid node \( n \), resulting from the execution of \( k \) distinct application specific benchmarks. Then we considered the value \( w_n \), the power consumption of Grid node \( n \) (whatever obtained) expressed in Watt. This way the pairs \( \langle P_n, w_n \rangle \) are associated with each Grid node \( n \). Finally to combine the performance and the energy figures of a node \( n \), we defined the Energy Congruent Index \( ECI_n \) as the vector:

\[
ECI_n = \langle eci_1, \ldots, eci_k \rangle,
\]

where \( eci_j = p_j * w_n, j=1,\ldots,k \)

Based on the \( ECI \) definition, the Energy Congruent policy works as follows. When a job \( j_h \) is submitted (i.e. whose association with benchmark \( B_h \) has been declared) a scheduler ranks the available resources on the basis of the \( eci_h \) values, and delivers \( j_h \) to the node \( \bar{n} \) with the minimum \( eci_h \). In Figure 1 a scheduler applying EC is schematised on a restricted pool of Grid resources.

For what concerns the chosen benchmarks, different from the Green500 proposal that just considers the HPL benchmark, we used the execution times (service times) of various application-specific benchmarks. This choice depends on the easiness of obtaining the related execution time from whichever benchmark application is chosen. Note that from the definition of \( ECI \), it follows that resources are ranked inversely (i.e. in ascending rather than descending order) with respect to the MFLOPS/W metric. As to the use of a unique \( w \) power instead of the power consumed in the execution of each benchmark, as proposed by the ED metric, we agree with Feng’s previously reported suggestions. We also want to point out that in the case of distributed heterogeneous infrastructures like Grid, it may be difficult (if not impossible) to ‘read’ the actual power consumed for each separate benchmark and a unique value is often the only one available for most resources. We recognize that a slight variation of our definition of \( ECI \) may capture this alternative without significantly changing our solution. This can be easily managed by using a vector of
power values \( W_n = \langle w_1, \ldots, w_l \rangle \) to record the per-benchmark power usage and defining \( ECI \) for a node \( n \) as the scalar vector \( ECI_n = P_n \cdot W_n \).

According to the definition of ECI, for a given benchmark \( B_h \), a low \( eci_h \) value may result from the execution of \( B_h \) on a very efficient but highly energy consuming node or vice-versa. The very intuitive meaning, of \( eci_h \) is that it may be preferable to execute a job efficiently while saving energy, thus selecting (at first glance) Grid nodes with lower \( eci \) values (with respect to the job at hand). However, in general, it could be observed that in the presence of energy-savings allocation policies, relevant loss in performance may be registered. As we discuss in Section 6, the EC strategy was less subject to performance reductions, with respect to the other examined strategies. Our analysis seems to suggest that EC tends to decrease energy consumption without too much increase in execution times.

4. Allocation Policies Comparison

As mentioned above, to understand the potential of the congruent approach we simulated and compared three classes of scheduling policies: a) order-agnostic, b) Performance Congruent allocation and c) energy-aware. For each of them we analysed and compared both efficiency (times) and energy (joules) figures. According to a), a job is assigned to one of the free resources available without any specific order. In particular for this point, we considered the well-known Round Robin (RR) strategy. As to b) Performance Congruent strategy allocates each job by ordering resources on the basis of the corresponding application specific benchmarks (i.e. by means of the performance vectors). As to c) we have considered three energy-aware strategies: LCR, MFW and EC. As to LCR, jobs are assigned to the less energy consuming free resource without taking into account any performance indicator. According to MFW jobs are assigned based on the MFLOPS/W ranking (Sharma et al., 2006). Finally, EC addresses jobs taking into account the energy-congruent index, i.e. the ordering of resources based of the ECI vectors.

4.1. Simulation Set Up and Metrics

Each allocation strategy has been simulated in the context of two different scenarios. They both model a Grid environment as a two-class queuing network composed of three nodes, corresponding to a subset of a real testbed, plus a scheduler node which delivers arriving jobs according to the strategy under investigation. The study of two alternative scenarios helped us to shed light on the impact that changes, in the composition of a system (e.g. the substitution of a node), have on the results achieved by each strategy. Scheduling activity has been simulated through the Java Modeling Tools (JMT) (Bertoli et al., 2009), a suite of open source applications for performance evaluation and workload characterization. We customized the tool to implement the different ranking-based scheduling strategies.

As our aim is to evaluate the effectiveness of our two congruent strategies in a realistic realm, the simulation considered workloads composed of two kinds of parallel jobs: linear algebra and isosurface extraction. Two application benchmarks have been associated and values obtained through experimentation (see Section 5.1). This correspondence is at the basis of the congruent allocation mechanism as explained in Section 3.1 and Section 3.2: when a job \( j_n \) is submitted, the user expresses its closeness to a specific benchmark \( B_h \).

Several studies have focused on the complexity of determining the workload characteristics (e.g. frequency of job arrival, size and degree of parallelism, etc...) for Grid environments (Iosup and Epema, 2011; Musculus and Wolters, 2006). However we followed some simplification hypotheses also suggested by other authors, and we have modelled the two-application workload by means of a queuing network with two open classes (Lazowska et al., 1984). Classes are characterized by an inter-arrival time with a Poisson distribution (Nou et al., 2007) and a service time with an exponential distribution (Berten and Gaujal, 2007; Bossendroek, 2009). Service times have been obtained through a real experiment which gave average execution times as reported in Table 4 (see Section 5.1).

To evaluate the impact of the different resource rankings, given a network of \( N \) nodes, for each strategy we have considered the following performance metrics:
- $R_j$, the average residence time (in seconds) per node;
- $X_j$, the average throughput per node;

collected for each node $j$ ($j = 1, \ldots, N$);

- $RT$, the average system response time.

The $R_j$ accounts for the time a job takes to be executed (service time) plus the time spent in queue (waiting to be executed). As our analysis focuses on the matching of suitable machines through the benchmarking approach, all the simulated strategies are based only on performance and/or power. These strategies did not take into account the role of specific queue policy, that are generally investigated by scheduling algorithms (Papazachos and Karatza, 2010). We adopted a simplified solution, and for any node we supposed to have a queue with infinity capacity, managed by a ‘First Come First Served’ policy.

