On Achieving Energy Efficiency and Reducing CO_2 Footprint in Cloud Computing

[1]Usman Wajid, [2]Cinzia Cappiello, [2]Pierluigi Plebani, [2]Barbara Pernici, [1]Nikolay Mehandjiev [2]Monica Vitali, [3]Michael Gienger, [4]Kostas Kavoussanakis, [5]David Margery, [6]David Garcia Perez, [1]Pedro Sampaio,
[1]University of Manchester (UK), [2]Politecnico di Milano (Italy), [3]Universität Stuttgart (Germany), [4]The University of Edinburgh (UK), [5]INRIA (France), [6]ATOS (Spain)

Abstract— With the increasing popularity of the cloud computing model and rapid proliferation of cloud infrastructures there are increasing concerns about energy consumption and consequent impact of cloud computing as a contributor to global CO_2 emissions. To date, little is known about how to incorporate energy consumption and CO_2 concerns into cloud application development and deployment decision models. In this respect, this paper describes an eco-aware approach that relies on the definition, monitoring and utilization of energy and CO_2 metrics combined with the use of innovative application scheduling and runtime adaptation techniques to optimize energy consumption and CO_2 footprint of cloud applications as well as the underlying infrastructure. The eco-aware approach involves measuring or quantifying the energy consumption and CO_2 at different levels of cloud computing, using that information to create scheduling and adaptation techniques that contribute towards reducing the energy consumption and CO_2 emissions, and finally testing and validating the developed solutions in a multi-site cloud environment with the help of challenging case study applications. The experimental and validation results show the potential of the eco-aware approach to significantly reduce the CO_2 footprint and consequent environmental impact of cloud applications.

Keywords: Energy-aware systems, Scheduling and task partitioning, Evaluation

1 INTRODUCTION

Cloud computing offers an approach for running applications in cost effective and highly scalable manner. However, despite the several business and technical advantages of the cloud-computing model, there are increasing concerns about the rising costs (resulting from considerable energy consumption) and the significant impact of cloud computing as a contributor to global CO_2 emissions [25].

In Europe energy consumption and consequent environmental implications are particularly attracting attention as ICT related CO_2 emissions approach 4% of EU's total CO_2 emissions [2]. The consideration for energy efficiency and CO_2 awareness becomes paramount in the wake of increasing demand for cloud resources resulting in exponential growth of cloud facilities.In this respect, this is not just financial but also a political, regulatory and ethical issue, which raises significant questions about the overall environmental sustainability of the cloud computing model as its uptake expands. The environmental implications of vast energy consumption are likely to result in CO_2 constraints being imposed by regulators and environmental agencies on cloud providers (such as [4]). Moreover, the environmental awareness and the pressure by regulatory authorities can influence consumers' selection criteria for cloud sourcing. Thus, it is no longer sufficient to optimize cloud sourcing models based solely on functional features, quality of services and low costs without considering their energy consumption and overall CO_2 impact.

With this backdrop, there is an ever increasing need for innovative solutions to address the issues concerning energy consumption and CO_2 emissions at different levels of cloud computing. For instance, the transparency in measuring energy consumption and CO_2 footprint can enable users to make informed choices while souring cloud services. Some other areas where such issues can be addressed include energy and CO_2 awareness at application design and deployment phases, development of energy and CO_2 aware cloud monitoring mechanisms and more transparency of the energy mix that goes into powering the cloud infrastructure.

In this paper we provide a timely, challenging and highly innovative approach to energy efficient and CO_2 aware cloud sourcing. The proposed approach brings togethers the following contributions:

- Quantification of energy consumption and environmental impact at different levels of cloud computing e.g. application, virtual machine and infrastructure level.

- Eco-aware scheduling of applications on cloud
- Eco-aware management of running cloud applications

With these key contributions, some complementary topics covered in the paper include consideration for energy and CO_2 aspects in application design, eco-aware monitoring in the cloud and finally validating the feasibility of energy and CO_2 aware solutions in the actual cloud environment.

By addressing some of the key issues in cloud sourcing our holistic eco-aware approach aims to bridge the gap between the availability of energy consumption data linked to cloud computing platforms and the capability to leverage this data in the formulation of an eco-aware cloud application deployment and management solution. The rest of the paper is structured as follows. Section 2 provides an overview of existing work in the area of energy efficient and CO_2 aware cloud computing. Section 3 introduces the details of our proposed approach and the different solutions developed to address some key issues. Section 4 provides details concerning the experimental validation of the eco-aware approach. Section 5 presents the results of experimental validation of our approach. Finally, we conclude the paper in Section 6 with a discussion about the results, their impact and setting out directions for future work in this area.

2 BACKGROUND ON ENERGY EFFICIENT AND CO_2 AWARE CLOUD COMPUTING

Existing research efforts in the area of cloud computing focuses on security, interoperability and other SLA related issues, but generally overlooks energy consumption and resulting environmental implications as relevant context information for workload deployment and management on the cloud. Early research on energy efficiency originated from grid and datacenter environments. For example, the use of game theoretical methods were explored to optimize energy consumption with system performance in a grid [27], and the use of data migration was explored in [31] where data migration techniques move data from one device to another in order to reduce the number of active storage or memory resources, and consequently the energy consumption. However, the above approaches are more suitable for datacenter environment as they address the need for reducing energy consumption of underlying infrastructure, without any emphasis on energy consumption and/or environmental impact of running applications.

To consider the application perspective some research efforts focus on the estimation of the energy consumption of applications based on their architecture. For example, [30] examine software methodologies and development tools that can be employed to enhance energy efficiency of application in mobile systems. However, while highlighting contributing factors for creating energy-aware applications such as computational efficiency and context awareness energy efficiency and CO_2 reduction of applications in cloud environment are not explored or qualified. To this end, the Greenhouse Gas (GHG) protocol provides calculation tools and guidelines for quantifying and managing emissions in the cloud platforms. The GHG guidelines [35] can provide abstract grounding for developing a metric aimed at quantifying and tackling energy consumption and CO_2 emissions in the cloud. Such metrics can support approaches that aims to reduce the energy and CO_2 in cloud computing.

Similar to the GHG protocol, various energy benchmarks are proposed as ecological parameters or eco-metrics. These eco-metrics are typically seen as an extension to the general concept of Key Performance Indicators (KPIs). For example, [22] defines eco-metrics also termed as Green Performance Indicators (GPIs) and discuss how monitoring of service applications from an energy perspective can reveal peaks and leakages of energy particularly in high performance computing (HPC) environments. However, the scope of the metrics in [22] is beyond typical cloud computing environments as it considers elements mostly relevant in the HPC.

Furthermore, the Greenpeace report [25] highlights several factors that can contribute towards environmental impact of cloud computing e.g. Carbon Utilization Effectiveness (CUE) is one of the parameters that is starting to be considered by some of the public cloud providers. Another contribution of the Greenpeace report is the clean energy index, which considers the energy mix of the utilized energy as well as siting and reuse of energy in cloud infrastructure. However, while these metrics serve as important reference; there is no standard or widely accepted metric set allowing for easy measuring and monitoring of energy consumption in cloud computing e.g. some other efforts propose different ways of monitoring and representing energy and other QoS parameters [28], [29]. While the above mentioned work focuses on the IT infrastructure, there is no clear indication on how energy efficiency and CO_2 can be evaluated at the level of application already deployed or running on the cloud.

In other related work, a comprehensive survey of the existing energy efficient approaches in the cloud environment is presented in [23]. The survey discusses dynamic power management approaches, both at the hardware and at the software level. At software level, the approaches range from management of workloads at the level of

the single server, to overall workload management at cloud site level. However, most of the surveyed approaches either require specific hardware support or insight into the application execution behavior, both of these are difficult to expect in typical cloud sourcing scenarios. For more general workloads [18] shows that multi-factor optimization can balance efficiency with quality of service and significantly outperform the typical levels of underprovisioning and drop in performance of highly consolidated VMs (virtual machines) caused by shared resources. A routing framework in [12] aims to dynamically control user traffic to different datacenters based on their carbon footprint. The framework also allows the three way trade-off between access latency, carbon footprint and electricity cost to determine optimal datacenter upgrade plan in response to increases in traffic. However, the framework is more suitable for infrastructure providers who own the hosted applications, for others the framework is only suitable for monitoring of application resource utilization. Further, [19] explores the effect of the inclusion of CO_2 intensity in a simulated multi-cloud scenario. However, the simulated scenario may not consider the unexpected behavior of an actual cloud infrastructure e.g. influence of other running applications on the shared resources.

