Journal of Network and Computer Applications 78 (2017) 83–96

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

E-eco: Performance-aware energy-efficient cloud data center orchestration

Fábio D. Rossi[a,*], Miguel G. Xavier[a], César A.F. De Rose[a], Rodrigo N. Calheiros[b], Rajkumar Buyya[b]

[a] *Faculty of Informatics, Pontifical Catholic University of Rio Grande do Sul, Brazil*

[b] *Cloud Computing and Distributed Systems (CLOUDS) Laboratory, Department of Computing and Information Systems, The University of Melbourne, Australia*

ARTICLE INFO

Keywords:
Cloud application performance
Data center
Energy-efficient management.

ABSTRACT

The high energy consumption of data centers has been a recurring issue in recent research. In cloud environments, several techniques are being used that aim for energy efficiency, ranging from scaling the processors frequency, to the use of sleep states during idle periods and the consolidation of virtual machines. Although these techniques enable a reduction in power consumption, they usually impact application performance. In this paper, we present an orchestration of different energy-savings techniques in order to improve the trade-off between energy consumption and application performance. To this end, we implemented the Energy-Efficient Cloud Orchestrator - e-eco - a management system that acts along with the cloud load balancer deciding which technique to apply during execution. To evaluate e-eco, tests were carried out in a real environment using scale-out applications on a dynamic cloud infrastructure, taking into account transactions per second as a performance metric. In addition to the empirical experiments, we also analyzed the scalability of our approach with an enhanced version of the CloudSim simulator. Results of our evaluations demonstrated that e-eco is able to reduce energy consumption up to 25% compared to power-agnostic approaches at a cost of only 6% of extra SLA violations. When compared to existing power-aware approaches, e-eco achieved the best trade-off between performance and energy-savings. These results showed that our orchestration approach showed a better balance in regard to a more energy-efficient data center with smaller impact on application performance when compared with other works presented in the literature.

1. Introduction

Cloud computing offers access to data, computation, and applications as utility services from anywhere through the Internet. Customers are not tied to a physical infrastructure because their data and applications are accessed through services. In addition, due to the pay-per-use model, cloud customers only pay for what they consume, without overspending and incurring unnecessary consumption of resources. Factors such as reliability, security, availability, fault tolerance, scalability, and sustainability made cloud computing a de facto standard in the industry (Buyya et al., 2011). This paradigm allows companies to focus more on innovation, without worrying about issues such as hardware acquisition or maintenance of services.

Nevertheless, with the migration of applications to cloud environments, data centers began to increase the amount of resources available to meet this new demand. Although the adjustment in the amount of data center resources is a necessary process, the greater utilization of resources requires a greater amount of energy to keep them active, which impacts sustainability (NRDC, 2014).

Power consumption and heat dissipated by computing equipment boost the emission of gases that cause the greenhouse effect, and entail phenomena such as droughts, floods and rising temperatures. One solution to reduce the heat dissipated by cloud data centers consists of powerful cooling systems, which in turn impact on electricity consumption. With the increase in the number of devices present in data centers in addition to the need for cooling these devices, power consumption has become not only an environmental, but also an economic issue. Furthermore, the increased processing power has only been possible with the increase in energy consumption by individual servers in the data center. It seems to be logical that energy savings imply in the reduction of performance of computing servers. Substantial performance loss occurs during power management and the standard deviation tends to exceed the mean in the measured data points (Cameron, 2014).

Still, Cameron (2014) argues that the performance loss is a recurrent problem that requires a great effort to be addressed. In

[*] Corresponding author.
E-mail address: fabio.diniz@acad.pucrs.br (F.D. Rossi).

http://dx.doi.org/10.1016/j.jnca.2016.10.024
Received 26 January 2016; Received in revised form 9 June 2016; Accepted 28 October 2016
Available online 16 November 2016

addition, data center operators can lose their jobs due to a substantial performance loss at a critical time. Thus, power management is often neglected in data centers to prevent power-saving features from compromise the performance of the servers. In this context, reducing energy consumption is a challenging task, since cloud providers must support the growing demands and maintain the performance expected by customers.