Furthermore we computed the average energy consumption $E$ (expressed in joules) of the system, as follows:

$$E = \sum_{j=1}^{N} R_j \times X_j \times w_j$$

where, $w_j$ (expressed in Watt) is the power consumption of node $j$, with $j = 1, \ldots, N$.

| Table 1 Abbreviations and Notations of Metrics and Policies. |
|-----------------|--------------------------------------------------|
| ECI             | Energy-congruent index                          |
| RR              | Round Robin strategy                            |
| PC              | Performance Congruent strategy                  |
| LCR             | Less Consuming Resource strategy                |
| MFW             | Mega Flops per Watt strategy                    |
| EC              | Energy-congruent strategy                       |
| $R_j$           | Average residence time of node $j$              |
| $X_j$           | Average throughput of node $j$                  |
| $w_j$           | Power consumption of node $j$                   |
| $E$             | Average system energy consumption               |
| $RT$            | Average system response time                    |
| PET             | Performance/Energy Trade-off function           |
| PIL             | Relative loss in performance improving          |
| ESL             | Relative loss in energy savings                 |

Table 1 summarizes the notations and abbreviations used to identify the several metrics and policies that are used in this paper.

4.2. Performance/Energy Trade-off

A clear evaluation of the benefits of performance and energy savings brought about by an allocation strategy often breaks into the trade-off between two competing goals. As noted by other authors (Buyya et al., 2010; Subrata et al. 2010) the actual challenge is trying to maintain a satisfactory level of performance (e.g. to satisfy some QoS requirement) without increasing the energy costs. Given a collection of scheduling strategies (e.g. RR, PC, LCR, MFW, EC), we introduced a trade-off function $PET$ (Performance/Energy Trade-off). It is aimed to comprehend the outcomes of each strategy and to support the ex-post analysis about the convenience of adopting an allocation solution. $PET$ combines the contribution of a strategy in terms of performance and energy savings, by generating a synthetic value. This value summarizes, for a
strategy $s$, the ability to contemporaneously satisfy the two objectives, where $\tau \in [0,1]$, is named the *trade-off* parameter. *PET* is defined as follows:

$$PET(\tau, s, u) = \tau p_u(s) + (1 - \tau) e_u(s)$$  \hspace{1cm} (1)

The factors $p_u(s)$ and $e_u(s)$ respectively highlight the relative performance and energy contribution of a strategy $s$, ($s \in S$, with $S$ a set of scheduling strategies), when applied to a system under a utilization factor $u$, and are given by the expressions:

$$p_u(s) = \frac{\text{min}\{RT_u(s)\}}{RT_u(s)}$$

$$e_u(s) = \frac{\text{min}\{E_u(s)\}}{E_u(s)}$$

where:

- $RT_u(s)$ is the (simulated) performance value of strategy $s$ at utilization factor $u$ (i.e. response time in seconds);
- $E_u(s)$ is the (simulated) energy consumption of strategy $s$ at utilization factor $u$ (joules);
- $\text{min}RT_u = \min\{RT_u(s), s \in S\}$ is the best performance value (i.e. lower execution time), achieved by a strategy $\tilde{s}$ at utilization factor $u$;
- $\text{min}E_u = \min\{E_u(s), s \in S\}$ is the lowest energy consumption, achieved by strategy $\bar{s}$.

By definition, we have $p_u(s)$, $e_u(s)$, $PET(\tau, s, u) \in [0,1]$, for any $\tau, s, u$. Moreover for strategies $\tilde{s}$ and $\bar{s}$, we have $PET(1, \tilde{s}, u) = PET(0, \bar{s}, u) = 1$.

Factors $p_u(s)$ and $e_u(s)$ testify the ability of a given allocation strategy $s$ to satisfy performance and energy requirements or, in other words, its closeness to the best performance and energy strategies (amongst a given collection). By varying $\tau$ it is possible to analyze the expected impact of the two objective metrics resulting from each strategy, when different weights (i.e. preferences) are given to one or the other. High values of $\tau$ mean that performance (i.e. the $p_u(s)$ factor) is preferred to energy savings and vice-versa. The *PET* function may help to clarify the potential of a strategy to accomplish performance and energy reduction objectives and to choose between alternative allocation solutions. To further explain its intended use, we re-write expression (1) as a linear equation (2):

$$PET(\tau, s, u) = \tau[p_u(s) - e_u(s)] + e_u(s)$$  \hspace{1cm} (2)

According to (2), given a strategy $s$ we can draw its *PET* function as a straight line on the ($PET$, $\tau$) axis plane, having slope $m = [p_u(s) - e_u(s)]$, and intercept $q = e_u(s)$. By just examining the sign of $m$, it is possible to obtain some information about the ability of a strategy $s$ (operating on a system under a given load $u$), to be more or less performance or energy oriented. A negative slope results from a more energy-oriented strategy, and vice-versa. In fact, we can see by Figure 2, that when $m < 0$ (i.e. $p_u(s) < e_u(s)$), a strategy $s$, diminishes the *PET* values, from its maximum $e_u(s)$ to $p_u(s)$ when $\tau$ increases, i.e. when performance is gradually chosen as the main objective of the allocation policy. In this case we can say that the energy-savings potential of $s$, is (partially) lost due to the preference given to performance. Conversely when $m > 0$, a strategy $s$, increases its overall effectiveness accordingly to the increase of $\tau$. If we look at the magnitude of the slope, we can obtain information about the size of the trade-off between the performance and the energy outcomes of a strategy $s$. The greater $m$ is, the deeper the gap between the two objectives. Therefore a relatively gentle slope may testify to the ability of a strategy to be more
fair in pursuing either goals, in respect to another with a greater \( m \). This is the case in strategy \( s_e \) in Figure 2 for example.