Another strand of related work focuses on allocating virtual machines with the primary goal of improving energy efficiency in cloud infrastructure and also to reduce the environmental impact of cloud applications [11], [20], [22], [24]. One such approach [7] evaluates the performance-per-watt achieved by a range of VM allocation heuristics and shows that the best available algorithms offer substantial reductions in energy usage. In common with many such approaches, these results do not translate energy consumption into environmental impact, were obtained from simulation rather than experimentation and do not address the need for post deployment management of applications.

In terms of industrial drive for energy efficiency and CO_2 reductions, in the UK the Carbon Reduction Commitment Energy Efficiency Scheme [4] requires organisations to pay a charge for carbon emissions related to electricity usage. In US Google announced its adoption of the ISO 50001 standard for energy management systems [5]. Unlike the widely adopted ISO 9001 (quality) and ISO 14001 (environmental performance) standards, ISO 50001 actually requires continual improvement in energy performance itself, rather than merely of the management system itself. E.g., Google maintained a long downward trend by reporting a fleet average PUE of 1.11 compared to 1.13 a year ago [6]. Here, the developments in alternative fuel sources (e.g. fuel cell generation) can help organizations to achieve more in terms of reducing CO_2 emissions by diverting or distributing traffic or workload to data centers that use eco-friendly energy sources [13]. Commercial cloud offering e.g. Amazon and GreenQloud promote the use of green or renewable energy with little or no emphasis on quantification of energy consumption at VM level or the use of application deployment and management techniques that can contribute towards energy efficiency in cloud computing.

In terms of current research projects, while there are a number of new projects in the area of energy-efficient computing, most appear to be concentrating on HPC rather than cloud computing. For example, EXCESS (Execution Models for Energy-Efficient Computing Systems) is an EU FP7 project aiming to provide new, holistic models for energy-efficient computing in future HPC systems [8]. Similarly, Exa2Green [9] is a multi-disciplinary project under EU FP7 programme that aims to develop the advanced energy monitoring, profiling and optimization techniques needed to enable sustainable exa-scale HPC. In the US, Argo [10] will, amongst other objectives, incorporate energy management at the OS level of future supercomputers. Meanwhile ADEPT [14] is investigating the effect of parallel software on energy efficiency in HPC. Another noteworthy trend in HPC systems is the increasing domination of the Green500 list [15] of energy efficient supercomputers by GPU/accelerator based systems. As an increasing number of codes are becoming able to take advantage of such accelerators, many Cloud providers are now offering GPUs, e.g. Amazon EC2 GPU Instances [16], and GPU vendors are providing new features targeting the virtualization market [17].

In the work presented in this paper we complement many existing efforts e.g. definition of metrics for quantifying energy consumption and environmental impact of cloud computing as well as eco-aware scheduling of VMs to achieve best CO_2 and performance ratios for cloud applications. We then introduce runtime adaptation of cloud applications and validate our solutions in real world scenarios by experimenting in an actual federated cloud infrastructure using a set of challenging case study applications.

3 A HOLISTIC APPROACH FOR ENERGY EFFICIENT AND CO_2 AWARE CLOUD COMPUTING

This paper adopts a more CO_2 focused approach to bring about key advances in cloud computing and the way applications are designed and deployed on the cloud. Our eco-approach involves three key phases namely Meas-

ure, Create and Test. The *Measure* phase focuses on quantification of energy consumption and environmental impact of cloud computing, the *Create* phase develops techniques and software artefacts to reduce energy consumption and CO_2 emissions of cloud applications, and the *Test* phase tests the outcome of previous two phases on an existing multi-site federated cloud facility known as BonFIRE (www.bonfire-project.eu).

BonFIRE platform came up as a suitable choice for implementation and testing of our approach as it allowed us to augment the existing infrastructure with Power Distribution Units at different cloud sites, thus enableing us to measure the electricity consumption at different levels of the cloud infrastructure. BonFIRE also allowed us to work on the cloud software stack in order to update the functionality of BonFIRE services, e.g. BonFIRE monitoring mechanism was extended to also expose the eco-metrics information. In this respect, the control and flexibility offered by BonFIRE, to support the implementation and testing of our approach cannot be expected from a commercial cloud platform.

The three phased approach addresses important research questions concerning quantification of environmental impact of cloud computing, enacting deployment and runtime adaptation actions that can decrease the energy consumption and CO_2 footprint of cloud applications as well as considering environmental implications in the design and subsequent execution of cloud applications.

Figure 1 provides an overview of implementation and execution aspects of our approach. The first step is triggered by an application deployment request. A deployment request typically entails the requirements for certain number of virtual machines with certain characteristics or the combination of CPU, memory and storage capacity. By providing such information users (as designers, developers or owners of cloud applications) can scale the resources in a VM to suit the requirements of their workloads. Resource requirements can be an estimate (based on the nature of workload) or drawn from scientific models e.g. the model in [38] can provide decision support for the deployment of workload on suitable resources by calculating the instantaneous power consumption of the workload. In our approach, the deployment of virtual machines is determined by an eco-aware Scheduler, a key decision making component in the cloud infrastructure.

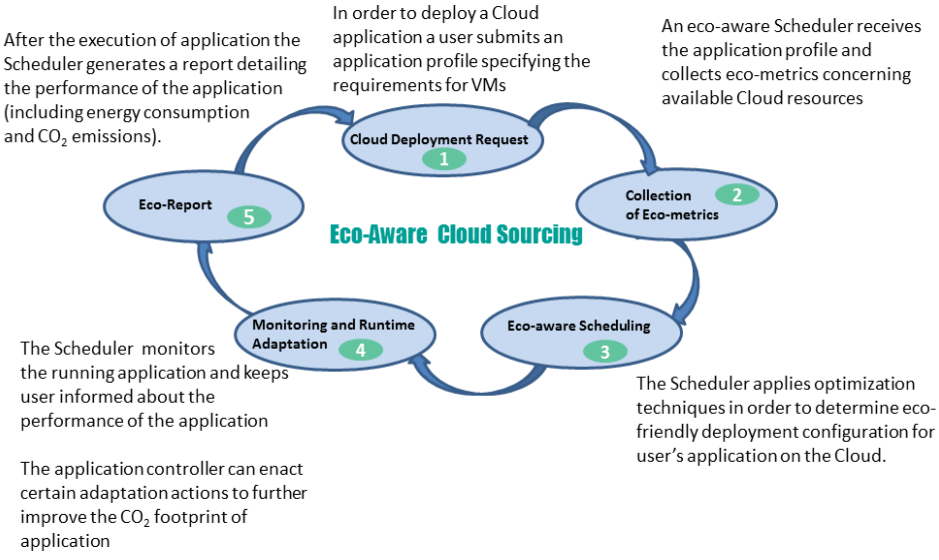

Figure 1: Implementation of the proposed approach

In step 2, the Scheduler collects eco-metrics from different levels of the cloud infrastructure. Eco-metrics is a term used here to refer to parameters that provide information concerning energy consumption and CO_2 footprint at different levels of the cloud infrastructure – further discussed in Section 3.1. The information provided by eco-metrics is used in the eco-aware scheduling of applications on the federated cloud resources, as shown in step 3 – further discussed in Section 3.2. Once deployed the (VMs hosting the) applications are constantly monitored together with the underlying infrastructure e.g. considering spikes in resource utilization, deployment of new applications and termination of already running applications to evaluate opportunities for further improvements in energy consumption and CO_2 footprint of running applications. The monitoring information is taken into account

when deciding and triggering certain adaptation actions that can reduce the energy consumption and CO_2 footprint of running applications – Step 4 is discussed in Section 3.3. After the application execution, the detailed monitoring information is compiled in an eco-report.