Understanding of the impact of energy savings options in large-scale cloud environments on data centers is one of the most studied topics today (Rossi et al., 2014; Freeh et al., 2007; Berl et al., 2010). Improved management of resources has the potential to reduce energy consumption, and consequently reduce the emission of heat, reducing cooling effort on the equipment, and reducing the emission of carbon dioxide in the atmosphere. In this way, several works are being developed using energy-saving techniques supported by current hardware and operating systems (Grover, 2003). These techniques range from the use of sleep states (Zhu et al., 2012), reducing the processors frequency (Eyerman and Eeckhout, 2011), and the placing of virtual machines (VMs) on available resources (Beloglazov and Buyya, 2012). However, decisions on when the processor should operate on its low frequency, when VMs should be moved through the network, or when power states should be modified may impact on the applications performance because of the reduction of the number of instructions that can be performed or because of the time spent performing these complex operations.

At the same time, applications performance metrics, such as response time, transactions per second, or any other metric that can be established through Service Level Agreements (SLA) became a competitive factor in the cloud service provider's perspective. On the customer side, these metrics directly impact the quality of experience (QoE) (Schatz et al., 2013), in other words, it implies the customer's impression on the compliance with their requests to the service.

Based on these facts, the management of power saving techniques incurring the minimal possible impact on applications performance introduces a new challenge. This paper proposes the Energy-Efficient Cloud Orchestrator (e-eco), an orchestrator of energy-saving techniques on a cloud environment that aims to improve the trade-off between power savings and applications performance. It allows on-the-fly management of what techniques should be applied based on the application behavior, reducing the impact of these techniques on application performance. The three main contributions of this paper are summarized as follows:

- An improved model of ACPI power-saving states including transition time and energy consumption;
- A detailed analysis of the impact of server consolidation and DVFS techniques on energy consumption and application performance;
- The proposal and implementation of an energy-efficient performance-aware cloud orchestrator.

Our results show that e-eco reaches energy saving rates very close to other proposals in the literature, with the advantage of far less impact on the applications performance, when compared to most of the previous works. The direct value driven by this work is the improvement of the trade-off between energy savings and applications performance in cloud environments. Beyond this immediate contribution, an expected impact of this work regards indirect benefits such as reduction in cooling cost and reduction of carbon dioxide emission in the atmosphere.

The rest of this paper is structured as follows. Section 2 builds up on background concepts and technologies used in this paper and drives the motivation for this work by uncovering open problems and discussing value potential. The problem statement is discussed in the Section 3. Section 4 presents our Energy-Efficient Cloud Orchestrator system. We present a quantitative evaluation of the benefit of our system to real and simulated workloads in Section 5. We review and put our work in the context of related work in the Section 6, and conclude with summary and elevation of the findings in Section 7.

2. Background

This section presents the technical background that underlies the development of the proposed e-eco system. Initially, we explain the ACPI specification, which allows the transition between different energy states and features aimed at energy management in modern operating systems. Afterwards, we present the energy savings opportunities provided by the virtualization layer, one of the building blocks of cloud computing infrastructures.

2.1. ACPI support for energy savings

The Advanced Configuration Power Interface (ACPI) (Grover, 2003) consists of an open standard for power management in modern operating systems (OS). ACPI is designed to allow the OS to control each hardware component. Before ACPI was developed, power management was performed by Plug and Play (PnP) and Advanced Power Management (APM) subsystems, which are implemented in hardware, and thus are less flexible regarding management capabilities. ACPI, on the other hand, is implemented in the OS layer, and thus provides greater flexibility for management of components, in addition to being independent of hardware (once the hardware supports the ACPI standard).

Through the ACPI, the OS has the ability to drive specific hardware devices to a low power consumption state when these devices are not in use. Similarly, when the OS detects that the applications do not require a large amount of resources, ACPI can direct the whole environment to a low power consumption state. In addition to offering several states with different levels of energy consumption, ACPI also controls the transition between these states. From a user-level perspective, the OS can be thought of as being in one of the states depicted in Fig. 1. ACPI specifies different levels of states, which are: global states, sleep states, device states, and processor states. The ACPI specification, available on newer systems is a expansion of the APM, which allows operating systems to control in greater detail the power management of various components. ACPI allows placing processors and motherboard components in different energy consumption (sleep) levels as needed. Global States (Gx) reflect the user perception of the machine. However, different levels of power saving granularity can be achieved from the G-States. Arrows represent the possible level transitions.

