Good practices for Cloud Computing Energy Consumption and CO_2 Emissions Optimisations

White Paper

ECO_2Clouds project participants

Contacts: eco2clouds@elet.polimi.it

Web: http://eco2clouds.eu

September 2014

Executive Summary

Data centers are responsible for the generation of a significant portion of CO_2 emissions. Consequently, the increasing adoption of cloud computing is having a big impact on the environment since the energy consumption of data centers and the resulting emissions are significantly growing. Researchers and practitioners in this field are looking for methods to improve the energy efficiency of data centers and decrease the energy related emissions. This paper describes how the definition, assessment, utilization and monitoring of energy metrics and the use of innovative application scheduling and adaptation strategies in EU funded ECO_2Clouds project help in optimizing CO_2 footprint and energy consumption of cloud applications as well as the underlying infrastructure. Based on the results of ECO_2Clouds project, this document hereby describes the lessons learnt and good practices that can be adopted to achieve energy efficiency and reduction in environmental implications of cloud computing.

Keywords: CO_2 emissions, energy efficiency, cloud computing

1 Introduction

With the increasing reliance on ICT to perform daily tasks and business operations there are growing concerns about the vast amount of energy consumed and the resulting environmental implication of IT systems. Such concerns are particularly relevant for large scale datacenters and cloud computing infrastructures; seen as biggest contributors of CO_2 emissions in the ICT domain. With ICT contributing up to 4% of EU's total CO_2 emissions, any improvements made in the achieving energy efficiency and reducing CO_2 emissions in this sector can have a significant impact on the environment.

It is therefore important to consider the environmental consequences of the growing adoption of cloud computing and to find methods and tools to improve the energy efficiency and decrease the CO_2 emissions related to cloud resource management and usage. The ECO$_2$Clouds project[1] has progressed towards this direction by providing tools that, if adopted by a cloud provider, allow (i) the cloud provider to determine CO_2 friendly deployment of applications and (ii) the users to be aware of the environmental impact of their cloud applications. The ECO$_2$Clouds solutions for energy aware and CO_2 friendly cloud computing have been prototyped over the BonFIRE [1] platform, but it is designed to be generally applicable to cloud environments.

In the ECO$_2$Clouds project we want to address new questions: what is the impact, in terms of CO_2, of running applications in a federated cloud environment? Is there space for the cloud provider to reduce the CO_2 emission of their sites also beyond the direct effect of energy reduction? How can CO_2 emissions be measured at cloud and at application level?

These questions are also raised in a recent report produced by Greenpeace [3] highlighting that, besides the amount of energy consumed, an important factor that cloud providers should take into account in their journey toward sustainability is the energy mix that reveals the sources used to produce the energy consumed by the cloud facilities. Different energy sources cause different CO_2 emissions and thus they have a different impact on the environment. Energy mixes and CO_2 emissions largely vary from country to country and if cloud providers use their own energy sources they might vary from site to site also in the same country. Such differences imply that considering the different percentages of green and non-green energy sources might become a valuable driver in the selection of the cloud site in which the applications should be deployed. Moreover, CO_2 emissions have been also included in the set of metrics defined by the Green Grid Consortium[2] that aims to provide indicators that go beyond the PUE (Power Usage Effectiveness) to measure the efficiency of a data center[3][4]. In such a set of metrics aspects like productivity and efficiency of the IT system as well as the resource saturation, and CO_2 emissions have been considered but, they are mainly limited to the infrastructure layer.

The goal of this white paper is to describe how the questions posed above have been addressed in the ECO$_2$Clouds project, taking into account the infrastructure, virtualization, and application layers according to three main perspectives: what are the requirements for the reduction of CO_2 emissions and the energy consumption, what are the metrics to be considered for evaluating the environmental impact of the cloud environment, how to collect data to compute these metrics.

---

[1] http://eco2clouds.eu
[2] http://www.thegreengrid.org/

As a result, the ECO_2Clouds project defines and assesses green-related metrics (see Section 2 and Section 3). Experience gathered from the theoretical definition of relevant metrics and their practical assessment after implemented in the ECO_2Clouds Monitoring system (see Section 4) provides a valuable knowledge that can be used in making the deployment and deployment adaptation decisions. The ECO_2Clouds project results suggest that it is important to adopt an application perspective, as energy savings and optimization to reduce CO_2 emissions can be achieved by considering the characteristics of the applications deployed and not only the features of the cloud site resources. Such characteristics are captured by the ECO_2Clouds Application Profile (see Section 5). Furthermore, the ECO_2Clouds Scheduler and the Application Controller (see Section 6) are the main components that aim to reduce the CO_2 emissions by optimizing the deployment phase and the application execution.

The modular approach undertaken in ECO_2Clouds project enables the reuse of key ingredients of ECO_2Clouds solution n other cloud architectures.