![Figure 2 PET functions of performance (s_p) and energy (s_e) oriented strategies.](image)

When considering two strategies \( s' \) and \( s'' \), the role of \( \tau \) is to help decide which is the preferable choice, not in absolute terms (i.e. performance or energy) where the respective metrics \( RT \) and \( E \) suffice, but according to their combined behaviour in terms of the two (competing) goals. If we examine the situation depicted in Figure 3, where both \( s' \) and \( s'' \) are performance oriented strategies (i.e. with positive slopes), we can see that for values of \( \tau \) smaller than the breaking point \( \bar{\tau} \), strategy \( s' \) shows better energy attitude than \( s'' \), thus for these values this strategy will be preferable to \( s'' \). This choice, however, is not neutral (e.g. system independent) but depends on the priority given to performance and energy requirements. For example, in case of heavy system loads, when resources are greatly utilized and cause greater response times, it should be preferable to select a performance oriented strategy which reduces users’ waiting time (jobs delay) but (if possible) still preserves some energy saving. In this case a strategy such as \( s'' \) will be reasonably preferable to \( s' \) due to its higher performance value \( p_u(s'') \). Note, however, that in Figure 3 the depicted \( PET \) functions are purely theoretical and do not refer to any specific utilization factor \( u \). Indeed as we discuss in Section 6, a proper comparison between alternative strategies cannot disregard this factor. Moreover we want to point out that for a given strategy \( s \) both \( p_u(s) \) and \( e_u(s) \), and hence \( PET \), strictly depend on the values obtained by the other strategies examined, all evaluated under the particular system workload. Finally, as we see in Section 6, \( PET \) is affected by the real capacity (both computational and energetic) of the resources involved.
Figure 3 Comparison of two performance oriented strategies by varying $\tau$ values.

For all these reasons the use of the PET function in this paper is only to assist our analysis of the different outcomes produced by the allocation strategies under simulated comparison. We recognize its validity in such a context of use and suggest its adoption in similar cases. However, we are aware of the efforts to design and implement a tool capable of autonomously selecting the best allocation policy in a dynamically changing working scenario. In a Grid environment such component may leverage on workload information supplied by services like the Monitoring and Discovery System (MDS) provided as basic service in Globus Toolkit (Globus, 2011), the Information System of gLite (gLite, 2011), or even exploit more advanced tools such as Hawkeye (Hawkeye, 2011), the monitoring tool developed by the Condor group.

5. Benchmark Driven Resources Characterization

Aimed at achieving well-grounded simulation outcomes, we conducted an experimental analysis on the behaviour of a four-resource testbed\(^2\) profiled under two application specific benchmarks. These actual measurements served as a basis to rank the nodes of the simulated infrastructures modelled through two distinct scenarios. Our goal was to describe the actual performance offered by these systems along different metric axes. Table 2 highlights the architectural heterogeneity of our testbed, especially in regards to computing capacity (number of cores), type of interconnection and memory, as well as power consumption. This is not casual, since we wanted to reflect the technical differences occurring in e-Science environment such as Grid.

<table>
<thead>
<tr>
<th>Table 2 Testbed infrastructure.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proc. Type</td>
</tr>
<tr>
<td>IBM</td>
</tr>
<tr>
<td>Michelangelo</td>
</tr>
<tr>
<td>Paperoga</td>
</tr>
<tr>
<td>Cluster1</td>
</tr>
</tbody>
</table>

\(^2\) Cluster1 and Paperoga are parallel machines belonging to our domain, Michelangelo has been accessed within the project LitBio (Litbio, 2011) and the IBM blade cluster is exploited in collaboration with the DIST Department of the University of Genoa (the resource has been granted within the IBM Shared University Research (SUR) Awards (SUR, 2011)).
5.1. Performance and Energy Profiling

As explained in Section 3.1, our congruent approach leverages on the ranking of Grid resources by a performance basis. This ordering is obtained by expanding the description of resources with indicators that record their reaction under different types of workloads (Clematis et al., 2010). To measure performance, we introduced a benchmarking methodology that integrates two complementary solutions: 1) the use of kernel-benchmarks, capable of supplying basic information about resources, based on low-level performance metrics; 2) the exploitation of application specific benchmarks, to get a closer insight into the behaviour of resources with respect to a class of applications, under more realistic conditions. The former are used to derive metrics for basic operations such as floating point operations, memory accesses, etc… For these measurements, standard benchmarks have been employed. The latter are helpful to describe performance in terms of some applications of interest (e.g. linear algebra, computational fluid dynamics, computational chemistry, etc…) usually submitted to the Grid environment. This approach was based on the assumption that a user will specify the most similar benchmark when submitting a job, through some job submission languages like JSDL, JDD or JDL. The selection of a benchmark may be done according to the user’s knowledge about the job (e.g. its functional behaviour, more stressed system component, the kind and size of input, etc.) or the presence of application benchmarks. When this information is supplied, jobs can be addressed towards those resources that show better performance with respect to the selected benchmark.

A more detailed discussion about the opportunity of a two-level benchmarking methodology and its deployment in Grid environments as a supporting means to the matchmaking task is beyond the scope of this paper. However, we want to point out that application-driven benchmarks are generally more suitable to mimic a real job workload due to their proximity to the application at hand. For this reason, in the following we have focused on two application benchmarks: High Performance Linpack benchmark (HPL, 2011) as a case of linear algebra code and a sequential version of an isosurface extraction parallel (ISO) application (D’Agostino et al., 2011). These two benchmarks have been taken as a basis to compute the performance values of each node in the testbed, and then reported in the performance vector and the ECI index. For both benchmarks, we considered the time required to complete a run under specific conditions and thus obtain a feasible computational cost. We have selected reference datasets and/or specific parameters: for example, ISO processes a small 3D dataset of 16 MB, and produces a mesh made by 4 million triangles. Table 3 presents average execution times for ISO and HPL.

<table>
<thead>
<tr>
<th></th>
<th>IBM (32)</th>
<th>Michelangelo (32)</th>
<th>Paperoga (8)</th>
<th>Cluster1 (16)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ISO</td>
<td>1</td>
<td>1.2</td>
<td>4.9</td>
<td>3.4</td>
</tr>
<tr>
<td>HPL</td>
<td>13</td>
<td>10.7</td>
<td>20.7</td>
<td>23.1</td>
</tr>
</tbody>
</table>

In Table 4 we reported the average execution times of the isosurface extraction (isoextraction for short) and linear algebra applications, employed to solve real scientific problems on the four nodes testbed. These values have been used during simulation to compute the performance and energy metrics (i.e. $X$, $R$, $RT$ and $E$). It is noteworthy that the comparison of this data with that reported in Table 3, highlights a ratio of about 40%. This is not casual, and it is due to the choice of adopting reduced dataset and/or specific parameters to benchmark resource performance. Indeed this was actually our objective, i.e. to obtain reduced execution times that are still representative of the related applications.
Table 4 Average execution times of the two applications.