2.1.1. Sleep states

The global states (Fig. 1 (a)) denote the entire system. Sleep states (Fig. 1 (b)) consist of states derived from the global states (G1). The power states of a particular device (Fig. 1 (c)) are usually not visible to the user. Devices may be turned off while the system keeps working, for instance. Processor states (Fig. 1 (d)) are states of power consumption within the overall working state (G0).

Although sleep states can save energy, their transitions leads to

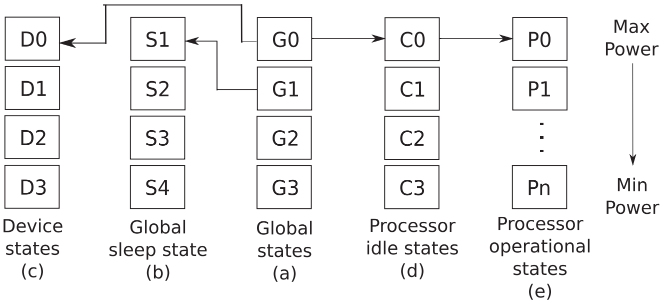


Fig. 1. The ACPI specification.

Table 1
Suspension States.

Level	Description
Global states	
G2	the system consumes a minimal amount of power, user mode threads and system processes are not running, and the system context is not saved (Soft Off)
Sleep states	
S3	fans, hard drives and other devices are stopped. The OS context is kept in RAM, allowing a quick return to a ready state when necessary (sleep)

overhead. The latency of the transition from one state to another may take long enough to negatively impact the application performance. The deeper in energy savings is the state, the greater the latency of transition between one state and another. Based on this, the choice of the ideal states for each environment should be a wise decision, if the intention is to improve the trade-off between energy savings and performance.

In particular, e-eco focuses in the S3 state (commercially called standby) and G2 (soft off). An explanation of each of these states is presented in Table 1.

2.1.2. Reducing the processors frequency

Besides these mentioned states, Dynamic Voltage and Frequency Scaling (DVFS) (Kolpe et al., 2011) (Pn) is the name given by the industry to P-States (Fig. 1 (e)). Each level denotes one of all available modern processors' frequencies which, in conjunction with the ACPI-based firmware, allows adjustment on-the-fly based on the CPU load. Description of these states is provided in Table 2.

Transitions between sleep states cause overhead on the performance of cloud applications. Furthermore, processor operational states also present some limitations influencing the performance of the applications. When a state changes from P0 to P1, the processor frequency is reduced, reducing its voltage. Consequently, the number of instructions that the processor performs is also reduced proportionately. Although both suspension and processor states provide energy savings, the main limitation of the former is the time taken for transitions between the system working and the power-saving state. In deeper shutdown states such as G3, the time taken for the host to become ready to receive a new demand can be large.

On the other hand, reduction in the processors frequency also reduces the amount of instructions that the processor performs in a given period, increasing the execution time of applications. For this reason, a strategy that can balance energy saving and application performance loss is required.

2.2. Vrtualization support for energy savings

Another possibility of energy saving in cloud environments is to take advantage of the features that the virtualization layer provides (Bhuiyan and Wang, 2013), as we can see in Fig. 2. The scenario describes an underutilized virtualized environment, where there is the possibility of using some resource-saving technique in order to reduce operating costs. The VM consolidation is applied to the virtualized environment shown above. With the VMs consolidation on fewer resources, resources that were idle can be turned off. In non-virtualized environments, there was the thought that only one service should be executed per physical server. This ensured greater security and greater availability of services, since the failure of a server only affected one service and the vulnerability of a service only exposed one server. However, the utilization of hardware resources by a server is typically extremely low, except for short periods when utilization reached its peak. As resources needed to be provisioned for supporting such rare peaks, this approach resulted in under-utilization of resources.

Table 2
Processor States.

Level	Description
Processor operational states	
P0	maximum processor performance capability and may consume maximum power
P1	the processor performance capability is limited below its maximum and consumes less than maximum power
Pn	the processor performance capability is at its minimum level and consumes minimal power while remaining in an active state

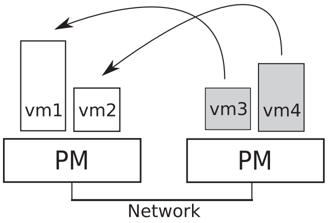


Fig. 2. Underutilized virtualized environment.