2 Infrastructure Support for Energy Efficiency in Cloud Computing

Infrastructure support for energy efficiency is mainly reliant on taking steps towards the provision of necessary information (or metrics) that can help quantify the energy consumption and environmental impact of application deployment and execution.

In cloud computing Carbon-aware infrastructures can provide sufficient information to estimate the cost of using computing resources in grams of CO_2. This raises the following five fundamental questions:

- How to gather information about the energy sources used by the infrastructure, and their CO_2 intensity? It is possible to measure power consumed on an infrastructure through external probes and/or computing hardware functionality like Intelligent Platform Management Interface (IPMI) implementations. There is no counterpart for CO_2 emissions, thus CO_2 intensity is necessary so as to transform any information about energy used into carbon costs. The energy mix or CO_2 intensity provided by the energy supplier can be fixed by contract in some cases, or collected from public data sources maintained by the supplier. Because energy can be stored at a cost (for example by pumping water up a dam), traded (as in imports and exports between countries) and lost in transport, it is sometimes difficult to evaluate accurately the energy mix and thus the corresponding CO_2 emissions. Moreover, it might be desirable to take into account a complete lifecycle analysis to account for grey energy, i.e., the hidden energy used in the building and electricity distribution infrastructures. In a first step towards building carbon-aware infrastructures, we recommend that infrastructure providers limit their data collection to the direct CO_2 cost of energy sources that contributes to the energy mix used by the infrastructure. Most energy providers or countries provide such data in real time in the public domain. As smarter energy grids become available, this strategy can be adapted to provide better accuracy, and include other costs of production energy such a radio-isotopes produced and grey energy.
- How to gather information about energy usage of all the computing elements powering the infrastructure? At a fundamental level, managed power distribution units (PDUs) can provide sufficiently accurate data. This needs to include power used by cooling equipment, compute, networking and storage elements used to run the infrastructure and those directly used by the clients of the infrastructure. Ideally, power distribution units would be able to report on energy usage for each plug it monitors with sufficient precision to report new

values every minute or more frequently. Unfortunately, we have found that some of the off the shelf PDUs do not have enough precision when reporting on energy usage through a given plug. Therefore, we recommend to measure frequently only the power usage of each plug and derive the energy usage from such values if necessary. Moreover, another issue to address when using PDUs is that computing elements are not always directly linked to power plugs. Blade servers for example might share 2 power plugs between 10 machines. We have found it important to build a monitoring infrastructure where power usage of each node is collected. When Intelligent Platform Management Interface (IPMI) enables self-monitoring of power usage by each node, power consumption assessment can be done dynamically. The power consumption of a node *k* can be calculated as:

$$PowerConsumption_k = PowerConsumptionPlug \times \frac{PowerConsumptionIPMI_k}{\sum_k PowerConsumptionIPMI_k}$$

where *PowerConsumptionPlug* is the power consumption measured at plug level and $PowerConsumptionIPMI_k$ is the IPMI measurement of power consumption of the k-th node. If chilling equipment cannot be measured, an estimation of the PUE of the datacenter room used can be enough.

- How to measure the usage of a virtual machine (VM)? We recommend here that monitoring tools provided by the hypervisor used to run the VMs be used to collect resource usage of each VM. We have been able to collect CPU usage, network usage, memory usage and disk I/O usage incurred by each VM running on a given host. On the basis of these data ECO_2Clouds derives the current VM power consumption.
- How to collect and store all data points for monitored values? We have found very useful to separate the monitoring infrastructure used for all long-lived elements (PDUs, physical hosts) and short-lived elements (VMs). The latter will be made available to users and the former only to infrastructure operators and trusted third parties. In order to push information about VMs measured at infrastructure layer to the user data plane we recommend a message queue architecture, with a component pushing messages published and collected on the infrastructure dataplane for monitoring purposes on the relevant user dataplanes so it can be consumed by users in a scalable fashion.
- What information is made available to users and how is it made available? We recommend that users be made aware of (i) the resource usage by their VMs as measured by the host of their VMs, (ii) a calculated estimation of the power usage of their VMs and (iii) the energy mix of the infrastructure provider and the related CO_2 intensity.

We will detail the approach chosen to answer these fundamental questions in the ECO_2Clouds project in the next sections.

3 Quantification of Energy Consumption and Environmental Impact

In ECO_2Clouds the quantification of energy consumption and environmental impact is performed based on the availability of a layered set of metrics. In fact, the model adopted in ECO_2Clouds consists of (i) an *infrastructural layer* that refers to a federated environment composed by several sites and their resources*;* (ii) a *virtualization layer* that contains the Virtual Machines (VMs), which are responsible to host the applications; (iii) the *application layer* that includes the applications running on one or more VMs (see Figure 1**Error! Reference source not found.**).