The eco-report informs users (e.g. application designer and developers) about the total energy consumption and CO_2 footprint of their applications. The link between step 5 and step 1 in Figure 1 reveals the reinforcement learning model of our approach where the user is encouraged to use the information from eco-report to improve the design and execution aspects of their applications in a bid to make them more energy efficient and eco-friendly.

3.1. Quantification of Energy Consumption and Environmental Impact

In order to achieve energy efficiency in cloud infrastructure and reduce the CO_2 footprint of cloud applications, it is necessary to assess the energy efficiency and environmental impact of the overall system. This requires that a set of key metrics is established that expose energy consumption at various levels in the cloud infrastructure and enable quantification of the ecological impact of cloud sourcing. While the metrics developed in previous work (such as [22]) can be adopted as a basis, these focus mainly on the infrastructure. Ideally the metrics should reflect the energy efficiency from a holistic perspective and should allow the derivation of the interrelation between different components of cloud computing. Therefore, we have developed a layered approach for the definition of usage based and power/energy based metrics (named eco-metrics) for each component as follows:

- Application layer
- VM layer
- Infrastructure layer

The layered structure of our eco-metrics is designed to unravel information from different levels of the cloud infrastructure, making it suitable for use in single site or multi-site cloud platforms, as shown in Figure 1. Our eco-metrics captures not only aspects about the environmental impact, but also the performances of the system. This allows our solution to mediate between improvements with respect the environmental sustainability and the performances. It is important to note that the set of metrics used in our work include a mix of metrics from existing literature (e.g. [32]) that we have implemented in our monitoring framework, customized metrics for the analyzed context and newly defined metrics such as VM energy consumption, application PUE, application energy productivity and application green efficiency. The newly defined metrics complement the existing and customized metrics to provide a complete view of the environmental impact of the cloud applications.

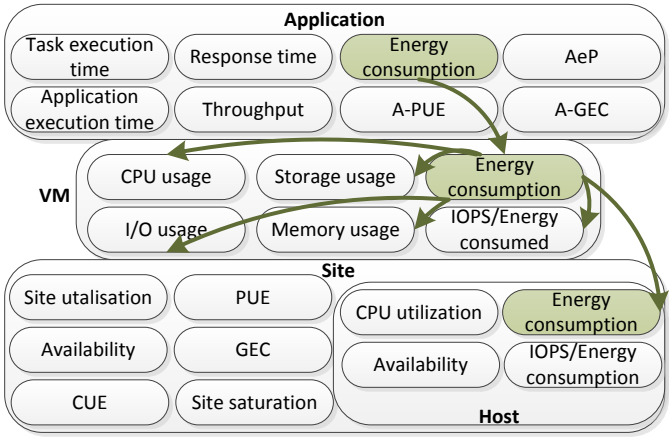

Figure 2: Layered set of eco-metrics

3.1.1. Infrastructure layer

The infrastructure level considers that the VMs are deployed on physical hosts and the hosts reside at a specific cloud site. More specifically, the infrastructure layer includes metrics that measure the behavior of the cloud resources ranging from the single physical machine to the entire cloud site. Especially for the energy or power metrics, this layer is fundamental as it is the place where we can measure the power consumption of physical hosts using PDU (Power Distribution Units). In our experimentation we augmented the existing federated cloud infrastructure by attaching PDUs with the physical hosts of the three cloud sites. Also, since the cloud sites (used in

our experimentation) were located in three different countries (UK, France and Germany) the different energy mix consumed by each site helped in determining the CO_2 footprint of the cloud site along with the GEC (Green Energy Efficiency) [32] i.e., the percentage of the power coming from renewable source. In addition to the site and host metrics, each site exposes the energy mix metrics, generally available from their energy grid. Table 1describes the metrics concerning the energy mix, the site metrics are presented in Table 2, the host metrics are presented in Table 3

Table 1: Infrastructure energy metrics

Metric	Definition
Energy Mix (%)	This metric measures the amount of different types of energy mix provided by the power grid such as: Biomass, CCGT (Combined Cycle Gas Turbine), Coal, Cogeneration (of heat and power), Fossil, Gas, Geothermal, Hydraulic, NPS hydro, Nuclear, OCGT (Open Cycle Gas Turbine), Oil, Other, Solar, Total green, Water and Wind generated energy.
Produced CO_2 (g/kWh)	This metric calculates the site emission factor revealing the amount of CO_2 generated by 1 kWh. It results from considering weighted sum of the percentage of usage of the different sources multiplied by their own emission factors.
Grid total (MW)	Grid total measures the total amount of produced power by the power provider

Table 2: Infrastructure layer site metrics

Metric	Definition
Site utilization (%)	Current utilization of a single site. The metric is defined as the ratio of available free cores of all worker nodes; over the total number of cores of all worker nodes.
Storage utilization (%)	The percentage of the frontend storage usage gives a good representation of the storage site usage (as the frontend hosts all permanently stored images). If necessary, this metric can be implemented for single worker hosts as well.
Availability (Boolean)	A site will be available if the Cloud API server provides a reasonable and well-defined answer to a request. Furthermore, at least one worker host has to be available (see definition in Table *3*).
Power Usage Effectiveness - PUE	The Power Usage Effectiveness metric is used to determine the energy efficiency of a site. It is measured as the ratio of power used by computing equipment over total facility power

Table 3: Infrastructure layer host metrics

Metric	Definition
Power consumption (W)	The power consumed by a specific host at a given time.
Disk IOPS (OPS/s)	This metric checks the rate of input / output operations of the disk within a physical host. This metric includes the operations of all virtual instances running on the host and those of the underlying operating system.
CPU utilization (%)	The utilization of the processor(s) inside a physical host. For each host this metric indicates how much processor capacity is used for the underlying operating system and all the virtual instances at a given time.
Availability (Boolean)	A host is available if it is seen as “online” within the cloud manager.

3.1.2. VM layer

The VM layer focuses on the metrics able to evaluate the performance and energy consumption of the VMs

running on the physical host (see Table 4). At this layer, most of the metrics are derived from the metrics measured at the infrastructure layer. For example, to calculate the environmental impact the energy consumed to run a VM is computed by considering the amount of resources (CPU, memory etc) used by the VM in order to subdivide the energy consumed by the physical host among other VMs running on it. This is similar to inferring energy from resource usage in [39]; however our approach is better tuned for the actual cloud scenario where more than one VM is expected to run on one physical host, whereas the direct measurement of VM energy consumption in [39] cannot be used to derive the overall energy consumed by the underlying host. Further details of the VM metrics can be gathered from [40].

Table 4: VM layer metrics

Metric	Definition
CPU usage (%)	Current processor utilization rate measured / as seen by the running virtual machine.
Storage usage (%)	$$S_{usage} = \frac{S_{used}}{S_{total}} * 100$$ Storage utilization (S_{usage}) on the corresponding virtual storage device is defined as the ratio of the used disk space S_{used} to the total amount of disk space S_{total}
Power consumption (%)	The power consumed by the analyzed VM at a given time. Our virtual machine power consumption formula defines the consumed power per virtual machine P_{VM} as the sum of the VM contribution to keep the host running plus the actual VM consumption, as follows: $$P_{VM} = \frac{P_{HostIdle}}{\#VM} + (P_{Host} - P_{HostIdle}) * CPU\%_{VM}$$ The first term is calculated by dividing: the idle power of the physical host ($P_{HostIdle}$); by the number of running VMs ($\#VM$). For the actual VM consumption term, the net physical host power consumption ($P_{Host} - P_{HostIdle}$) is calculated and multiplied by the CPU utilization of the VM as seen from the physical host ($CPU\%_{VM}$). In order to assess this term, the CPU utilization of the VM as seen by the physical host is calculated by dividing: the average CPU usage of the analyzed VM in the last measurement interval (ΔCPU_{VM}); by the sum of the average CPU usage of each VM in the same measurement interval. $$CPU\%_{VM} = \frac{\Delta CPU_{VM}}{\sum_i \Delta CPU_{VM_i}}$$

3.1.3. Application layer

At the top-most layer, there are the metrics that evaluate the resource consumption of applications running on the VMs. Assuming that an application can be distributed among several VMs and these VMs can be hosted on different sites, the metrics included at this layer, compute the environmental impact as well as the performances starting from the metrics at the lower levels. For instance, the energy consumption of the application is obtained considering the execution time and energy consumed by all the VMs used by the application.