<table>
<thead>
<tr>
<th></th>
<th>IBM</th>
<th>Michelangelo</th>
<th>Paperoga</th>
<th>Cluster1</th>
</tr>
</thead>
<tbody>
<tr>
<td>isoextraction</td>
<td>2.4</td>
<td>3</td>
<td>13</td>
<td>7</td>
</tr>
<tr>
<td>linear algebra</td>
<td>33</td>
<td>25</td>
<td>55</td>
<td>62</td>
</tr>
</tbody>
</table>

The energy profiles have been obtained by analysing the real power consumption of each machine. The Power column in Table 2 lists the average values read with an energy cost meter during the execution of diverse consuming applications. This method, according to the suggestions of Sharma et al. (2006), was preferred instead of analyzing consumption only under specific loads as more representative of a general machine activity. The power values of each node have been used to obtain the ECI indexes and the other two metrics used by the energy strategies LCR and MFW, i.e. Less Consuming Resource and Mega Flops per Watt. According to LCR, resources are ranked solely based on their power consumption. MFW follows the Green500 approach, and ranks resources based on the MFLOPS/W ratio. We computed MFLOPS values as a result of the execution of the Flops benchmark (Flops, 2011).

Figure 4 depicts the ranking of the testbed nodes according to the application benchmarks and energy indexes (based on a scale from 1 to 4). During simulation, these ranks supported the routing decisions of the scheduler according to the various strategies under examination. A quick look at Figure 4 reveals that IBM and Michelangelo nodes alternatively ranked first and second respectively, and outperformed the other two when only considering the execution times results of the two application benchmarks ISO and HPL. The energy rankings indicate the near supremacy of IBM (3 times on 4). This is essentially due to the better energy and performance figures IBM achieved, synthesized in both ECI and MFW indexes. Michelangelo directly follows IBM for almost the same reason, and its relatively worse ranks are due to its very poor energy profile (2800 Watt) even if balanced by good performance. It is clear that the less powerful nodes (i.e. having fewer and slower cores), cannot reasonable compete – in these rankings, except for the LCR case, where Paperoga ranked first because of its lowest power consumption.

Figure 4 Ranking of the four-resource testbed according to performance and energy metrics (on a 4 points scale): the higher the rank, the better the efficiency of the machine.
5.2. Grid Resources Tagging

In order to benefit from the several ranks resulting from the benchmark activity, a Grid scheduler needs a usable description of available resources thus to select the most suitable one for user requirements.

The GLUE schema (Andreozzi et al., 2009), deployed by the GLUE Working Group, is a conceptual model of Grid entities which consists of a set of specifications for Grid resources. Implementation through an XML Schema given in (Andreozzi et al., 2008) provides a (simple) benchmarking characterization of resources by defining the Benchmark_t complex type. By using the extension mechanism defined in the GLUE specification, we augmented Benchmark_t by adding three sub-elements (namely BenchLevel, Order and Power) used to deploy our benchmark-based allocation approach. BenchLevel reports the kind of benchmark considered (i.e. kernel, application) or the presence of an energy metric. The Order sub-element is used to establish the direction (i.e. ascending or descending) where resources are to be ranked. For all the metrics discussed here, except MFW, this field is set to ascending. Finally, the Power sub-element records the power consumption. Note that the choice to associate a (distinct) power value for each benchmark is in contrast with our decision to assign a unique power value for all benchmarks as previously discussed (see 3.2, 5.1). However, we considered that the general case can easily deal with alternative scheduling scenarios. An example of a Glue description of the benchmark elements related to resource Paperoga (with respect to the ISO, HPL and MFW indexes), is given in Listing 1.

```
<Benchmark>
  <LocalID>paperoga.ge.imati.cnr.it</LocalID>
  <Type>ISO</Type>
  <Value>4.9</Value>
  <BenchLevel>application</BenchLevel>
  <Order>ascending</Order>
  <Power>1450</Power>
</Benchmark>

<Benchmark>
  <LocalID>paperoga.ge.imati.cnr.it</LocalID>
  <Type>HPL</Type>
  <Value>20.7</Value>
  <BenchLevel>application</BenchLevel>
  <Order>ascending</Order>
  <Power>1450</Power>
</Benchmark>

<Benchmark>
  <LocalID>paperoga.ge.imati.cnr.it</LocalID>
  <Type>MFW</Type>
  <Value>3.255</Value>
  <BenchLevel>energy</BenchLevel>
  <Order>descending</Order>
  <Power>1450</Power>
</Benchmark>
```

Listing 1 Benchmark elements associated to node Paperoga.

This resource description, to be effective, requires that a counterpart of this extension, related to the benchmark specification selected by users, is supplied at the job side. To this end we applied an analogous enrichment to the JSDL schema, probably the more widely adopted job submission language used to submit jobs to the Grid (Clematis et al., 2010).
6. Discussion

Considering the typical heterogeneity of Grid environments, it is particularly relevant to evaluate the behaviour of each allocation policy when significantly varying the set of available resources. Given our quite reduced testbed, we proceeded by simulating two different scenarios, each composed of three nodes, where the two machines with higher and lower energy impact were alternatively included. Indeed we noted that Paperoga consumes almost half the power absorbed by Michelangelo (i.e. 1450 W vs. 2800 W). On the other hand Paperoga proved to be four times slower in executing isoeXtraction jobs (13sec vs. 3sec) and more than twice as slow in executing a linear algebra job (55sec vs. 25sec) than Michelangelo (see Table 4). These significant differences, due to the small size of the modelled networks, heavily affected the outcomes of the system as a whole and resulted in scenarios more or less inclined towards energy savings or efficiency.