In the following section, for each level we provide the list of considered metrics for each level. More details about assessment formula and methods can be found in [5].

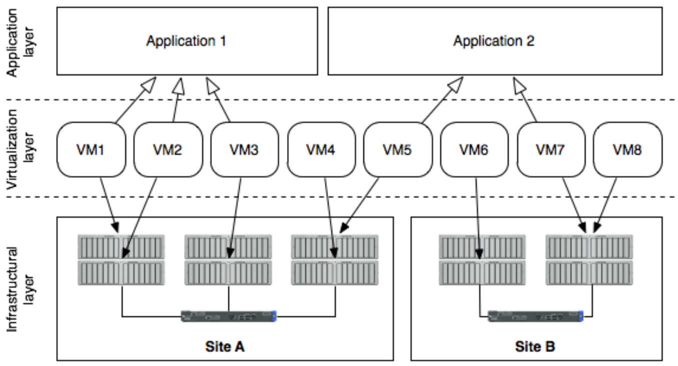

Figure 1 - Layered cloud model used in ECO_2Clouds

3.1 Infrastructure Layer Metrics

This set of metrics includes metrics related to the hosts and the cloud site. They are presented in Table 1 and Table 2.

Table 1: Infrastructure Metrics for the Host

Metric	Monitored parameter
Power consumption	Power consumption of a host
Disk IOPS	I/O operations of the disk, including VM and operating system activity
CPU utilization	Utilization of the processor cores inside a host
Availability	A host is available if it is seen as "online" within the cloud manger of the site.

Table 2: Infrastructure Metrics for the Site

Metric	Monitored parameter
Site utilization	The ratio of available free cores of all worker nodes over the total number of cores of all worker nodes
Storage utilization	Percentage of used storage
Availability	The cloud API is responsive and at least one host is available
PUE	Determines the energy efficiency of a site

In addition to the infrastructure site and host metrics, each site measures the following energy mix metrics:

- *Energy Mix:* measures the amount (in percentage) of the different energy sources used by each provider: 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, Pumped storage, Solar, Total green, Water and Wind.
- Produced CO_2: provides the amount of CO_2 *per kWh* (g/kWh). It is important to note that the providers could have a dynamic energy mix measured by the above mentioned energy mix metrics. However, providers could have fixed values due to the power supplier contract. Thus, in

case a measurement of the energy mix is not possible, there is the possibility to consider fix values.

- *Grid Total:* measures the total amount of produced power (MW) by the power provider/provider grid operator.
- *Imported/Exported:* measures in percentage the amount of imported and/or exported power.

3.2 Virtualization Layer Metrics

The virtualization layer metrics used in ECO_2Clouds are defined in Table 3. These metrics refer to the VMs characteristics.

Table 3: Virtual Machine Metrics

Metric	Monitored parameter
CPU usage	Processor utilization as seen by the running virtual machine
Storage Usage	Storage utilization percentage
I/O usage	Process execution time for disk read/write activities
Memory usage	Ratio of used over available VM memory
Power consumption	Currently consumed power
Disk IOPS	Disk input/output rate reported by the operating system

3.3 Application Layer Metrics

These metrics are defined to enable monitoring of application behavior during their execution in the cloud environment. Table 4 presents the selected application layer metrics used in ECO_2Clouds.

Table 4: Application layer metrics

Metric	Monitored Parameter
Task execution time	The time taken to execute the specific task.
Application execution time	The time taken to execute the whole application.
Power consumption	The power currently consumed by the analyzed application.
Response Time	The average time taken to handle user requests. In particular, it measures the average time interval from a user request to the service response. This metric is particularly relevant for interactive applications; for batch applications the response time will coincide with the application execution time.
Throughput	Rate of executions of an application: e.g., the amount of operations executed in a specific time interval (e.g., 30 seconds).
A-PUE (Application PUE)	The application power usage effectiveness (PUE) reveals energy wastes during the execution of applications.

Application Energy Productivity (A-EP)	The Application Energy Productivity measures the energy required to execute a task.
Application Green Efficiency (A-GE)	The Application Green Efficiency metric measures the percentage of green energy used to run the analyzed application.

3.4 Energy Consumption

The assessment of the *Energy Consumption* is performed at the host level for the physical nodes. The measurement at host level is enabled through the installation of power distribution units (PDUs). The monitoring system can collect data regarding the consumed power and energy for each physical node in VA, W and Wh. These measurements are the base for the calculation of the power consumption per VM (see Figure 2).