In this layer we have newly defined a metric A-PUE (Application PUE) that is inspired by the classical PUE. As the latter is defined as the quantity of power used by the site divided by the power used by the IT devices, the A-PUE is defined as the quantity of power used by all the VMs involved in the application divided by the power used for running the application. Here the numerator counts the amount of power really consumed by the application. While the denominator takes into account several other processes that run in the VM alongside the application e.g., operating system processes. Assuming that the power consumed by each process is proportional to its CPU usage, the power P_{ijk} = Pik * CU_{ijk}, where CU_{ijk} is the percentage of CPU used by the process *j* associated to the application *i* running on the VM *k*. In this respect, the goal of A-PUE is to make the user aware of this fraction of power that is not consumed specifically for the application. Table 5 shows the key eco-aware application layer

metrics that are adopted and customized [22] [25] and defined by us [34].

Table 5: Application layer eco-metrics

Metric	Definition
Power consumption (W)	The power currently consumed by the application. This metric is derived by aggregating the power consumption of the VMs hosting the application.
A-PUE (Application PUE)	The application power usage effectiveness (PUE) is the ratio between the total amount of power (*P)* required by all VMs of an application *i* and the power used to execute the j-th application task: $$A - PUE_i = \frac{\sum_k P_{ik}}{\sum_{jk} P_{ijk}}$$
Application Energy Productivity (Transaction/Wh)	The Application Energy Productivity is the ratio between the number of executions of the tasks hosted by all VMs of application *i* and the total energy for the execution of the VM: $$A - EP_i = \frac{NTrans_{i\Delta t}}{\sum_k (P_{ijk} * \Delta t)}$$
Application Green Efficiency – AGE (W)	$$A - GE_i = \sum_k GEC * (P_{ik} * \Delta t)$$ The Application Green Efficiency metric measures how much green energy is used to run application *i*. We multiply the power consumed by all VMs of application *i* by the percentage of used green energy.

3.1.4. Implementation and collection of eco-metrics

The definition of layered eco-metrics was the basis for the design of a monitoring infrastructure to extract relevant data and its implementation in the BonFIRE cloud infrastructure, as shown in Figure 3. The monitoring infrastructure collects, calculates and stores monitoring data. It was implemented by customizing the Zabbix[1] monitoring API used in BonFIRE. All BonFIRE Cloud sites were instrumented with PDUs to expose the required metrics.

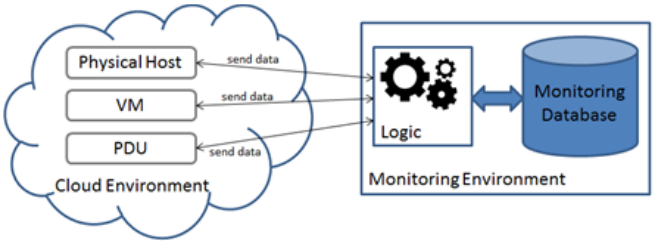

Figure 3: Monitoring infrastructure for eco-metrics

As shown in Figure 3, a monitoring infrastructure collects eco-metric data from the BonFIRE cloud sites, and sends it to a persistent monitoring database (DB) from where it is made available on-demand to other solutions using a dedicated web service.

3.2. Eco-Aware Scheduling of Applications

The availability of the monitoring data from the layered set of eco-metrics allowed us to quantify energy consumption and CO_2 emissions at different levels of cloud infrastructure. The monitoring data also provides use-

[11] www.zabbix.com

ful information about the availability and performance aspects of cloud resources. Having access to such information from different levels of cloud infrastructure enabled us to devise a set of application deployment strategies that take into account not only the resource level parameters of cloud resources but also their energy and CO_2 profiles. Our deployment strategies can be characterized as eco-aware heuristics that perform decision making at two different levels (i.e. at cloud site and physical host level) to work with the single site as well as within federated cloud infrastructure. The strategies are implemented (as Phase 3 in Figure 1) by an eco-aware Scheduler that represents the central deployment decision making entity in the cloud infrastructure.

Cloud Site Selection: In line with the underlying multi-site cloud infrastructure the first scheduling step is to select a suitable cloud site. This step is realized by switching our eco-aware Scheduler into one of the following modes:

Individual Deployment (IDM) performs the selection of a site (in a cloud federation) for each individual VM in the deployment request. In this mode, the VMs in a single deployment request (representing a distributed application) may be deployed on different sites. For example, after the allocation of a single VM, the suitability of a site may change and thus, the next VM in the same deployment request may be allocated to another site that fits the suitability criteria. The pseudo-code of this strategy is:

Pseudo-code: Cloud Site Selection

1. Get VM resource requirements
2. IF a site does not meet requirement THEN
   Remove site from the next steps
3. Apply load balancing function on each site
4. IF site does not pass load-balancing test THEN
   Remove site form the next steps
5. Assign weights to site level optimization parameters
6. Normalize the weighted parameters
7. Calculate weighted sum of normalized parameters
8. Select a cloud site with highest weighted sum

Bulk Deployment (BDM) performs the selection of a site for all VMs in a particular deployment request. In this mode, all VMs (belonging to an application) will be deployed on a single suitable site. The pseudo-code for this mode is the same as shown below, the only difference being in Step 1, where the aggregate resource requirements of all VMs (in a deployment request) are considered, which means the workflow (above) is done once for each deployment request. This is different to the previous case (i.e. *Individual Deployment*) where the workflow is repeated for each VM in the deployment request.

In these two modes, the 'suitability' of a cloud site is determined by a combination of multi-criteria optimization (weighted sum approach) and load balancing techniques. Typically, in the multi-criteria weighted sum approach, weights are assigned to different criteria based on the priority or importance of one criterion over another. For example, the weight $W_x > W_y$ if one unit of criterion C_X is worth more than one unit of C_Y considering both criteria (x and y) are measured in the same unit. The weighted sum approach aggregates multiple criteria into a single preference function representing a feasible decision with the highest preference function value being identified as suitable solution. In line with the objective of reducing the CO_2 footprint of applications, weights are assigned by giving more preference (upto 80%) to energy and CO_2 related parameters exposed by the cloud sites, leaving the rest for the resource level parameters such as a load balancing operator that represents the capacity of suitable resources (e.g. available CPU and number of running VMs) at the site. More specifically, in our experimentation we assigned 20% weightage to site power consumption, 30% to CO_2 emissions, 15% to PUE and 35% to load balancing operator.

As it has been stated that the eco-awareness in the Scheduler enables it to route the workload to optimized (in terms of energy efficiency and CO_2 footprint) cloud sites, however when workload spikes occur on a specific (e.g. optimized) site, the load balancing function in the eco-aware Scheduler directs the spill over to other sites that represent a good option. When deploying over a federated cloud environment (such as BonFIRE) effective load balancing is a critical task since based on their locality different cloud sites may rely on different energy sources with consequent implications on their CO_2 footprint. For example, a cloud site in France may have a lower CO_2 footprint compared to a site in UK since France utilizes mainly nuclear energy sources whereas the energy mix in UK has a large proportion of fossil fuel based energy that has a relatively higher CO_2 footprint. Thus, op-

timizing workload distribution solely on CO_2 footprint criteria may result in overloading a particular cloud site. To solve this problem, the load balancing solution (of the eco-aware Scheduler) takes into consideration multiple parameters in its decision making model in order to achieve energy and CO_2 related objectives while assuring optimal utilization and performance of available resources.