Server consolidation (He et al., 2011) is a technique that utilizes one physical machine to host multiple isolated virtual machines, each performing the work of a server. This approach ensures isolation of servers and has the advantage of increasing the utilization of physical machine, which reduces operating costs, creates more flexible environments, and reduces IT administration costs. The most important result of server consolidation is the best use of resources, as if there are n servers with a utilization rate α, it is less costly and more advantageous to consolidate the n servers in a single physical machine, with $n \times \alpha$ utilization rate, provided that $n \times \alpha < 100\%$.

For a better understanding of this process, Fig. 2, depicts a virtualized environment where each physical machine (PM) supports two virtual machines (VMs). If both PMs are underutilized, from the perspective of energy savings, it would be more beneficial to migrate all VMs from the second PM to first, and turning off the second one.

VM consolidation allows reduction of physical space used by servers, as they are now just a physical machine for several virtual servers. This provides lower electricity costs and maintenance machines, since the number of physical machines utilized is reduced. However, the time spent to transfer a VM allocated on one host to another generates an overhead on application performance.

3. Problem statement and motivation

This section presents preliminary assessments carried out to drive decision-making process of e-eco for energy saving and performance trade-off. These experiments have two aims: (i) identify the cost, in terms of time and energy of state transitions; and (ii) quantify the impact of VM consolidation and DVFS on the execution time of applications.

3.1. Modeling power-saving transitions

As cloud data centers are typically very large scale, it is likely there will be idle hosts most of the time. These energy-agnostic environments waste resources, consume more energy than would be necessary, increase operating costs with cooling, and impact negatively on issues of sustainability, such as heat emission and harmful gases to the environment. The state transitions performed by servers on such environments that do not target energy savings can be seen in Fig. 3. Power-agnostic transitions, where there are only two states: running application (busy) and deallocated (idle). In both cases, energy consumption could be reduced, either by reducing the frequency at host

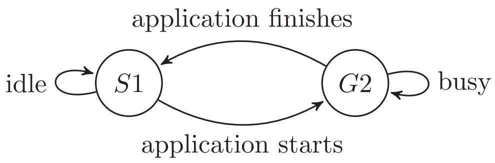


Fig. 3. Power-agnostic transitions.

running (in the case of underutilization) or by the use of a sleep states on the idle host. However, both options incur performance losses.

In the ACPI specification, the idle state corresponds to S1. It turns off the screen, the hard drive, and the fans. Although all executing programs are kept stored in RAM, the memory remains active, requiring little power for maintaining user data until some external event occurs and turn the subsystems back on. The advantage of this state is the short time required for the host to be reactivated. This is fundamental in situations where the machine must react to all possible events or become available very quickly. As the context of the operating system is stored in a volatile memory that requires power to keep up the data, there is a disadvantage when instabilities occur in the power grid.

To explore new states that can be used to replace the idle state, we perform evaluations in energy consumption in each sleep state. The tests were conducted in one host equipped with Intel Xeon processors E5645 2.40 GHz, 12/24 cores, 64 MB L3 Cache, with Ubuntu Linux 12.04 LTS - Kernel version 3.13.0. As the focus of our work is to improve the trade-off between energy savings and applications performance, we also evaluated the time required to perform each state transition and the potential impact of such time in the application performance. Results of these tests are depicted in Fig. 4.

The problem addressed by our work is to decide which states or which set of states can save more energy with less impact on performance metrics of applications. There are disadvantages in the use of all available states, therefore, to develop our proposal, we choose the best states to balance performance with power savings.

A better understanding of the overheads among the transitions of the energy saving states enables the development of more efficient strategies optimized for different computing scenarios. Our previous work (Deng et al., 2013) presents an energy-efficient strategy for HPC clusters and served as a starting point for this proposal, now focused on cloud data centers. Therefore, based on the previously performed evaluations, we propose the energy-aware transitions graph presented in Fig. 5. This model was used in the implementation of e-eco, in order to improve the trade-off between energy savings and applications performance. Based on this model, we present the implementation of e-eco in order to improve the trade-off between energy savings and applications performance.