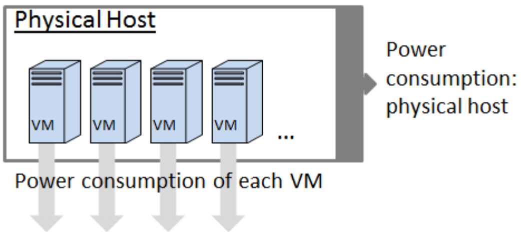

Figure 2: Power consumption per VM

In fact, at the virtualization layer, the power consumption of a VM is estimated as follows:

$$\frac{IDLE\ power\ of\ the\ host}{Number\ of\ VMs} + Dynamic\ power\ of\ the\ host \times \frac{CPU\ used\ by\ the\ VM}{CPU\ used\ by\ all\ VMs\ running\ on\ the\ host}$$

All these values, measured at the infrastructure layer are made available on the user data plane. Therefore, both the infrastructure provider and the users are able to keep track of the power consumption of a VM.

3.5 Carbon Emissions

Carbon emissions can be assessed in different ways. Some countries publish the real time energy mix via public web sites. For example, France energy mix can be retrieved through the information service *éCO2mix* available on the RTE website[3] and data about the energy generation in UK are available through the Balancing Mechanism Reporting System (BMRS) website[4]. The availability of real time data allows us to have a more accurate measurement of carbon emissions. Since the web sites publish the electricity power generated by the different energy sources, it is possible to calculate the emission factor as a weighed average of the sources emission factors on the basis of the percentage of power generated. Formally, if TP is the total power generated in a country, SP_k is the power generated by the k-th source and ef_k is the emission factor related to the k-th source, the total emission factor of the cloud site can be calculated as follows:

[3] http://www.rte-france.com/fr/
[4] http://www.bmreports.com/

$$ef = \sum_{k=1}^{K} \frac{SP_k}{TP} ef_k$$

Besides real time values, there are electricity operators that periodically publish aggregated emission factors of the different countries in a specific period. In this case, assuming that we know the average power consumption (AP) for a specific site, the energy (kWh) consumed in a specific period can be estimated by multiplying AP by the number of hours in the considered period. CO_2 emissions result by multiplying the energy consumed by the emission factor (that is a constant calculated considering the average energy mix in a certain period).

4 The ECO$_2$Clouds Monitoring Infrastructure

In the scope of the ECO$_2$Clouds project, in order to collect all the parameters needed to assess the metrics introduced in the previous section, a monitoring infrastructure was developed and installed at all infrastructure providers' sites.

In details, monitoring infrastructure is based on Zabbix[5], an enterprise open source monitoring solution. The monitoring enables users to retrieve the required data from the infrastructure and VM layer. Figure 3 presents the *monitoring infrastructure*, including the three layers *infrastructure*, *virtualization* and *application* layer, and the related monitoring components, the *abstraction API* and the *monitoring collector*.

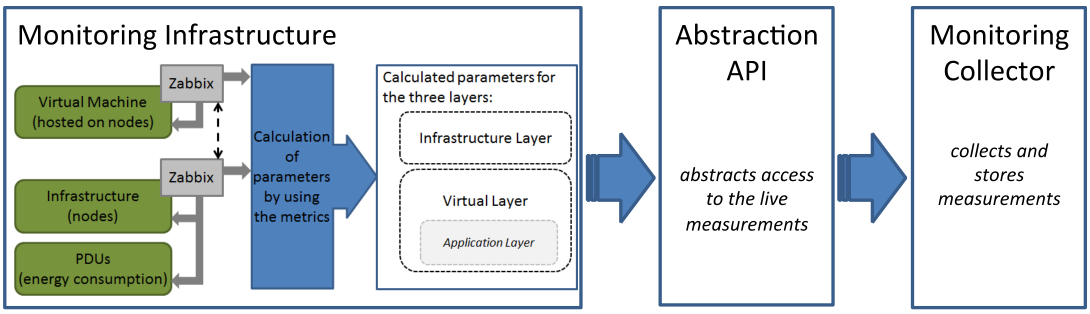

Figure 3: The Monitoring Infrastructure

The *monitoring infrastructure* collects the required monitoring data from the PDUs, physical hosts and VMs. The *abstraction API* facilitates the access to the live measurements so that the *monitoring collector* can collect and store those measurements in the *Accounting database*. Afterwards, the other ECO$_2$Clouds components are enabled to make use of the measurements accessing to such database through a specific API described in the following sections.

4.1 Collecting metrics

The data produced by the infrastructure metrics (defined in Table 1 and Table 2) are stored in a specific Zabbix DB. The Zabbix server communicates with the Zabbix agents of each physical node in the cloud environment as presented in Figure 4 in order to populate this DB. Afterwards, the data are made available through the *abstraction API* as described in Section 4.3.