To validate the effectiveness of the site selection strategies we designed and carefully conducted some experiments in similar settings/circumstances. In addition to the two eco-aware strategies (as discussed above), experiments were also conducted with a non-eco-aware deployment strategy to get the baseline measures of the application CO_2 footprint. The non-eco-aware deployment strategy performs random selection of a cloud site.

During the experimentation, a sample distributed application was deployed and executed multiple times on BonFIRE infrastructure using different site selection strategies. The mean of the application CO_2 footprint under each deployment strategy is plotted against the application execution time – as shown in Figure 4.

The results of the experimentation reveal clear differences in the CO_2 footprint of non-eco-aware deployment and the two eco-aware deployment strategies (IDM and BDM). In Figure 4 the close proximity of CO_2 footprint achieved by IDM and BDM can be attributed to the fact that both use the same criteria to select a CO_2 friendly site. It is expected that the difference between the two eco-aware deployment strategies may only become radically visible when one cloud site is not able to deploy a complete application, hence some VMs of the application may be deployed in another site under IDM, whereas BDM will deploy the complete application on one site thus ignoring the possibility of deploying some VMs on a more CO_2 friendly site that cannot accommodate all VMs of an application. Furthermore it is important to note that the deployment decision is made at the VM level, the Scheduler does not interact with the application or what goes on inside a VM.

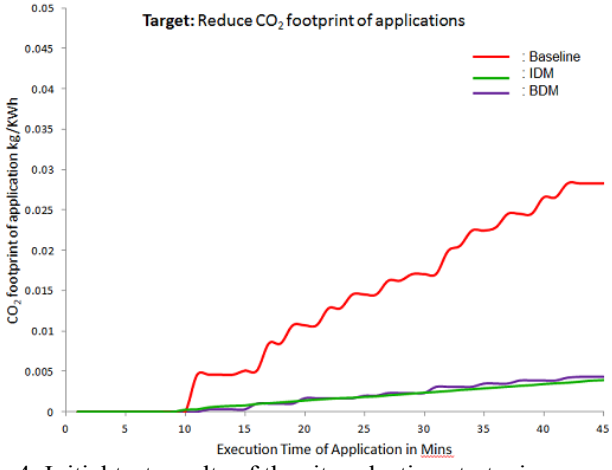

Figure 4: Initial test results of the site selection strategies

Physical Host Selection: Once a site is selected, the following deployment policies of the eco-aware Scheduler aim to determine physical host level deployment configuration of new VMs/applications.

Max-Utilization or VM Consolidation tries to maximize the utilization of individual physical hosts by deploying VMs on fewer hosts within a site e.g. hosts with highest energy consumption. Although there was no support from the underlying BonFIRE infrastructure to take advantage of this, in principle any idle hosts can be switched off or switched to hibernate mode to save energy at infrastructure level and max-utilization is conductive to this scenario.

Min-Utilization or VM Dispersal tries to minimize the utilization of individual hosts by deploying VMs evenly on all available hosts within a site e.g. host with lowest energy consumption. This approach favors VM performance. The pseudo-code of host selection policies is:

Pseudo-code: Host Selection

1. Get VM resource requirements
2. IF host does not have enough capacity for the VM THEN remove host from next steps
3. Assign weights to host level optimization parameter
4. Normalize optimization parameters

5. Calculate weighted sum of normalized parameters
6. Select the host with highest weights

Apart from direct energy consumption information (gathered directly from individual hosts) the host level policies also take into account some resource level parameters that influence the energy consumption of hosts e.g. CPU utilization, Memory and running VMs.

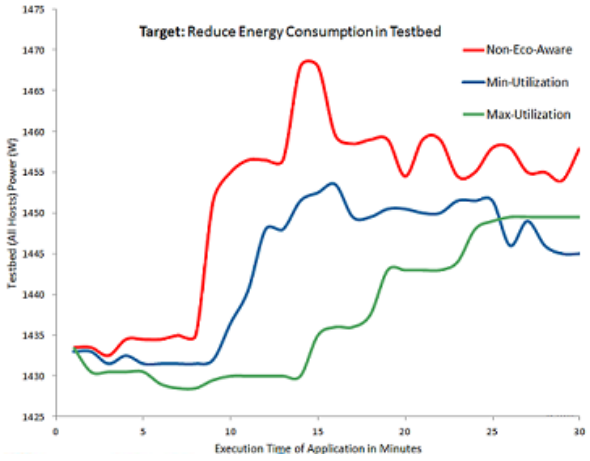

Figure 5: Initial results of physical host selection strategies

Where the traditional cloud schedulers such as OpenNebula Scheduler[2] determine placement of VMs based on resource level parameter of available physical hosts e.g., free CPU, the distinctive feature of our eco-aware scheduling strategies (Max/Min-Utilization) is their consideration for energy consumption of physical hosts, which cannot be solely determined by the number of running VMs or a single resource-level parameter. With energy consumption as the most weighted optimization parameter, our eco-aware host selection policies consider a mix of resource level parameters e.g. CPU, Memory and number of running VMs to determine suitable physical hosts for VM(s).

As we did for the site selection strategies, experimentation was carried out to determine the behavior of the physical host selection policies and their impact on underlying cloud infrastructure. The reason for focusing on cloud infrastructure was the fact that the energy mix remains same within a site and CO_2 is directly derived from the energy consumption of cloud resources or physical hosts. Thus any improvement in the energy consumption of physical hosts can directly impact on the CO_2 footprint at site and application levels. During the experimentation, a sample application was executed multiple times under the two eco-aware host selection policies in a single cloud site. The power consumption of the cloud site (i.e. all host power consumption) was recorded throughout the experiment duration and the median was plotted against the experiment timeline. A third non-eco-aware host selection strategy was introduced in the experimentation for comparison and baseline measures. The non-eco-aware strategy selects a physical host based only on the availability of resources e.g. the host that has the most free resources gets preference for VM allocation.

As shown in Figure 5 the eco-aware Min-Utilization and Max-Utilization policies perform better in minimizing the power consumption of the overall cloud site due to their consideration for eco or energy related parameters in deployment decision making. Between these two eco-aware policies, Max-Utilization gives the better performance proving that by stressing only a subset of available physical hosts the power consumption of the cloud site remains low during the application execution time.

The initial testing of the eco-aware site and host selection strategies helped in determining their effectiveness over non-eco-aware deployment strategies. In this respect, the consideration of environmental impact (by means of considering energy consumption as well as CO_2 emission derived from energy sources) allowed us to move a step forward towards realizing ecologically aware cloud computing.

[2]http://archives.opennebula.org/documentation:rel4.4:schg#pre-defined_placement_policies

3.3. Adaptation of Running Applications

Once the applications are deployed using eco-aware deployment strategies, there are opportunities to further improve their CO_2 footprint and energy efficiency by fine tuning their execution behavior. However, in our study we found out that at runtime or during the execution of application any changes in the deployment configuration of the application may affect the execution behavior or performance of the application, unless there is adequate support from within the application. To address this concern, we have devised an application design framework. The main feature of our framework is a component called Application Controller (AC) that allows the analysis of the status of application and the enactment of certain adaptation actions to reduce application energy consumption and CO_2 footprint.

Assuming that an application might run not only on a single VM but also on several VMs, the AC acts as a middleware layer to communicate with the underlying cloud infrastructure in order to allow the application to both monitor the status of the infrastructure and to enact some adaptation actions to adjust or tune the application execution behavior at run time. More specifically, using the (eco-metrics) monitoring data the AC defines the adaptation logic and an adaptation plan to improve energy efficiency and to reduce emissions. The adaptation actions are enacted after detecting a violation of certain predefined threshold. The use of following adaptation actions has been investigated:

- Changing load distribution between the active VMs: this action changes the workload among the VMs for an application aiming to reach a configuration where the response time might increase and the power consumption decreases, or the power consumption remains stable but distributed in sites where the energy mix is preferable in order to reduce the CO_2 emissions.
- Turning off a VM: if the work of a VM is no longer required, the VM is stopped to reduce the energy consumption of the application;
- Application execution time-shifting: aims to reduce the CO_2 emissions by running the application, or a part of it, later in time when the energy mix is preferable.