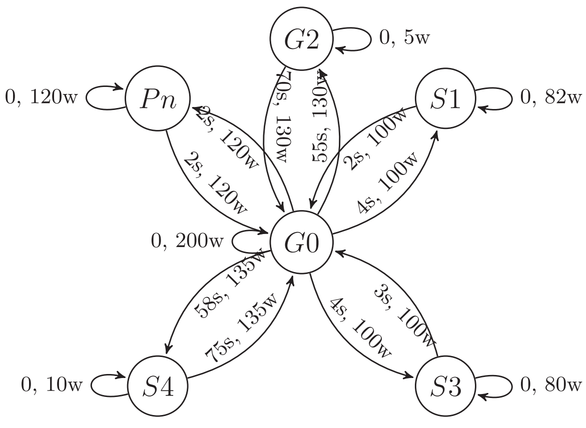


Fig. 4. ACPI transition time and energy consumption.

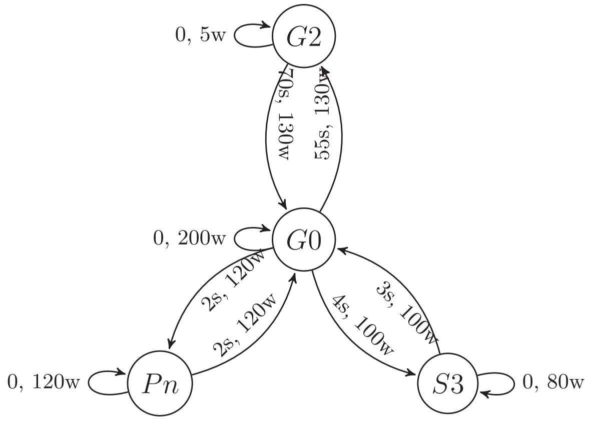


Fig. 5. ACPI transitions chosen for the e-eco.

3.2. *Quantifying the impact of server consolidation and dvfs*

Virtualization technology, which is embedded in most modern computing infrastructures, offers an opportunity for energy savings through server consolidation. Besides, new CPU architectures allow the reduction of the processor frequency with the same purpose. An open question in the area of cloud computing concerns the quantification of energy savings and application performance impact of both these techniques.

To this end, we performed experiments that quantifies how each technique contributes to the trade-off between performance and energy savings. Tests were carried out on an infrastructure with 2 servers and a shared storage (PowerVault MD3200i SAN Storage Array) configured with 1 Gigabyte iSCSI as the communication protocol between storage and the host. To range the size of VMs, we used as a basis the VM sizes offered by Amazon Web Services, as described in Table 3.

As the application load on each VM, we used a stress test. The energy consumption was obtained using two power meters connected between the power source and the servers. These devices (Accuracy: ± 0.3%+2D) have a USB connection that allows periodic external reading and storage of the measurement.

Tests were carried out taking into account two scenarios, in which initially there are two VMs over two hosts (Fig. 6 - (a) Energy-Agnostic Scenario with two underutilized PMs using 100% of processor frequency, and Two Energy approaches for energy savings in cloud data centers: (b) VM Consolidation, turning off the idle PM, and (c) Reduction of the frequency of both PMs). In the first scenario, one of the VMs is consolidated in another host, making the idle host to be turned off in order to save energy. This scenario can be divided into more sub-scenarios (Fig. 7 - (a) VM Consolidation through point-to-point network, and (b) VM Consolidation through centralized images storage), depending on the location of the images that support the VMs: on the PMs or in a centralized storage. Another factor evaluated was the communication between the hosts and the storage via iSCSI, or if there is no centralized storage, the communication either point-to-point hosts using Gigabit Ethernet or Infiniband. In the second scenario, VMs remains on the hosts, but the frequency of the processor of each host is dynamically adjusted via DVFS.

Power consumption measurements for each scenario were per-

Table 3
Evaluation VM characteristics.