6.1. Energy Oriented Scenario

We started our analysis by considering a network of three nodes simulating the behaviour of a system composed of IBM, Paperoga and Cluster1 machines. This sample is characterized by strong differences in respect to architectural configuration (e.g. 32, 8, 16 cores), and in performance and energy profiles (see Table 2-4 and Figure 4). Table 5 reports the values for the different metrics that drive the corresponding strategies during the (simulated) scheduler allocation activity. These parameters were derived from the experimental study carried out on our testbed.

In regards to EC, for each node $n$ we computed the $ECI_n$ vector $\langle eci_{ISO}, eci_{HPL}\rangle$ according to the two kinds of arriving jobs (e.g. we recall that $eci_{ISO} = p_{ISO} \times w_n$). For example, for the IBM node we have: $ECI_{IBM} = <1s * 1900W, 13s * 1900W >=<1900 J, 24700 J>$.

<table>
<thead>
<tr>
<th></th>
<th>PC</th>
<th>LCR</th>
<th>MFW</th>
<th>EC</th>
</tr>
</thead>
<tbody>
<tr>
<td>p_{ISO}</td>
<td>1s</td>
<td>13 s</td>
<td>1900 W</td>
<td>10.69 MFLOPS/W</td>
</tr>
<tr>
<td>p_{HPL}</td>
<td></td>
<td></td>
<td>1450 W</td>
<td>3.25 MFLOPS/W</td>
</tr>
</tbody>
</table>

In Figure 5 the performance of each strategy at increasing workloads is shown as the average system Response Times ($RT$) (in seconds). For sake of clarity, we did not report the results of Round Robin (RR) for utilization of 100%. It is evident that the Performance Congruent strategy overcomes any other (up to 4 times w.r.t RR). This is not surprising since PC allocates each submitted job, to the free resource that better performs with respect to the associated benchmark. To understand the higher $RT$s employed by EC with respect to PC, consider for example an isoeXtraction job $j_{ISO}$. Based on the $eci_{ISO}$ values, Energy Congruent strategy, if IBM is busy, allocates $j_{ISO}$ to the slower Paperoga (which on average takes 4.9 seconds to execute the ISO benchmark) but with $eci_{ISO} = 7105J$, instead of Cluster1 (only averagely employing 3.4 seconds) which has an $eci_{ISO} = 7480J$. The quicker Cluster1 is instead selected by PC which considers the ISO benchmark execution times. Similar observations can be made to compare PC with LCR and MFW strategies. LCR is penalized by assigning all jobs first to the slower Paperoga and only afterwards to IBM, if the former is busy. MFW behaves like PC when IBM is free (i.e. they have the same best ranking for this resource), but if IBM is busy, the second MFW’s choice is Cluster1 independently from the kind of submitted job. In the case of a linear algebra job, MFW will assign it to Cluster1 which is slower in executing this kind of application than Paperoga (selected by PC) thus increasing the average system $RT$. 

Table 5 Observational parameters which feed simulated allocation policies.
To further outline the contribution of each strategy in the reduction of the system Response Times, Figure 6 highlights $PIL(s)$, the relative loss in performance improvement (i.e. decreasing in Response Times) with respect to PC (i.e. the best performance strategy), that occurs when any other strategy is adopted instead of PC. For each strategy $s$ we have:

$$PIL(s) = \left(1 - \frac{RT(\text{PC})}{RT(s)}\right)$$

(3)

Due to the excessive times obtained by Round Robin, we did not think it was worthwhile to report its increases in Figure 6. Certainly RR results compared to those of PC (Figure 5) outline
the greater advantage of using a performance driven scheduling policy such as PC. LCR is always worse than the other two with a performance loss of about 26% with respect to PC (at 20% system utilization). By examining Figure 6, the superiority of EC compared to LCR and MFW is quite clear. We also note that the worst performance of EC compared to PC is at 20% utilization, where the use of EC causes a 9% reduction in performance improvement (i.e. PC employed 9% fewer execution times). To understand the greater ability of EC in supporting performance let us consider – for example – LCR. LCR tries to allocate all jobs (i.e. independently of their type) first to Paperoga (if free) and only afterwards to IBM (which performs better for any kind of workload). For this reason this allocation strategy causes higher residence times due to the more demanding services times supplied by Paperoga (for both applications).

Let us consider now the energy outcomes of the five strategies. Figure 7 reports $E$, the average energy consumption per seconds (in joules), obtained by adding the consumption of each node. The Round Robin strategy, in accordance with the already worst performance in Response Times, produced the highest energy consumption (as done for $RT$, to improve readability, RR values are not shown for utilization of 100%). Apart from EC which provides the lowest consumption for any utilization factor, the other three strategies appear more or less aligned to the same values. To understand the better results of EC as compared to PC, let us consider again the case of an isoextraction job $j_{ISO}$. Energy Congruent assigns $j_{ISO}$ to the lower consuming Paperoga instead of the faster but more energy consuming Cluster1, selected by PC. This resulted in the slightly higher $E$ returned by PC.

![Figure 7 Energy consumption at increasing loads.](image)

In order to highlight the difference between strategies, we reported in Figure 8 (in percentages) the relative loss in energy savings ($ESL(s)$), with respect to EC, that occurs when any other strategy is adopted instead of EC. For each strategy $s$ we have:

$$ESL(s) = \left(1 - \frac{E(EC)}{E(s)}\right)$$

(4)

The ESL graph of Figure 8 reveals that PC showed the worst results among the three (except at 80%), with a loss in energy savings almost up to 6% as compared to EC. MFW has better figures (except at 40% utilization) as compared to the other two strategies, in particular for utilization greater than 60%. From Figure 5 and Figure 7, it is quite clear that PC and EC outperform all other
strategies, when considering performance and energy respectively. Furthermore EC is the best, just after PC, when only performance is considered also (Figure 6). Thus, in any real allocation scenario, the choice of one or the other strategy may reasonably depend on the relevance assigned to performance with respect to energy savings.

![Figure 8](image-url) Relative energy savings losses with respect to the EC strategy.

As explained in Section 4, the definition of the PET trade-off function aims to support this decision problem. In the following, to exemplify the PET usefulness, we analyse its results at the loading extremes (i.e. 20% and 100% utilization). This choice is especially motivated by the fact that reasonably, at maximum loads, the first concern is performance (i.e. not a further increase in the response times) – while, when systems are almost underutilized, energy savings may be a major incentive to address job allocation tasks.