The AC can enact the above adaptation actions by considering certain application and VM level metrics (e.g. CPU load and Power consumption). In addition, some task[3] level metrics (i.e., task CPU load, task energy, and task response time) can be considered as they are here required to calculate the Application level metrics.The algorithm used by the AC consists of the following steps:

- The state of the application is evaluated and indicators (preset at application deployment stage) that needs to be improved are identified;
- Starting from this set of indicators, the action that best fit the problem is selected using the information available about the actions impact over indicators and the Adaptive Action Selection Algorithm as described in [33]. The algorithm evaluates the actions which have higher probability of improving the selected indicators while evaluating side effects on others.
- The selected action is enacted and the new state of the system is evaluated. The algorithm also exploits knowledge about correlations among variables when evaluating the set of indicators to be improved.

It is worth noting that the adaptation actions taken at application level are focused on the reduction of CO_2 emissions specifically for a given application, thus the AC aims to find the local optimum in terms of the CO_2 reduction.

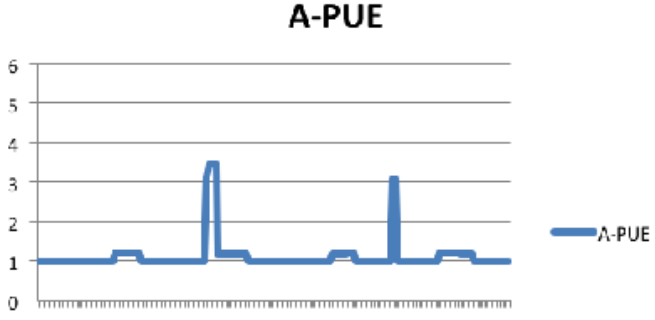

Figure 6: Effects of adaptation on A-PUE

[3] Assuming applications are composed of different tasks that can be deployed different VMs

Initial testing of the AC and adaptation actions reveals encouraging results in terms of energy consumption and CO_2 reductions during the execution of applications [37].

Furthermore, based on to the results obtained during the initial experimentation concerning runtime adaptation actions, we verified how the A-PUE metric could be a good indicator to figure out if a waste of energy is occurring. A-PUE computes the ratio between the energy consumed by the VMs involved in the application and the energy really used by these VMs to run the application. A-PUE equals 1 when all the energy is used to run the application and this corresponds to the ideal situation. On the contrary a value greater than 1 means that there is a fraction of energy consumed by something different than the application. In our case, possible sources of wasting can be the processes running on a VM that are not directly associated to the application (e.g., system daemons).
An example of A-PUE trend is shown in Figure 6 where the effect of the adaptation (i.e., switching off the no longer used VMs) is clear. Here a spike of the A-PUE corresponds to the termination of one application tasks and, as adaptation is enacted, the power consumed of the host is not wasted.

4 VALIDATION OF THE ECO-AWARE SOLUTION

The approach and its proposed components have been evaluated within four case study applications. All case study applications were distributed, long running, compute intensive in nature and were designed (in previous EU projects or commercial initiatives) to solve complex problems specifically within cloud or HPC environment, in this respect they were considered to represent typical cloud workloads:

- **The Eels** case study refers to a computation intensive HPC application, initially monolithic that it has been adapted to run in a cloud environment in order to exploit the scalability and computation power on offer.
- **The eBusiness** case study is a SOA based simulation of a common e-business application. It uses web service to implement cross-organizational workflows.
- **The Bones** case study is a computation intensive and long running HPC application computing finite element simulations which has been ported to a federated cloud environment.
- **Data Analytics as a Service** (DAaaS) is a commercial cloud application that receives input data from different sources in a marketplace and allows users to configure different analytics procedures that can be performed over the data. The results of the analytics are generated graphically for the users.

4.1. Experimental Design

The AC has been adopted only for the Eels case study and the eBusiness case study since they allow individual controlling of their components/modules without disturbing the overall execution of the application. The case studies selected for the validation of our eco-aware approach can be divided into two main groups:

A. Case studies that support only the eco-aware scheduling: Bones and DAaas.

B. Case studies supporting both eco-aware scheduling and run-time adaptation: Eels, E-Business.

This distinction depends on the nature of the application i.e. the Eels and E-Business applications can be modelled as business processes and their structure can be altered during the execution without compromising the soundness of the application. In the Bones and the DAaaS case studies, the existing code does not allow the system to modify the behavior of the application. As a consequence, this difference in the nature of case studies provides a good way for analyzing the effectiveness of our eco-aware scheduler (with the help of case studies in group A) and the Scheduler + AC (as in the case of group B case studies).

As the goal of the experimentation is to verify the effectiveness of our solutions in reducing CO_2 emissions as a consequence of minimizing energy consumption and other contributing factors, we compare the CO_2 emission of a **baseline scenario** where applications are executed using traditional or existing scheduling techniques, and an **optimized scenario** where our eco-aware mechanisms are switched on. Regardless of the case study, the baseline scenario is obtained as follows:

- Submission of an application deployment request to the existing or traditional cloud scheduler.
- Then, executing the experiment (i.e. application) and monitoring its CO_2 emissions.

Having the eco-aware Scheduler disabled means that each of the VMs requested by the application is randomly deployed on one of three available cloud sites (i.e., UK, Germany and France). Then, within a site, the VM is assigned to a host according to the internal policies of the cloud manager currently installed on the site e.g., BonFIRE uses OpenNebula.

For the optimized scenario, two distinct approaches were followed depending on the case study:

- The experimentation procedure for Group A follows:
  - Sending an application deployment request (with number of VMs and specific resource requirements) to the eco-aware Scheduler enabling the eco-aware deployment.
  - Running the experiment (the AC is not invoked in this case)
- The experimentation procedure for Group B follows:
  - Sending an application deployment request (with number of VMs and their specific resource requirements) to the eco-aware Scheduler enabling the eco-aware deployment.
  - Running the experiment enabling the AC.

In addition to the comparison between the baseline scenario and optimized scenario involving the three cloud sites, we also compare:

- For the case studies in Group A, a baseline scenario and optimized scenario where individual cloud sites are considered. Experimentation in the single site configuration allowed us to validate that the optimization of energy consumption and CO_2 footprint was indeed possible within single site configuration i.e. by selecting specific physical hosts in a cloud site in a way that can contribute towards minimizing the energy consumption and CO_2 footprint of the cloud site.
- For the case studies in Group B, the baseline scenario and optimized scenario are computed excluding the cloud site in France as a possible deployment option for the VMs. This was needed because the cloud site in France reports CO_2 emissions that are extremely lower than the other two sites. In this way, we aim to verify that the eco-aware optimization techniques at scheduling and application level really work with sites that have comparable energy mix factors

To obtain reliable results, for both the baselines and the optimized versions, the experiments with all case study applications have been executed up to 300 times to consider different site utilization scenarios and different values for the energy mix factor as this factor changes by the day. Although the case studies applications can be scaled up to use large number of VMs, the experimentation was conducted using a moderate number of VMs to ensure uninterrupted execution of applications within the resource constraints of the BonFIRE production infrastructure.

5 RESULTS

The results of the experimental validation of the case study application are described next.

5.1. Eels Case Study

The Eels case study is an HPC application that simulates the trajectories followed by Eels during their travel in the Atlantic Ocean from where they were born (Sargassi Sea) to the European Coast. This application implements a parameterized model providing a biologically reasonable description of the life-cycle of Eel larvae based on the current knowledge of the species and the predictions of metabolic ecology. The necessary data is already stored on a cloud storage resource (Oceanographic Data Block).

Unit $grCO_2eq$	**Baseline**	**Optimized**	**Sites**
CO_2 Application	150.556	9.315	3 sites
CO_2 Overhead	10.98	2.353	3 sites

Figure 7: Eels case study CO_2 footprint reduction of 93.90% considering the three cloud sites

To test this application, for each run, we reserve five VMs; 3 VMs (2 CPUs, 2 GByte memory) for the actual execution of the application, 1 VM (0.5 CPU, 256 MByte memory) for the execution of the application controller (for the optimized scenario) and 1 VM (1 CPU, 1 GByte memory) for monitoring the experiment.