[5] http://www.zabbix.com/

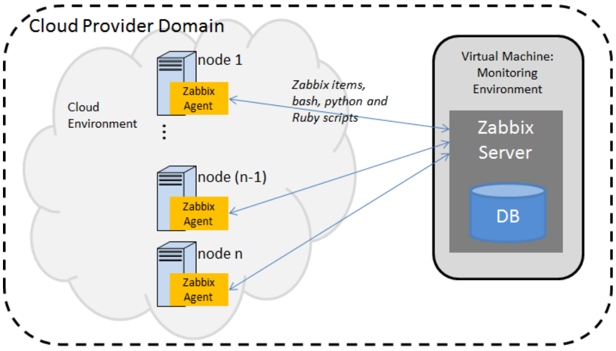

Figure 4: Fetching the monitored Data

The metrics are generated on a defined time interval. Generally, the metric set includes the infrastructure monitoring at site and host level. In addition the energy mix of each provider, the power consumption at node level and, in the cases where providers report, the total power consumption of the used grid are measured. Moreover, the physical hosts of each site are monitored separately, which means that each cloud provider's site metrics can be configured to deal with the individual physical resources.

The metrics for the VMs are based on a dedicated set of infrastructure metrics and customized VM metrics including the power consumption per VM metric. The VM metrics are contextualized automatically when hosting a VM. Zabbix is configured to provide the VM metrics shown on Table 3. As a result, experimenters do not need to perform additional steps for monitoring the VMs.

Finally, beside the infrastructure and the VM layer metrics, experimenters are enabled to monitor their individual applications. These metrics are mainly defined as a guideline: each application needs to be adapted to fulfil the best coverage and guarantee comparable results. Further, these metrics are also captured by an aggregator. Data gathered by these metrics can be used to adapt the application deployment and execution.

4.2 Storage of the Monitored data

In ECO$_2$Clouds the metrics or monitoring data collected from the monitoring infrastructure is stored in a specific database, i.e. the Accounting database. The stored metrics are used to create accurate energy usage information for each physical host and virtual machine on the infrastructure over time. The stored metrics can be mined to identify trends and correlations and can be used to identify opportunities to reduce carbon usage (see Section 4.4).

The ECO$_2$Clouds database schema can accommodate infrastructures which:

- use a single site or span across multiple sites;
- have a single physical host or hundreds of physical resources;
- use a static or dynamic number of metrics.

4.3 Access to metrics

In order to be usable by usable by the ECO$_2$Clouds software and other such client or infrastructure applications, the described metrics need to be available through specific abstraction APIs.

The ECO_2Clouds implementation exploits the *ECO_2Clouds Metrics Abstraction API* that has been implemented on top of BonFIRE to publish the lists of metrics and give access to them (latest value and history) independently from the collection infrastructure used (Zabbix in the ECO_2Clouds case). It therefore serves as a generic layer between the other components of ECO_2Clouds (such as the deployment decision making entity i.e. Scheduler) and the testbeds it uses. This abstraction API can be implemented by other testbeds willing to adopt the ECO_2Clouds solution.

In detail, the API publishes a list of infrastructure metrics, a list of metrics available on each host of the infrastructure and a list of supported metrics for all VMs running on the infrastructure. For each of these metrics, it is possible to retrieve the latest value as stored in the infrastructure or experiment Zabbix servers and to get access to and range of values in the past. The API does not store any information. When the application execution is finished, the abstraction API cannot give access to values that refer to such an execution. This is the role of the Monitoring Collector.

The API follows the REST architecture. Figure 5 lists the ECO_2Clouds API operations.

GET /locations – returns a list of locations.
GET /locations/<location_id>/locationmetrics/aliases – returns a list of infrastructure wide metrics.
GET /locations/<location_id>/locationmetrics/hostmetrics – returns a list of metrics available on all hosts.
GET /locations/<location_id>/hosts – returns a list of hosts in a location.
GET /locations/<location_id>/locationmetrics/aliases/<metric_name> – returns the specified infrastructure wide metric..
GET /locations/<location_id>/hosts/<short_host_name>/hostmetrics/aliases – returns a list of supported metrics for a specific host.
GET /locations/<location_id>/hosts/<short_host_name>/hostmetrics/aliases/<metric_name> – returns the specified metric for a given host.
GET /locations/<location_id>/computes/<compute_id>/hostmetrics/aliases – returns a list of metrics for a specific virtual machine.
GET /locations/<location_id>/computes/<compute_id>/hostmetrics/aliases/<metric_name> – returns the specified metric.

Figure 5 - ECO_2Clouds API operations

4.4 Analysis of metrics

The collection of the metrics gave the possibility to perform some analysis on historical data. In particular, correlation analysis on the collected values has been performed. This is significant for the discovery of relations among indicators. In fact, the set of metrics defined above are not independent and the improvement of the status of an indicator can impact positively/negatively on another one. This knowledge can be the basis for different improvement strategies. In ECO_2Clouds, for example, the correlation between CPU load and energy consumed has been validated. On such correlation, some optimization algorithms have been developed. Another useful finding in data analysis regards energy mixes: they are often characterized by a specific trend. The emission index, for example in UK decreases in the night but between h4.30 and h6.00 its starts to increase and after some fluctuations it decreases between h19.30 and h21.00. Such trends can be exploited to decide (if the users have not time constraints) at which time an experiment should be deployed to minimize the CO_2 emissions. Execution shifting can be therefore a strategy for a greener deployment [6] (see also Section 6.2).