![Figure 9](image-url) Performance/Energy Trade-off function at 20% utilization, by varying the trade-off parameter $\tau$.

By varying the $\tau$ parameter the Performance/Energy curves in Figure 9 reveal the trade-off between performance and energy showed by the allocation strategies, when considering a utilization factor of 20%. First of all we notice that, according to what was explained in Section
4.2, the three energy oriented strategies show a negative slope, while PC a positive one. Moreover, we see that \( PET(1, PC, 20) = PET(0, EC, 20) = 1 \), being PC and EC the best performance and energy saving strategies, respectively. As expected, EC is always better than LCR and MFW while it is outperformed by PC for values of \( \tau \) (circa) greater than 0.4. The comparison between MFW and LCR reveals that the former, which combines performance with energy, is always better than LCR which exclusively exploits energy resource characteristics. Indeed the LCR PET graph shows a great (negative) slope; this fact reveals the very poor performance outcome of this strategy \( (p_{20}(LCR)) \) as compared to its somehow acceptable energy profile \( (e_{20}(LCR)) \). These facts have already emerged from the distinct analysis of Figure 6 and Figure 8 but are, without a doubt, more immediately synthesized by Figure 9. If we consider a different utilization factor, the breaking point (i.e. the point on the \( \tau \) axis at which the two strategies have the same value, according to \( PET \)) between PC and EC, makes \( \tau \) shift from about 0.4 to about 0.7 (Figure 10).

In order to analyse these differences, and to comprehend at which conditions one strategy may be preferable to another, we considered the \( PET \) functions of the two best behaving strategies, i.e. PC and EC, and expressed the breaking point inequality as:

\[
PET(\tau, PC, u) \geq PET(\tau, EC, u)
\]

For the definition (1) of \( PET(\tau, s, u) \):

\[
\tau \frac{\min RT_u}{RT_u(PC)} + (1 - \tau) \frac{\min E_u}{E_u(PC)} \geq \tau \frac{\min RT_u}{RT_u(EC)} + (1 - \tau) \frac{\min E_u}{E_u(EC)}
\]

For the definition of \( \min RT_u \) and \( \min E_u \), and for the results obtained through simulation, we have:

\[
\min RT_u = \min \{RT_u(s), s \in S\} = RT_u(PC) \quad \text{and} \quad \min E_u = \min \{E_u(s), s \in S\} = E_u(EC).
\]

Therefore by substituting:

\[
\tau + (1 - \tau) \frac{E_u(EC)}{E_u(PC)} \geq \tau \frac{RT_u(PC)}{RT_u(EC)} + (1 - \tau)
\]

\[
\tau \left(1 - \frac{E_u(EC)}{E_u(PC)}\right) + \tau \left(1 - \frac{RT_u(PC)}{RT_u(EC)}\right) \geq 1 - \frac{E_u(EC)}{E_u(PC)}
\]

Finally by substituting with (3) and (4) (slightly modified to consider utilization \( u \)):

\[
\tau ESL_u(PC) + \tau PIL_u(EC) \geq ESL_u(PC)
\]

This leads to the synthetic expression:

\[
\tau \geq \frac{ESL_u(PC)}{ESL_u(PC) + PIL_u(EC)}
\]

Equation (6) expresses the range of values \([\tau, 1]\), at utilization \( u \), for which strategy PC may be preferable to EC (i.e. its mix of performance and energy savings is higher than EC). From (6) we obtain the two breaking point values for utilization at 20% and 100%, for the current scenario.

\[
\tau_{20} = \frac{ESL_{20}(PC)}{ESL_{20}(PC) + PIL_{20}(EC)} = \frac{5.02}{5.02 + 8.51} = 0.37
\]

\[
\tau_{100} = \frac{5.84}{5.84 + 2.6} = 0.68
\]
We note that at 20% the performance loss between EC and PC ($PIL_{20}$) is 8.51%, while the energy gap ($ESL_{20}$) between PC and EC is 5.02%. Therefore, giving up performance improvement makes sense only for a considerable commitment towards energy reduction that is reasonable at low utilization factors. This somehow reflects on the quite low value of the $\tau_{20}$ breaking point. We can read $\tau_{20}$ as the need to have great motivation (at least a 63%) to prefer energy savings over performance improvement. At 100% utilization we have a performance gap ($PIL_{100}$) of 2.6% against a 5.84% energy gap between PC and EC ($ESL_{100}$). In this case the highest ESL gap of PC may be seen as the superiority of EC – i.e. its advantages in energy saving are greater than its performance deficits. Thus a 0.68 breaking point value, for $\tau_{100}$, may be read as quite a strong motivation (at least 68%) in order to select PC instead EC. To conclude the analysis of this scenario we observe that when a system is almost underutilized EC seems to be an acceptable alternative to PC to drive the allocation process. While at maximum loads PC slightly outperforms EC and could be considered a preferable choice if performance is the first goal.

![Figure 10](image)

**Figure 10** Performance/Energy Trade-off function at 100% utilization, by varying the trade-off parameter $\tau$.

These observations suggest the importance of the careful calibration of all aspects (e.g. not only the relative relevance between performance and energy, but also the kind and intensity of workloads) that affect the behaviour of a particular system, when choosing a scheduling strategy. It is worthwhile to note that as the PET function strictly depends on the particular configuration of the systems as a whole (i.e. the measurements related to the testbed at hand), the above mentioned results are to be intended as purely indicative, and strongly linked to the scenario examined.

### 6.2. Performance Oriented Scenario

In this scenario we replaced a slower node, simulating the Paperoga resource, with a new node with characteristics derived by the more performing (but energy greedy) Michelangelo cluster (see Figure 4 and Table 2). Table 6 resumes the observational values of Michelangelo, used to feed simulation (for reader convenience the values for IBM and Cluster1 are reported again). As a consequence of this resource swapping we can reasonably expect a decrease in overall system response times from which an increase in total power consumption may result. These expectations have been confirmed by the simulations carried out and documented by Figure 11 and Figure 12 which compare the results obtained by EC and PC strategies, in the two alternative scenarios. For the sake of clarity, measurements at 100% utilization were not reported.
Table 6 Observational parameters which feed simulated allocation policies.