Comparison between the baseline scenario and the optimized scenario involving all the three cloud sites is shown in Figure 7. *Overhead* is an addition level of details that we studied in our experimentation, which may not be relevant in production settings. Overheads refer to the CO_2 emissions of the VM where the AC and the Monitoring of the experiment are running. These represent the additional CO_2 emissions that can be attributed to the overall application CO_2 footprint. In case of the baseline, only the monitoring is considered, as the AC is not required. As reported in Figure 7 the CO_2 reduction is of 93.90% in the optimized scenario. This set of experiments considered the site located in France as possible deployment option where, as mentioned above, the energy mix factor is very low. However, the CO_2 emissions related to the overhead remain the same (21.39% for the baseline, 20.16% the optimized running). Figure 8 reports the CO_2 emissions excluding the site in France from being a possible deployment option for the VMs.

Unit $grCO_2eq$	**Baseline**	**Optimized**	**Sites**
CO_2 Application	235.763	91.892	Germany
CO_2 Overhead	57.786	43.752	Germany

Figure 8: Eels case study CO_2 footprint reduction of 32.25% while excluding France as deployment option

As reported in the figure, even in this case, the CO_2 emission are significantly reduced, i.e., 53.80%, with respect to the baseline, with an impact of the overhead of 32.25% (19.68% for the baseline by not including VM for the AC).

5.2. Bones Case Study

The Bones case study application is concerning the simulation of human bones to model and support diagnostic decisions with respect to the type and positioning of implants used for certain pathologies.

Unit $grCO_2eq$	**Baseline**	**Optimized**	**Sites**
CO_2 Application	0.205	0.046	3 sites
CO_2 Overhead	0.032	0.005	3 sites

Figure 9: Bones case study CO_2 footprint reduction of 77.52% considering the three cloud sites

The Bones case study is parallel in nature and may be of interest for the HPC community to study how HPC oriented programs can be executed on a cloud infrastructure. The total experiment's duration was set to 2 hours while the actual intense computation took abound half of this time. This case study application requires 3 VMs with 2 CPUs, 4096 MBytes RAM and 5 VMs with 2CPUs, 2048 MBytes RAM.

While analyzing the experimental results we have observed that the eco-aware scheduling results in CO_2 savings when compared to non-optimized values, both where the VMs are set to be all deployed at either of the three sites as shown in Figure 9 or distributed across the three sites, as shown in Figure 10. This validates our rational for experimentation in single site configurations since the heterogeneity of hosts even in a single site provides the Scheduler with considerable optimization opportunities.

For this case study the single-site experiments was conducted using the cloud site in UK due to lack of adequate resources at other BonFIRE sites. In a way this further validates our scheduling and alleviates concerns of any association with the underlying infrastructure.

Unit $grCO_2eq$	**Baseline**	**Optimized**	**Sites**
CO_2 Application	0.015	0.008	UK

CO_2 Overhead	0.253	0.04	UK

Figure 10: Bones case study CO_2 footprint reduction of 56.41% in a single cloud site.

5.3. eBusiness Case Study

The eBusiness case study is a simulation of web services that uses certain benchmarks to mimic business operations with different characteristics e.g. CPU and Memory intensive operations and I/O intensive operations. The case study simulates web services by randomized queries running on different hosts. Further, different processes are called simultaneously after randomized time intervals and various kinds of load (CPU, I/O, memory) are stressed.

Unit $grCO_2eq$	**Baseline**	**Optimized**	**Sites**
CO_2 Application	0.156	0.068	3 sites
CO_2 Overhead	0.033	0.015	3 sites

Figure 11: eBusiness case study CO_2 footprint reduction of 68.21% considering the three cloud sites

The eBusiness experiments were performed using similar configuration as for the Bones case study i.e. 3 VMs with 2 CPUs, 4096 MBytest RAM and 5 VMs with 2 CPUs, 2048 MBytest RAM). The experimental results show significant reduction in CO_2 footprint of the application in multi-site (Figure 11) and in single site deployment (Figure 12).

Unit $grCO_2eq$	**Baseline**	**Optimized**	**Sites**
CO_2 Application	0.014	0.007	Germany
CO_2 Overhead	0.111	0.035	Germany

Figure 12: eBusiness case study CO_2 footprint reduction of 84% in a single cloud site (Germany)

5.4. DAaaS Case Study

Data Analytics as a Service is a proof of concept analytical platform delivered using a cloud-based model, where various tools for data analytics are available and can be configured by the user to efficiently process and analyze huge quantities of heterogeneous data.

For the testing in this case study we simulated the behavior of a user that connects to the DAaaS page, uploads a small collection of data, executes some simulations over it using Apache Hadoop and sees the different results. This test with the same input data and same user behavior was repeated all over again in simulated settings. A complete deployment of the DAaaS case study was composed of 5 VMs (1 CPU, 1 GB of RAM).

Unit $grCO_2eq$	**Baseline**	**Optimized**	**Sites**
CO_2 Application	30.641	3.658	3 sites
CO_2 Overhead	6.644	1.354	3 sites

Figure 13: DAaaS case study CO_2 footprint reduction of 88% considering the three cloud sites

Figure 13 presents the results of the experiments concerning the DAaaS case study in multi-site deployment mode of eco-aware Scheduler i.e. using physical hosts available at the three different sites.

Unit $grCO_2eq$	**Baseline**	**Optimized**	**Sites**
CO_2 Application	280.401	52.281	Germany
CO_2 Overhead	67.934	20.242	Germany

Figure 14: DAaaS case study CO_2 footprint reduction of 82% in a single cloud site (Germany)

It is important to note that the DAaaS case study does not involve the use of the AC, hence the reduction of CO_2 footprint can only be attributed to the eco-aware scheduler, and we are observing a reduction of CO_2 usage of up to 88%. The overhead for both cases is quite high, of around 21% for the baseline case and 37% for the optimized case. Figure 14 represents 82% reduction of the CO_2 footprint of the DAaaS application deployed in a single site.

6. CONCLUSION AND FUTURE WORK

In the wake of increasing popularity of cloud computing, achieving energy efficiency has been highlighted as one of the key challenges in this area [36]. Tracing energy consumption and CO_2 footprint of cloud applications becomes more difficult when applications span multiple clouds and use heterogeneous resources.

In this respect, the contribution of this paper is an approach that brings together a set of innovations such as eco-metrics, eco-aware scheduling, eco-aware monitoring and eco-aware adaptation mechanisms to not only quantify the environmental impact but also to deliver significantly reductions in CO_2 emissions of cloud applications. We have shown that an eco-aware scheduler can have a positive impact on the CO_2 footprint of cloud applications and the adaption actions enacted by the application controller can add further efficiency at run-time, supported by an eco-aware monitoring that measures and collects eco-metrics.

We have performed real experiments of our approach over an experimental federated cloud facility called BonFIRE. BonFIRE allowed us to implement eco-metrics and extend the existing monitoring mechanism, enabling us to quantify the environmental impact of cloud computing. These features are now part of the BonFIRE Open Access offerings. The flexibility and control offered by BonFIRE cannot be expected from any commercial cloud provider. For example, experimentation of our eco-aware scheduling mechanism in BonFIRE has demonstrated the ability to significantly reduce the energy consumption and CO_2 footprint of cloud applications, as compared to the baseline scenario using the scheduling mechanism currently in place at the BonFIRE sites. In some cases, such a reduction reaches over 80%, which can be considered a big achievement of our work. The eco-aware scheduling has been tested with different kinds of applications including HPC, cloud and interactive applications. The results of our experiments also validate that the eco-aware Scheduler provides substantial reduction of CO_2 emissions not only for multi-site experiment deployment but also for one-site deployment where the scheduling logic leads to the selection of more efficient hosts for VMs. If the site has significant heterogeneity in its computational resources, this makes a bigger difference e.g. we noticed that the eBusiness case study achieves larger CO_2 emission reduction at single-site deployment in Germany than single-site deployment in UK since the site in UK provides less heterogeneous resources. The results described in the paper can be useful for cloud application developers and infrastructure providers that are interested in decreasing the CO_2 emissions and improving energy efficiency in cloud computing. The solutions discussed in this paper have been made available as open source resources[4]. Further, the eco-metrics are already incorporated in BonFIRE offering[5] to enable users to monitor the energy consumption and CO_2 emissions of their VMs. The eco-aware scheduler and application controller are currently being exploited in order to unravel their potential in other cloud environments.