4.5 Carbon footprint estimation by infrastructure providers

The ECO_2Clouds infrastructure providers prototyped a site-based mechanism to estimate the carbon footprint (expressed in g of CO_2) of basic units of resource consumption. This is exposed to the user across three timescales: non-committing, long-term forecast for information of carbon footprint for a VM with specified characteristics; short-term future (e.g. 24 hours), committing quote for a VM with specified characteristics; or past account for a running or deleted VM. This enables users or Cloud brokers such as the ECO_2Clouds Scheduler to rely on that carbon footprint to take scheduling decisions.

The methodology allows each site to develop its own algorithm, for example, our prototype implementation employs a very simple formula to associate CPU utilisation with energy consumption, and uses site monitoring data as per Section **Error! Reference source not found.**, Grid energy mix mentioned in section **Error! Reference source not found.** and results of the metrics analysis discussed on Section **Error! Reference source not found.**. In future, other algorithms and sources of information, like power Grid forecasts can be employed and analysis tools can be developed to allow infrastructure providers to refine their algorithms. Subject to suitable standardisation and regulatory control, this mechanism can also underpin site competition on the basis of CO_2 emission, renewable-energy use and other Green characteristics.

5 Setting up Energy Aware Applications

Cloud applications might be very different from each other and some knowledge about their characteristics can help scheduling activities. For this reason, knowledge about the business process executed by the software application is desirable. This allows, at design time, gathering data dependencies, information flows and any other information that would support the resource allocation.

In this respect, one finding of the ECO_2Clouds project is that in order to decrease the CO_2 emissions and the energy consumption in the cloud environment, it is useful to annotate the application and thus to create an *Application Profile.* Such a profile provides useful information for the deployment phase. It is submitted to the ECO_2Clouds Scheduler on the basis of the analysis of ecological parameters and performance metrics and decides for an initial deployment able to optimize energy efficiency and CO_2 emissions. The more information the Scheduler knows about the application through the Application Profile, the more appropriate the allocation of the resources will be.

On the other hand to understand the real-time behavior of the applications, it is necessary to monitor them and the resources they are using and thus a set of application level metrics have been defined (see Table 4).

In details, the Application Profile includes a set of profile elements, policies and guidelines defined for a particular application. In the application profile we include the following types of metadata:

- *Resource metadata*: they provide information regarding the VMs needed to execute the different application activities (e.g., number and size of virtual machines, and network connectivity, etc.);

- *Flow metadata*: they provide information regarding the business process control flow and thus the dependencies among VMs;
- *Energy and performance requirements*: they refer to energy and performance conditions or constraints within process flows, including eco-metrics such as CO_2 emissions and time constraints;

An ECO_2Clouds user is able to submit an Application Profile through the ECO_2Clouds portal[6] (see Figure 6).

6 Optimization of Energy Consumption

The ECO_2Clouds Scheduler and the Application Controller are the main entities responsible for optimizing the energy consumption of an application. The former uses a decision making process to deploy the VMs on the federated cloud, while the latter manages the applications at run-time.

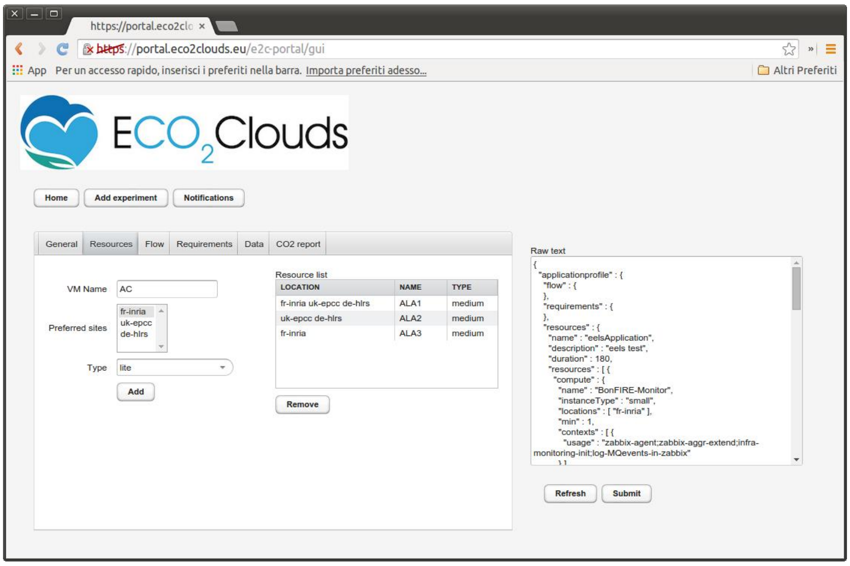

Figure 6 - ECO_2Clouds portal

6.1 ECO_2Clouds Scheduler

The ECO_2Clouds Scheduler is designed as a REST service that can be hosted on a VM within the cloud environment. Keeping in mind the federated cloud infrastructure used in ECO_2Clouds, the Scheduler makes the decision about application deployment at two different levels (i) testbed level, and (ii) physical host level.

