presented at the Energy Efficient Systems Workshop at ICT4S, Stockholm, Aug. 2014

# Eco-reports in Clouds

Cinzia Cappiello, Barbara Pernici, Pierluigi Plebani, Monica Vitali
Dipartimento di Elettronica, Informazione e Bioingegneria
Politecnico di Milano
Piazza Leonardo da Vinci 32, 20133 Milano
Email: [name.surname]@polimi.it

***Abstract*—The present paper discusses the issues arising from the need of presenting to the user information about the environmental impact of their applications in a cloud environment. Several aspects have to be considered: to make aware the users of the energy consumption and environmental impact of their applications, but also to make them aware about the actions taken in the environment with the goal of satisfying non functional requirements and optimizing the energy consumption and environmental impact of applications.**

## I. Introduction

Energy consumption and the environmental impact of applications running in clouds are being studied from different points of view. In the $ECO_2$Clouds project[1] the focus is on providing an optimized environment, taking actions during the execution of applications in the clouds, both at the infrastructural and at the application level.

In this short paper, we focus on how to present the information to the users about the energy consumed and the environmental impact expressed in terms $CO_2$ emissions. The ability of showing to the users the consumption of their applications derives from an existing monitoring infrastructure and the availability of information about the energy mix in the sites where the clouds infrastructure is available. This problem has a general nature, in fact, it is a possible case of the more general case of presenting to users the environmental impact of applications and also of showing them how this impact can be reduced and which actions are taken, or can possibly be taken, to achieve this goal. In their seminal paper [1] Watson and Boudreau discuss the implications of monitoring and controlling energy consumption in controllable systems, with the goal of reducing energy consumption. The problem has a general nature and possible solutions valid in a given field, could be applied also in other fields. In the present paper we focus on presenting energy consumption and environmental impact in so called eco-reports, containing useful information for the users to assess their environmental impact, to understand optimization actions undertaken by the system they are using, and to take decisions on the basis of this information. The approach is based on the $ECO_2$Clouds project [2], which focuses on environmental impact of federated clouds and provides a monitoring infrastructure for assessing eco-metrics [3] and possible for reducing the $CO_2$ emissions.

In the following sections, we first present the $ECO_2$Clouds approach for presenting environmental impact of applications in Section 2, then we discuss awareness reports and adaptivity reports in Sections 3 and 4 respectively.

[1] eco2clouds.eu

## II. Architecture and Infrastructure Requirements

In the $ECO_2$Clouds project the goal is to improve energy efficiency and to reduce the environmental impact of applications. Eco-reporting is achieved in two ways:

- *Awareness reporting*: showing the users information about the execution of their applications, including values of eco-metrics.
- *Adaptivity reporting*: describing how the platform applied adaptivity actions to improve the energy efficiency and reduce the environmental impact of applications, showing for each decision taken by the system the motivations for the actions.

In Fig. 1 we show the architecture that enables this approach. This general architecture shows that the reporting is based on the interaction with a cloud ecosystem which is providing monitored eco-metrics (left arrow) and information about adaptivity actions (right arrow). The latter includes not only information about the action itself, but also motivations for taking it in a given context. In order to provide environmental impact information, the cloud ecosystem must also be based on information given by energy providers about the current energy mix being provided. Such a general architecture can be adopted also in other contexts, thus not being limited to cloud computing, but it can be generalized to all those cases in which the reduction of the environmental impact is a combination of automatic actions performed by an adaptive and optimizing infrastructure and human decisions, based on the awareness on energy consumption given to the users by reports on eco-metrics.

## III. Awareness Report

This section focuses on how the data to be presented to the user can be obtained.

### A. Environmental impact of the cloud site

In the cloud scenario, one of the important choices for the deployment of an application is the selection of the cloud site. Such decision usually takes into consideration factors as the state of the machines and their average performance. In order to decrease the environmental impact of the applications, we also consider energy-related indexes such the percentage of