In our future work, we aim to provide support for our open source solutions and extend our work by (a) devising reliable CO_2 based accounting models for cloud providers, and (b) developing the Application Controller as a formal framework for developing CO_2 aware applications.

[4] https://github.com/ECO2Clouds/r2
[5] http://www.bonfire-project.eu/home

REFERENCES

[1] CPD Webpage: https://www.cdp.net/en-US/WhatWeDo/CDPNewsArticlePages/cloud-computing-can-dramatically-reduce-energy-costs-and-carbon-emissions.aspx - accessed on 21.11.2-14

[2] The Climate Group Webpage: http://www.theclimategroup.org/what-we-do/news-and-blogs/eu-and-global-tech-sector-measuring-ict-footprint/

[3] European Union Emissions Trading Scheme, http://ec.europa.eu/clima/policies/ets/.

[4] Carbon Reduction Commitment Energy Efficiency Scheme, http://www.environment-agency.gov.uk/business/topics/pollution/126698.aspx. .

[5] "Pushing our energy performance even higher with ISO 50001 certification", http://googlegreenblog.blogspot.co.uk/2013/07/pushing-our-energy-performance-even.html, July 24th 2013.

[6] "Efficiency: How we do it", http://www.google.com/about/datacenters/efficiency/internal.

[7] Nguyen Quang-Hung, Nam Thoai, Nguyen Thanh Son: Enery Efficient Allocation of Virtual Machines in High Performance Computing Cloud. arXiv:1310.7801v2.

[8] Execution Models For Energy-Efficient Computing Systems – EXCESS, http://excess-project.eu/.

[9] Exa2Green: energy-aware numeric, http://exa2green-project.eu/

[10] Argo: An exascale operating system, http://www.mcs.anl.gov/project/argo-exascale-operating-system.

[11] Zhi Zhou; Fangming Liu; Yong Xu; Ruolan Zou; Hong Xu; Lui, J.C.S.; Hai Jin, Carbon-Aware Load Balancing for Geo-distributed Cloud Services, In 21st IEEE Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems, 2013

[12] Peter Xiang Gao, Andrew R. Curtis, Bernard Wong, and Srinivasan Keshav. It's not easy being green. In Proceedings of the ACM conference on Applications, technologies, architectures, and protocols for computer communication. 211-222, 2012

[13] Zhi Zhou, Fangming Liu, Bo Li, Baochun Li, Hai Jin, Ruolan Zou, Zhiyong Liu: Fuel Cell Generation in Geo-Distributed Cloud Services: A Quantitative Study. ICDCS 2014: 52-61, 2014

[14] Adept: addressing energy in parallel technologies, http://www.adept-project.eu.

[15] "The Green 500", http://www.green500.org/lists/green201311.

[16] Amazon Elastic Compute Cloud (Amazon EC2), Amazon Web Services, http://aws.amazon.com/ec2/.

[17] "Nvidia Grid: Graphics-Accelerated Virtual Desktops and Applications", http://www.nvidia.com/object/enterprise-virtualization.html.

[18] Alan Roytman, Aman Kansal, Sriram Govindan, Jie Liu, and Suman Nath, PACMan: Performance Aware Virtual Machine Consolidation, in 10th International Conference on Autonomic Computing (ICAC*)*, 2013.

[19] Atefeh Khosravi, Saurabh Kumar Garg, and Rajkumar Buyya, Energy and Carbon-Efficient Placement of Virtual Machines in Distributed Cloud Data Centers, Proceedings of Euro-Par, 26-30 Springer, 2013.

[20] Kolodziej, J., Khan, S.U., Wang, L., Zomaya, A. Y., 2012. Energy efficient genetic-based schedulers in computational Grids. Concurrency and Computation: Practice and Experience

[21] Lindberg. P., et al., 2012. Comparison and analysis of eight scheduling heuristics for the optimization of energy consumption and makespan in large-scale distributed systems. In Journal of Supercomputing, Vol. 59, Issue 1, 323-360.

[22] Kipp, A., Jiang, T., Fugini, M.G., Salomie, I., 2012. Layered Green Performance Indicators, Future Generation Comp. Systems. 28(2), 478-489

[23] Beloglazov, A., Buyya, R., Choon Lee, Y., Zomaya, A.Y., 2011. A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing Systems. Advances in Computers 82: 47-111

[24] Wajid, U., Marín, C.A, Mehandjiev, N., 2013. Optimizing Service Ecosystems in the Cloud, in *The Future Internet*, LNCS, Volume 7858, Springer.

[25] Greenpeace, 2012. How clean is your cloud?, http://www.greenpeace.org/international/en/publications/Campaign-reports/Climate-Reports/How-Clean-is-Your-Cloud/

[26] J. Baliga, R.W.A. Ayre, K. Hinton, R.S. Tucker, Green Cloud Computing: Balancing Energy in Processing, Storage and Transport, Proceedings of the IEEE 99(1): 149-167, 2011.

[27] S. U. Khan, I. Ahmed. A cooperrative game theoretical tehcnique for joint optimization of energy consumption and response time in computational grids. IEEE Transactions on Parallel and Distributed Systems, pp 246-360. 2009

[28] 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.

[29] 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, volume 5900 of Lecture Notes in Computer Science, 99–114, 2009.

[30] B. Steigerwald, et al.. Creating energy efficient software. http://isdlibrary.intel-dispatch.com/isd/146/creating energy-efficient software.pdf, 2007

[31] C. Lefurgy, K. Rajamani, F. L. Rawson, Wesley M. Felter, Michael Kistler, Tom W. Keller: Energy Management for Commercial Servers. IEEE Computer 36(12). 2003

[32] The Green Grid Consortium: Harmonizing global metrics to measure data centre energy efficiency. Oct 2012. Available on line: http://iet.jrc.ec.europa.eu/energyefficiency/sites/energyefficiency/files/files/d ocuments/ICT_CoC/harmonizing_global_metrics_for_data_center_energy_efficien cy_2012-10-02.pdf

[33] M.Vitali, “Measuring and Improving Energy Efficiency of a Data Center in a Self-Adaptive Context”, PhD thesis, Politecnico di Milano, Italy, 2014

[34] Cinzia Cappiello, Pieluigi Plebani., and Monica Vitali. Energy-Aware Process Design Optimization. EuroEcoDC Workshop, IEEE, 2013.

[35] GHG Protocol ICT Sector Guidance Webpage: http://www.ghgprotocol.org/files/ghgp/GHGP-ICT-Cloud-v2-6-26JAN2013.pdf- Accessed on 25.11.2014.

[36] Adel. N. Toosi, Rodrigo. N. Calheiros, Rajkumar Buyya. Interconnected Cloud Computing Environments: Challenges, Taxonomy and Surveys. ACM Computing Surveys. Vol 47, Issue 1. 2014

[37] Good practices on cloud computing energy consumption optimization – Whitepaper, available at: http://eco2clouds.eu/wp-content/uploads/D3.5-v1.0.pdf

[38] Bartalos, P.; Blake, M.B., Green Web Services: Modeling and Estimating Power Consumption of Web Services, In proceedings of IEEE ICWS, 2012 .

[39] Kansal, F. Zhao, J. Liu, N. Kothari, and A. A. Bhattacharya, Virtual machine power metering and provisioning, In 1st ACM symposium on Cloud computing, ser. SoCC ’10. pp. 39–50, 2010,.

[40] Realising and enhanced monitoring and data analysis environment. ECO2Clouds project report, 2014. http://eco2clouds.eu/wp-content/uploads/D3-4_EnhancedMonitoringRealizationAndDataAnalysis_v1_2.pdf