6.1.1 Deployment Decision at a Site level

Once a deployment request is received, the first Scheduling step is to select a suitable cloud site. This step is realized by switching the Scheduler in one of the following modes – using a simple REST command:

Individual Deployment performs the selection of a site for each individual VM in the deployment request. In this mode, the VMs in a single deployment request (representing a distributed

[6] http://portal.eco2clouds.eu/e2c-portal/gui

application) may be deployed on different testbeds. For example, after the allocation of a single VM, the suitability of a testbed may change and thus, the next VM in the same deployment request may be allocated to another testbed that fits the suitability criteria.

Bulk Deployment 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 testbed.

The suitability of a site is determined by a combination of multi-criteria optimization (using a weighted sum approach) and load balancing techniques. The multi-criteria optimization prioritises a testbed based on its energy consumption and CO_2 footprint while considering other parameters corresponding to nature and availability of infrastructure resources.

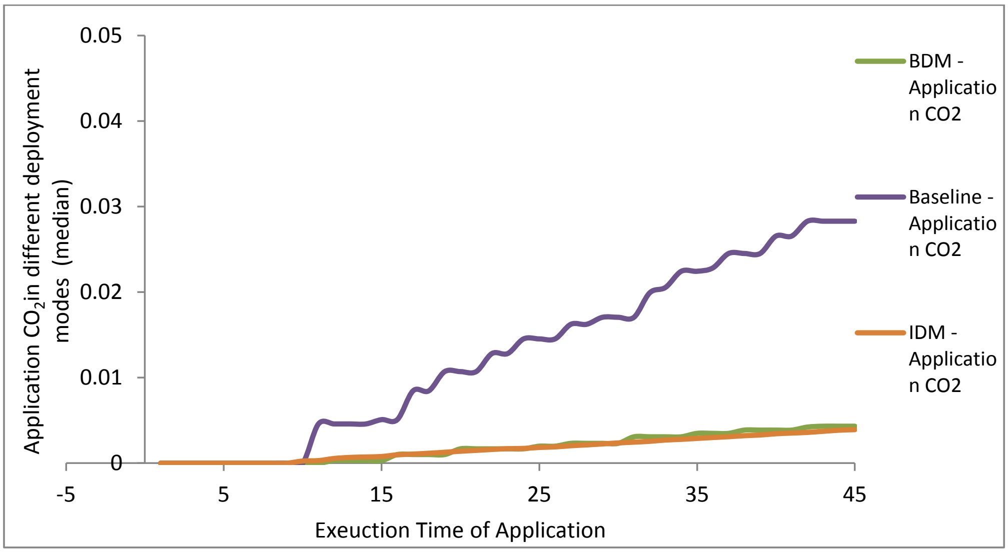

Figure 7: Results from experimental evaluation of site level deployment modes

Effects of the reduction in CO_2 footprint of a sample application when these deployment modes are used instead of the baseline (random deployment) policy on BonFIRE sites is shown in Figure 7

The results in validate the CO_2 aware deployment approach adopted in ECO$_2$Clouds. The CO_2 aware deployment modes in ECO$_2$Clouds (described above) consider the energy mix of all available testbeds, thus selecting a testbed most suitable for the application deployment based on the estimated CO_2 emissions. This approach helps in keeping the application CO_2 footprint at minimum.

6.1.2 Deployment Decision at Physical Host Level

After determining a suitable cloud site the next step for the ECO$_2$Clouds Scheduler is to determine physical host level deployment configuration of new VMs/applications. Since the physical hosts in a testbed use the same energy mix, the optimization potential lies in reducing the overall energy consumption of the physical hosts. Hence, at this stage the objective of ECO$_2$Clouds Scheduler is to reduce the overall energy consumption of the testbed, which can directly contribute towards reducing the CO_2 emissions. In this respect, the optimization mechanism in ECO$_2$Clouds Scheduler tries to *maximize the utilization of individual physical hosts by deploying VMs on high energy consuming hosts* e.g. hosts with highest energy consumption [6].

It should be noted that the above policy was selected as best performer after experimenting with several policies. The unique feature of the above host level deployment policy is the emphasis on energy consumption of physical hosts as the most weighted optimization parameter in the decision making model. Apart from energy consumption, host level deployment considers a mix of resource level parameters e.g. CPU, Memory and number of running VMs to determine suitable hosts for application deployment.