Type	*CPUs*	*Memory*	*Image Disk*
Tiny	1 CPU	1 Gb RAM	1 Gb
Small	1 CPU	2 Gb RAM	4 Gb
Medium	2 CPUs	4 Gb RAM	8 Gb
Large	2 CPUs	8 Gb RAM	16 Gb

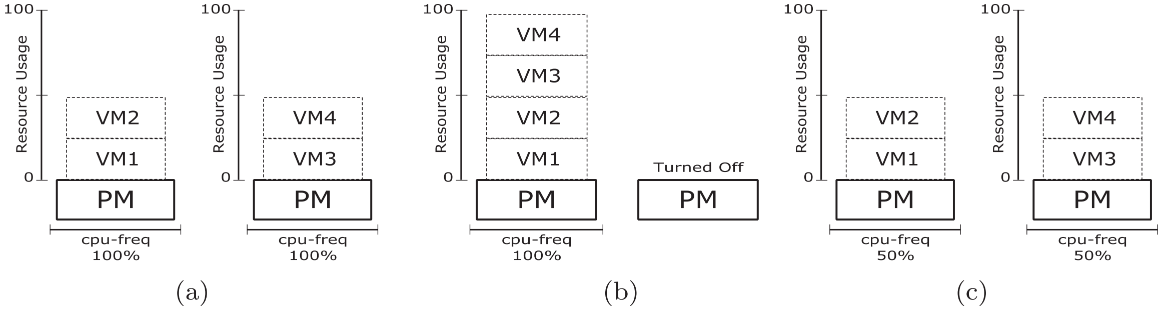


Fig. 6. (a) Energy-agnostic scenarios.

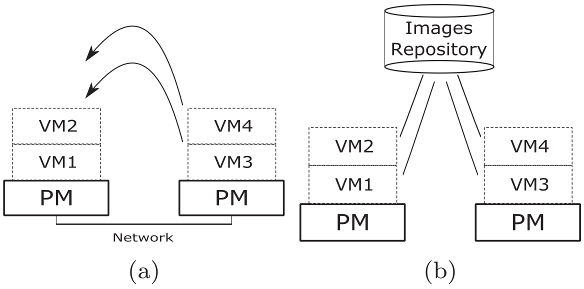


Fig. 7. Proposed techniques evaluated in the paper.

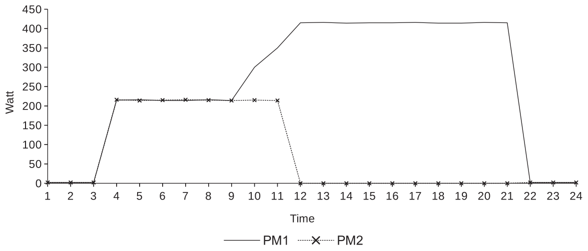


Fig. 8. Example of VM consolidation.

formed at two different times. Fig. 8 (performance and energy-efficiency evaluation of each evaluated VM storage alternative - A: VM consolidation through the storage, B: VM consolidation through the network, C: DVFS on 2 hosts without VM consolidation) shows the behavior of the majority of the tests and supports the understanding of the results presented in Tables 4 and 5. In this Figure, two hosts, PM1 and PM2, running one VM each, start their execution in t=3 s. As both hosts are underutilized, between t=9 s and t=12 s, the process of consolidation is triggered, where the VM allocated in PM2 migrates to the PM1. Consequently, PM1 increases its use of resources now with 2 VMs, and PM2 becomes idle and is turned off. Thus, in t=12 s power consumption is measured only in PM1.

Table 4 shows measurements performed during the VM consolidation (in Fig. 8, the interval between t=9 s and t=12 s). Tests were conducted in four separated sets: the first set (tests 1–4) consists of VM consolidation through a shared storage via iSCSI protocol; the second set (5–8) consists of VM consolidation through point-to-point Gigabit Ehernet; the third set (9–12) consists of VM consolidation through an Infiniband network; and the last set (13–16), consists in reduction of processor frequency (adjusted to the behavior of workloads) and consolidation is not utilized.

In the tests, we used 4 different sizes of VMs as mentioned above, for each test set. The table shows the total time the VM consolidation took to be completed for each case, and the last set shows the time of the exchange of DVFS states (assuming one second for each of them).

In an analysis of the first 3 sets (where VM consolidation is applied), we can see that consolidation through a point-to-point network over Gigabit Ethernet causes higher energy consumption (EC) the time needed to carry out the migration of the VM is taking into account. Although the final consumption of the tests with Infiniband network show increased energy consumption, it takes almost a third of the time to complete the migration process, when compared to Gigabit Ethernet. However, the best design aimed at energy savings in VM consolidation is the one that uses a centralized storage.