<table>
<thead>
<tr>
<th></th>
<th>PC</th>
<th>LCR</th>
<th>MFW</th>
<th>EC</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$p_{ISO}$</td>
<td>$p_{HPL}$</td>
<td>$e_{ciISO}$</td>
<td>$e_{ciHPL}$</td>
</tr>
<tr>
<td>IBM</td>
<td>1 s</td>
<td>13 s</td>
<td>1900 W</td>
<td>10.69 MFLOPS/W</td>
</tr>
<tr>
<td>Michelangelo</td>
<td>1.2 s</td>
<td>10.7 s</td>
<td>2800 W</td>
<td>7.13 MFLOPS/W</td>
</tr>
<tr>
<td>Cluster1</td>
<td>3.4 s</td>
<td>23.1 s</td>
<td>2200 W</td>
<td>3.48 MFLOPS/W</td>
</tr>
</tbody>
</table>

In the following we use the I-M-C (I-P-C) as a subscript to denote a strategy operating in the context of the present (I-M-C stands for IBM, Michelangelo and Cluster1) or the previous scenario (I-P-C stands for IBM, Paperoga and Cluster1).

![Graph showing comparison of PC and EC response times according to the two simulated scenarios.](image)

**Figure 11** Comparison of PC and EC response times according to the two simulated scenarios.

Please note that for the sake of readability, given the superiority of Performance Congruent and Energy Congruent both on RT and E, compared to other strategies, these last are not reported in Figure 11 and Figure 12. However, the behaviour of MFW and LCR, according to the present scenario, can be easily deduced by the PIL and ESL graphs, depicted in Figure 13 and Figure 14, which show the relative increase in response time and power consumption of both policies with respect to PC and EC respectively.
It is quite interesting to note how the modified conditions affected the outcomes of each strategy and their relative comparison. If Figure 13 confirms the superiority of EC as compared to LCR and MFW as already observed in the previous scenario, then Figure 14 highlights that PC obtained better energy results as compared with respect to the other two strategies. For example, if we compare PC and LCR, we see that the latter always “prefers” Cluster1 while PC chooses Michelangelo, when IBM is busy. Therefore, with PC each job is assigned to a node with 27% more power consumption, but which always takes less than 50% of the time to execute, thus allowing PC to achieve a better energy outcome.

We may conclude that when using the congruent approach, the ability of Michelangelo to quickly execute the ‘right’ jobs it has assigned, (may) compensate for its greater consumption, more than that of Paperoga. Once again this fact confirms the importance of adapting the policy selection to the real operative scenarios.
From the previous observation, it seems reasonable to further analyze the two scenarios. A quick glimpse at Figure 11 and Figure 12 suggests that while the gaps in response times seem to follow a quasi-constant trend at increasing utilization – when considering energy consumption, the gaps between the two scenarios tend to diminish as utilization rises. To better analyse these relative variations we depicted the relative increase in $RT$ and $E$, of $PC_{(I-P-C)}$ and $EC_{(I-P-C)}$ with respect $PC_{(I-M-C)}$ and $EC_{(I-M-C)}$ and vice-versa, in Figures 15 and 16.

From a more accurate analysis of Figure 15, we can observe that Performance Congruent strategy tends to worsen, at increasing workload, when the slower Paperoga cluster is operated. In this case RTs are at least 65% higher than the I-M-C scenario. Energy Congruent strategy showed a quite similar trend. These dramatic differences between the two scenarios, however, should not
be surprising if we note that Michelangelo employs less than one half the time to execute a linear algebra job (25 secs against 55) and less than one fourth for an isoextraction job as compared to Paperoga (3 secs against 13). The previous observation is overemphasized by the reduced size of the model considered: for a network with only three nodes, the influence of each single node becomes particularly relevant.

![Figure 16](image_url) Relative increase in $E$ of $PC_{(I,M,C)}$ and $EC_{(I,M,C)}$ as compared with $PC_{(I,P,C)}$ and $EC_{(I,P,C)}$.

When looking at the difference in energy consumption we can observe the constant reduction between the gaps of the two scenarios from Figure 16. These gaps range from a maximum of 60% (for $PC_{(I,M,C)}$ at 20% utilization) to a minimum of around 10% for both strategies (at 80% utilization). This (relative) reduction is due to the fact that at increasing workloads, Michelangelo is able to process more jobs (in unit time) than Paperoga does, but still maintains a low residence time. Table 7 presents the residence times, throughputs and the energy consumption, for Paperoga and Michelangelo, resulting from the adoption of the Performance Congruent strategy in the two scenarios. It is easy to see that when utilization rises the relative energy gaps diminish. A throughput increase of about 247% and 340% for Michelangelo and Paperoga corresponds to a residence time increase of 46% and 350% respectively is seen from Table 7. These numbers resulted in a 1600% increase in energy consumption by Paperoga compared to (only) a 360% increase by Michelangelo - thus explaining the trend shown in Figure 16. In other words, if the energy advantage of a low consumption machine, like Paperoga, is evident at low system utilization, situation changes at higher utilization factors. Indeed, when the workload increases, the disadvantage of being less powerful penalizes Paperoga's energy behaviour due to its higher service times. The opposite is true for a more powerful but more energy consuming machine such as Michelangelo.

**Table 7** Residence times, Throughput and Energy consumption of the two resources in the two simulated scenarios.

<table>
<thead>
<tr>
<th></th>
<th>Residence times (sec)</th>
<th>Throughput</th>
<th>Energy consumption (joules)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20%</td>
<td>40%</td>
<td>60%</td>
</tr>
<tr>
<td>Paperoga</td>
<td>6.55</td>
<td>8.01</td>
<td>10.37</td>
</tr>
<tr>
<td>Michelangelo</td>
<td>3.77</td>
<td>3.86</td>
<td>4.60</td>
</tr>
</tbody>
</table>
The last observation seems to be confirmed by analysing the PET graphs, at 20% and 100% utilization, as reported in Figure 17 and Figure 18. In particular, by examining Figure 17, we can note the relevant trade-off between performance and energy showed by PC, which is characterized by the greatest slope amongst all strategies. This trade-off is noticeably reduced at highest load, as shown by Figure 18, resulting from the good energetic behaviour of PC which closely approaches PC to EC (i.e. with $e_{100}(PC) = 0.98$).