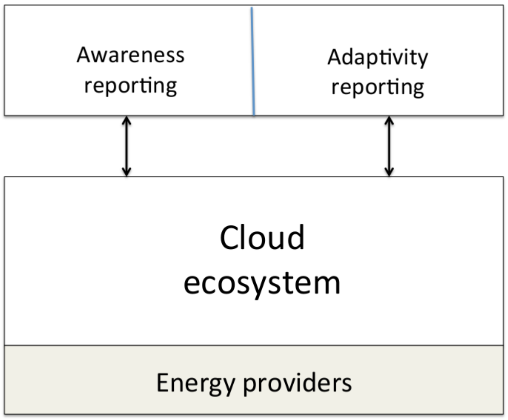


Fig. 1. Presenting the environmental impact

green energy sources used by the site and the carbon emission factors. In order to retrieve such data, it is fundamental to know the energy mix that the sites are using, that is the type of the exploited energy sources together with the average percentage of provided power. In fact, the evaluation of the $CO_2$ emissions is based on the emission factors $ef$ (gCO2e/kWh) that depend on the type of considered energy source. If the $i-th$ cloud site is powered by more than one energy source, the emission factor $ef_i$ results:

$$ef_i = \sum_{s \in sources} ef_s \cdot es_{s,i}$$

where $ef_s$ is the emission factor related to the $s-th$ energy source while $es_{s,i}$ is the percentage of utilization of the $s-th$ energy source.

The energy mix (and thus the percentages $es_{s,i}$) can be easily determined if the data center is powered by dedicated energy sources while it might be difficult to be calculated if the data center is powered by the national grid. In this latter case, we have to find information about the national energy provision. We might rely on public documents that periodically provide national average emission factors or on the web sites of some countries (e.g., France and UK) that provide the real time energy mix. Accessing accurate and real time values allows us to perform a more useful and comprehensive assessment and analysis of the cloud site $CO_2$ emissions. The availability of historical data can be exploited in order to identify regular and/or seasonal patterns that can be used in the deployment of applications.

### *B. Computing the CO2 emissions*

Given the emission factor of each site in a given moment, the computation of the $CO_2$ emissions related to the execution of an application running in a cloud environment can be obtained by the following formula:

$$CO_2e = \sum_{i \in site} \cdot \int_t ef_i(t) \cdot P^{app}(t) \cdot dt$$

where $P^{app}$ represents the power consumed by an application obtained as the sum of power consumed by the $j-th$ virtual machines assigned to the application and running on the host $k-th$ installed in site $i-th$:

$$P^{app}(t) = \sum_{i \in site} \sum_{k \in hosts} P^{vm}_{j,k,i}(t)$$

Starting from this premise, how to compute the power required by an application becomes the central and most problematic point. Indeed, the only place where the power can be physically measured is at host level where many applications are running. At the higher level, i.e., virtualization, the application can be identified as the user of some VMs but the power of such VMs can be only estimated. Moreover, as could-based applications might run on different VMs that, in turn, involve several physical host the computation of the power needs to take care of the possible heterogeneity among the different severs installed on the sites.

The approach proposed in this paper starts from the assumption that a positive correlation between the CPU load and the power consumption exists at both physical and virtualization level, as also demonstrated in [4]. Based on this, we monitor the CPU load and the power consumption of each $k-th$ host in a site $i-th$, i.e., $C^h_{k,i}$ and $P^h_{k.i}$. This data are measured and not computed so they are reliable by definition. Collecting the values of $C^h_{k,i}$ and $P^h_{k.i}$ allows us to identify the relationship of these two variables. An important aspect concerns the min and max values that these two variables can assume. Indeed, while $C^h_{k,i} \in [0..100\%]$, the $P^h_{k,i} = [min, max]$ where $min \in \mathbb{R}^+_0, maxmin \in \mathbb{R}^+_0, min \leq max$ and the range vary from host to host as it depends on the architecture and the hardware components. On this basis, $P^h_{k,i}$ can be divided into two components:

$$P^h_{k,i}(t) = Pidle^h_{k,i} + Pexec^h_{k,i}(t)$$

where the first element represents the power consumed by the when $C^h_{k,i} \simeq 0$, i.e., the host is inactive. The second element varies during the time based on the CPU load according to a function $f$ that captures the dependency between the CPU load and the power based on the analysis of the monitoring values.