6.2 Application Controller

In addition to the ECO_2Clouds Scheduler, the ECO_2Clouds project provides the Application Controller framework enabling the analysis of the status of the application and the enactment of an adaptation action as required. The adaptation at application level involves actions able to change the behaviour of the application in order to reduce the CO_2 emissions. The actions investigated are as follows:

- Changing load distribution between the active VMs: this action changes the workload among the VMs 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 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 power consumption of the application.
- Application execution time-shifting: this action 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.

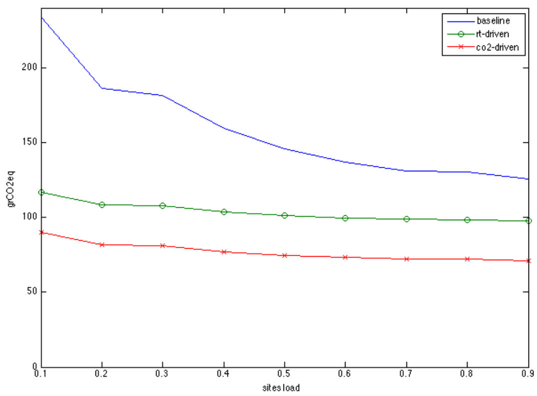

Figure 8 - Effects of the eco-aware workload rearrangement driven by response time and CO_2

Based on the status of the applications, the Application Controller can enact one or more of these adaptation actions in order to reduce the CO_2 emissions. An example of the effect of using the Application Controller is shown in Figure 8 where the CO_2 emissions of an application (baseline line) have been significantly reduced through a workload distribution driven by the response time of the task (rt-driven line) and by the CO_2 emissions of the task (CO_2-driven line). Figure 8 illustrates the results of the application of this strategy on one of the case studies considered in the project (a HPC application). It is possible to notice that the emissions decrease for both of the policies applied.

7 Conclusions

This white paper aims to present the architecture together with the methods and tools that have been used in the ECO_2 Clouds project to improve energy efficiency and decrease CO_2 footprint in federated cloud infrastructures. The experiments performed in the project highlighted that monitoring and collecting knowledge about the execution of the applications enable a “green optimization”. It is important to define a comprehensive set of metrics able to capture all the relevant aspects of the sites, hosts and the deployed applications. Furthermore, the application metrics include some novel indicators (i.e., A-PUE) that are useful to detect waste of energy during the application execution and to understand if adaptation actions are needed. Energy efficiency and CO_2 emissions optimization are also enabled by the adoption of (i) a scheduler that decides where (site and host) to deploy the application in order to guarantee the greenest deployment by continuing satisfy performance requirements and (ii) the Application Controller that analyses the status of the applications and enacts adaptation actions if required. In fact, results show that CO_2 emissions can be decreased by a suitable workload rearrangement, postponing executions to times in which the emission factor is lower and switching off VMs when not needed.

All the methods and tools designed in the ECO_2Clouds project can be applied to other cloud environments. The results achieved in the project can be useful for cloud developers and providers and for adopting a greener approach for the development and deployment of their application able to decrease the CO_2 emissions and improve energy efficiency.

8 References

[1] K. Kavoussanakis; A. Hume; J. Martrat; C. Ragusa; M. Gienger; K. Campowsky; G. Van Seghbroeck; C. Vazquez; C. Velayos; F. Gittler; P. Inglesant; G. Carella; V. Engen.; M. Giertych; G. Landi; D. Margery, "BonFIRE: The Clouds and Services Testbed," Cloud Computing Technology and Science (CloudCom), 2013 IEEE 5th International Conference on , vol.2, no., pp.321,326, 2-5 Dec. 2013.

[2] G. Cook, “How Clean is Your Cloud?,” tech. rep., Greenpeace International, April 2012.

[3] Nlyte Software, “Why PUE alone is not Enough!” Tech. Rep., 2011

[4] D. Azevedo, J. Cooley, M. Patterson, and M. Blackburn, “Data Center Efficiency Metrics: mPUE, Partial PUE, ERE, DCcE,” GreenGrid, Tech. Rep., 2011.

[5] ECO_2Clouds Project. “D3.4 - Realization of an enhanced Monitoring and Data Analysis environment (II)”, September 2014

[6] Cinzia Cappiello, Paco Melia, Barbara Pernici, Pierluigi Plebani, Monica Vitali “Sustainable choices for Cloud Applications: A focus on CO_2 emissions”, Proceedings of ICT4S 2014.Theodore Pertsas, Usman Wajid. “Load Balancing to Save Energy in Cloud Computing”, Energy-efficient systems workshop at ICT4S, Stockholm, Aug. 27, 2014