![Figure 17 Performance/Energy Trade-off function at 20% utilization, by varying the trade-off parameter $\tau$.](image)

By looking at the breaking points $\tau_{20}$ and $\tau_{100}$, for this scenario, we have the following values:

$$\tau_{20} = \frac{ESL_{20}(PC)}{ESL_{20}(PC) + PIL_{20}(EC)} = \frac{17.92}{17.92 + 10.51} = 0.63$$

$$\tau_{100} = \frac{2}{2 + 9.15} = 0.18$$

At 20% utilization we have a performance gap ($PIL_{20}$) of 10.51% against a 17.92% energy gap between PC and EC ($ESL_{20}$). In this case the highest $ESL$ gap may be seen as the superiority of EC – i.e. its advantages in energy savings are greater than its performance deficits (as compared to PC). Thus a 0.63 breaking point value, for $\tau_{20}$, may be read as a 63% greater motivation to select PC instead EC. Conversely, as to $\tau_{100}$, we note that at 100% the performance loss between EC and PC ($PIL_{100}$) is 9.15%, while the energy gap ($ESL_{100}$) between PC and EC is limited to 2%. Therefore, giving up consistent performance improvement, in favour of a minor (relative) gain in energy savings, makes sense only for a great commitment towards energy reduction. This reflects on the quite low value of the $\tau_{100}$ breaking point. We can read $\tau_{100}$ as the need to have a strong motivation (at least 82%) to pursue energy savings over performance improvement, which, as previously noted, is not usually the case at high loads. This also indicates that in the presence of a relative high performance node (like Michelangelo) a strategy such as PC may adequately respond either to performance and energy requirements – while at low loads EC may still be a satisfactory choice.
These last remarks, along with those concerning the IBM-Paperoga-Cluster1 scenario, show that the \(PET\) function strictly depends on several factors (e.g. system load, resources configuration, number of system nodes, etc.) and the information it provides must always be referred to the actual operative context.

7. Conclusions and future work

In complex environments like Grids it is of paramount importance to find the most appropriate application-resource matching. Not only a correct matching impacts on execution codes efficiency, but may also favour a more energy-aware usage of computational resources. The heterogeneity of Grid environments leads us to investigate the influence of resources and their real configuration on the improvements in application performance and on energy savings.

In this paper, we have presented a novel energy related metric, ECI, aimed at combining performance and energy characterizations of Grid resources. Through simulations, we have focused on the deployment of two allocation strategies, Performance Congruent and Energy Congruent, based respectively on a performance index and on ECI. We have compared these strategies in terms of efficiency and energy savings achieved with other scheduling policies (i.e. Round Robin, and energy aware strategies like MFW and LCR). Furthermore, in order to support the analysis of different scheduling policies, we have introduced the \(PET\) function. By studying two quite different simulated scenarios we have observed that PC and EC strategies generally achieved better results. In particular, from the performance perspective, PC outperformed all other strategies, immediately followed by EC. Instead, considering energy savings, the EC strategy outperformed the others in all scenarios, while PC showed a variable behaviour with respect to the examined scenarios, although always outperformed the energy-agnostic Round Robin. The choice of one of these two strategies depends, first of all, on the importance assigned to the performance and the energy objectives. As pointed out in the discussion about the \(PET\) function, such a choice is affected by a number of factors (e.g. workload variation, system configuration and capacity of resources) that may hinder the decision.

As a final remark we want to highlight the tagging proposal briefly introduced in Section 5.2. It is aimed at supporting the adoption of the resource characterization in Grid environments, thus fostering the deployment of our congruent allocation strategies.

The future direction of this work will address the study of power consumption during job migrations in Grid environments. As it is known, the migration of a job may happen due to a number of reasons, e.g. system load balance, fault recovery, QoS assurance, etc.; accordingly, different elements have to be considered to reschedule a job and, in our opinion, energy should be one of these parameters. Indeed, this topic has received poor attention from the scientific
community and few efforts have been spent so far to investigate the issue. Starting from the results presented in this paper, we plan to analyse how power consumption relates to QoS parameters, and investigate the implications of energy constraints on migration mechanisms.

References
EPA, Environmental Protection Agency, 2007. ENERGY STAR Program, Report to Congress on Server and Data Energy Efficiency. Available at:


GCPI, 2011. SiCortex Green Index. Available at: http://sicortex.blendinteractive.com/green_index (July 2011)

GLite - Lightweight Middleware for Grid Computing http://glite.cern.ch/ (September 2011)

The Globus Toolkit, www.globus.org/toolkit (September 2011)


Green500 list, 2011. Available at: http://www.green500.org/ (July 2011)


Hawkeye, A Monitoring and Management Tool for Distributed System: http://www.cs.wisc.edu/condor/hawkeye/


LITBIO, 2011. Michelangelo Hardware, LITBIO Project (http://www.supercomputing-it/At_Cilea/michelangelo_eng.htm - password required)


SuperMUC, supercomputer hosted at Leibniz Supercomputing Centre. Description available at: http://www.lrz.de/services/compute/supermuc/systemdescription/ (July 2011)


Vitae

Dr. Antonella Galizia started collaboration with the Institute for Applied Mathematics and Information Technology of the Italian National Research Council in June 2003. Her research interests are in the fields of parallel and distributed computing with a specific focus on scheduling policies for the dynamic allocations of the parallel jobs on the distributed resources on the basis of performance information. She is also involved in hydro-meteorological research exploiting HPC resource to improve spatio-temporal predictability of severe weather events.

Dr. Alfonso Quarati has been a Researcher at the Institute for Applied Mathematics and Information Technology (IMATI) of the Italian National Research Council (CNR) in Genoa since 1997. His current research interests cover models and tools for grid and cloud computing, QoS assessment and performance evaluation methodologies, mashup and web-services design, collaborative working technologies. He has co-authored more than 50 scientific papers, published in journals, book chapters, technical reports and conference proceedings.