Moving to the virtualization layer, we can measure, for each $j-th$ VM (deployed on a $k-th$ host) used by the application, the CPU load $C^{vm}_{j,k,i}$. This can be done looking to either the system tools inside the VMs (e.g. top command) or the monitoring system provided by the hypervisor. Finally, the power of a VM is given by the following formula:

$$P^{vm}_{j,k,i} = \frac{Pidle^h_{k,i}}{\#VM_k} \cdot f(Pexec^h_{k,i}, C^{vm}_{j,k,i})$$

It has to be noted that the awareness module can help the users assessing not only the environmental impact of the applications as they are running, but also to support evaluations of possible shifts in time of the execution of the application. In fact, assuming that the energy computed by the VMs is available, the total value of emissions can change depending on when the application is started, since the emission factor of a site is variable in time. The users can thus assess if it is

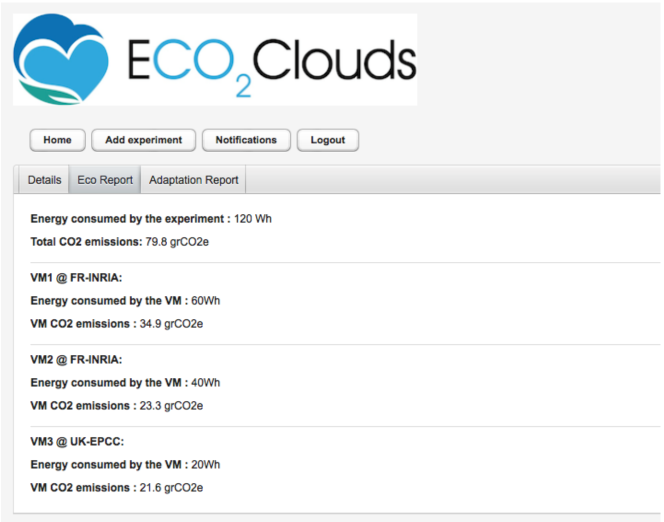


Fig. 2. The Eco Report section of the Final Analysis Report

convenient to delay the execution of an application to exploit reduced emission factors at other times of the day.

## IV. Adaptation Report and the Final Analysis Report

In this section we discuss the Final Analysis Report, which is an instrument provided to the user allowing a detailed analysis of his experiments. The aim of this report is to show to the user some information about the behavior of the terminated experiments.

At the end of an experiment the user can access the Final Analysis Report, showing a summary of the behavior of the experiment. This summary is divided into three sections: the "Details" section, the "Eco Report" section, and the "Adaptation Report" section.

The "Details" section includes a general summary of the experiment consisting in some general information. Information consists in the ID of the experiment, start time of the experiment which consists in the moment when the experiment has started, and end time, which is the time stamp when the experiment has been completed. Finally, the last part contains also the list of the VMs involved in the experiment with their IDs.

The second section is the "Eco Report". This section gives an overview of the behavior of the experiment, giving some general information about the sustainability of the experiment in terms of energy consumption and $CO_2$ emissions. This information is given for the whole experiment with e general value aggregating the impact of all the components of the experiment involved in $CO_2$ emissions, and for each VM, from the creation to the expiration. This first analysis is useful to make the user aware in an immediate and easy to understand way of the impact of his experiments. An example of the "Eco Report" is shown in Fig. 2.

During the execution of an experiment, some aspects of the system can misbehave decreasing the performance of the experiment or increasing the environmental impact of its execution. The system administrator can perform some adaptation

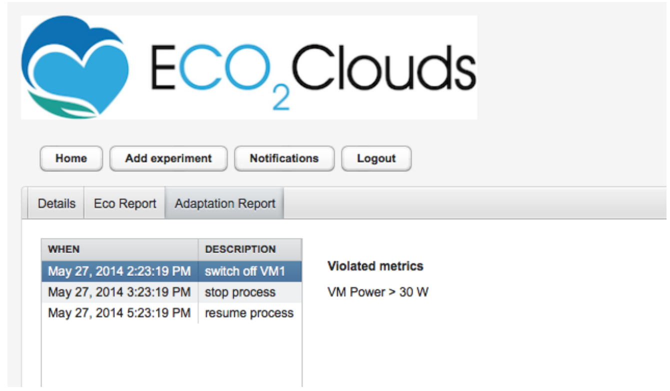


Fig. 3. The Adaptation Report section of the Final Analysis Report

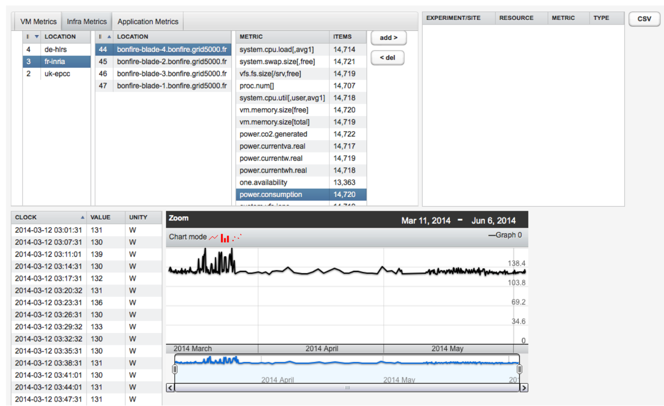


Fig. 4. The Metric Report is a detailed view of the Final Analysis Report

actions in order to fix these misbehavior. The last section of the Final Analysis Report, the "Adaptation Report", contains some information about what has been done for improving the state of the system, during the lifetime of the whole experiment. The report consists in a list of adaptation actions with the time stamp of enactment and a description. Selecting one of the actions in the list, the user can have additional information about the triggers of the selected adaptation action. Triggers are usually the violation of the constraints about the values that each metric can take. An example of the Adaptation Report section is shown in Fig. 3, where three actions has been enacted during the execution of the experiment, and the first action (switch off of a VM) has been executed to reduce the power consumption of the experiment.

The user can also access to a more detailed information about his experiments, called "Metric Report". Each experiment is monitored using a monitoring system collecting information about several aspects of its behavior. The collected metrics can be directly or indirectly related to the efficiency of the experiment, but also to the quality of service, which is an important information for the user. The user can analyze one by one the collected metrics to evaluate the behavior of its experiment and to make considerations about improvements and future deployments. An example of the Metric Report can be observed in Fig. 4. Information is shown at three granularity levels: (i) the infrastructure level; (ii) the virtual machine level; and (iii) the application level.

At the infrastructure level, the user can select a site and then one of the hosts in the site. Selecting the host, the list of collected metrics for the host can be seen. When clicking on one of the metrics, the user can read the list of the collected values and a graphical representation of the trend of the metric. The user can also select a time slot using the zoom at the bottom of the graph. Examples of metrics at the infrastructure level are resources utilization (CPU, memory, and IOPS), number of running VM, and power consumption of the host.

The VM level shows similar metrics but for the VMs involved in the experiment. From the list of VMs, the user can select a VM and can observe the behavior of the metrics collected for it. Example of metrics are resources utilization (CPU, memory, and IOPS) and power consumption from the VM perspective.

The last level, the application, contains metrics collected to measure the environmental impact and the performance of the application. The user can see both metrics aggregated for the whole application and for the specific activities of the application. Example of metrics are response time (time to serve a request), throughput (number of transactions executed), and application performance (ratio between the number of transactions executed and the energy consumed).

Information contained in the Final Report Analysis is necessary to make the user aware of the ecological impact of his experiments. Moreover, the user can use this information to take informed decision about how to deploy and set further executions of his experiment in order to make them more sustainable or more efficient.

## V. Concluding Remarks

The present paper presents a general approach to visualize energy and environmental information for applications in eco-reports. The approach distinguishes between awareness information and information about the actions that have been performed by the system to keep it compliant with the given requirements. The approach has been developed for a federated cloud system, however its eco-reports can be applied in more general settings, in which the system is able to report on eco-metrics and is able to provide an active behaviour to maintain given constraints.

## Acknowledgments

This work has been supported by the $ECO_2$Clouds project (http://eco2clouds.eu/) and has been partly funded by the European Commission's IST activity of the 7th Framework Program under contract number 318048.

## References

[1] R. T. Watson, M.-C. Boudreau, and A. J. Chen, "Information systems and environmentally sustainable development: Energy informatics and new directions for the is community," *MIS Quarterly*, vol. 34, no. 1, pp. 23–38, 2010.

[2] U. Wajid, B. Pernici, and G. Francis, "Energy efficient and CO2 aware cloud computing: Requirements and case study," in *IEEE SMC*, pp. 121–126, 2013.

[3] C. Cappiello, S. Datre, M. Fugini, P. Melia, B. Pernici, P. Plebani, M. Gienger, and A. Tenschert, "Monitoring and assessing energy consumption and CO2 emissions in cloud-based systems," in *IEEE SMC*, pp. 127–132, 2013.

[4] A. Verma, G. Dasgupta, T. K. Nayak, P. De, and R. Kothari, "Server workload analysis for power minimization using consolidation," in *Proceedings of the 2009 conference on USENIX Annual technical conference*, USENIX'09, (Berkeley, CA, USA), pp. 28–28, USENIX Association, 2009.