This can be explained by the different characteristics of VM migration in a point-to-point or via shared storage. When VM migration occurs in a point-to-point network, the whole VM image is copied over the network, from one host to the other. This causes an overload on the network throughout the process, and even in fast networks such as Infiniband, depending on the size of the VM, the migration process can take a long time. This overload does not exist when the images are located on a shared storage because in this case only a reference to the image within the storage is created in the destination host, and network traffic between the two hosts is used most of times only to update memory pages.

Therefore, we can define the design that further improves the trade-off between performance and energy saving is in a VM image storage, even with the increase of power consumption of the storage, where there is a shared storage. However, the results of using DVFS also seem to be very promising, considering that they do not impute migration time, and presents a total low energy consumption.

As the DVFS does not carry out migration, we must compare the energy consumption of using DVFS in both hosts with the option of migrating using a shared storage, in the moments after the migration (as depicted in Fig. 8, between t=13 s and t=22 s). This is the time when the idle host is turned off, and the two VMs are running on a single host, allowing maximum energy savings. The results of this evaluation are shown in Table 5, where VM Consolidation also includes the power consumption in the storage.

Results show that, regardless of the size of the VM, if it is possible to keep the VMs on a shared storage, this is the best option to save power. On the other hand, if the data center design does not present a shared storage, or if the PMs are interconnected by point-to-point, regardless of the speed of the network, the best option is to use DVFS. These findings drove the design of e-eco, which is detailed in the next section.

4. E-eco: energy-efficient cloud orchestrator

The reduction of energy consumption is a challenge for the provider of cloud services, as they must cope with increased demands and still maintain the performance expected by customers. Although several efficient strategies to save power are available, they usually have a high impact on performance. Strategies based on the reduction of the processor frequency for example, end up reducing the amount of instructions that the processor can execute at a given time interval, increasing the process runtime. Another strategy that turns off idle nodes, impacts on application performance when an environment has an organic behavior, with many context switches in small time intervals. The time during the host transitions and the migration of virtual machines involved in the consolidation ends up being added to

Table 4
Performance and energy-efficiency evaluation.

Test	*Proposed Arch.*	*Comm. Protocol*	*VM Size*	*Total Time (s)*	*Hosts EC (kWh)*	*Storage EC (kWh)*	*Total EC (kWh)*	*EC (kW h) VM Cons.*
1			Tiny	8	0.215	0.20	0.63	0.0014
2			Small	10	0.24	0.21	0.69	0.0019
3	A	iSCSI	Medium	12	0.275	0.216	0.766	0.0025
4			Large	15	0.305	0.22	0.83	0.0034
5			Tiny	240	0.325	–	0.65	0.0433
6		Gigabit Ethernet	Small	490	0.345	–	0.69	0.0939
7			Medium	925	0.36	–	0.72	0.185
8	B		Large	1800	0.37	–	0.74	0.369
9			Tiny	50	0.3	–	0.6	0.0083
10			Small	150	0.34	–	0.68	0.028
11		Infiniband	Medium	340	0.37	–	0.74	0.0698
12			Large	740	0.374	–	0.748	0.153
13			Tiny	1	0.209	–	0.418	–
14			Small	1	0.238	–	0.476	–
15	C	–	Medium	1	0.269	–	0.538	–
16			Large	1	0.295	–	0.59	–

Table 5
Energy consumption after VM consolidation or DVFS for different image sizes.

VM Size	*VM Consolidation through the storage*	*DVFS on 2 hosts without VM Consolidation*
Tiny	0.415 kW h	0.418 kW h
Small	0.44 kW h	0.476 kW h
Medium	0.475 kW h	0.538 kW h
Large	0.505 kW h	0.59 kW h

the total time of process execution. This raises the need for strategies to improve the energy efficiency of cloud computing data centers but at the same time maintaining the desired levels of operation.

The motivation of this work is based on the assumption that it is possible to improve the trade-off between energy savings and application performance in cloud environments, taking into consideration the overhead imposed by each technique on performance metrics. The knowledge about these overheads provides choices on which energy saving techniques can suit better the cloud applications behavior. These choices enable data center operators to be confident that the application of energy saving strategies in the data center will not incur negative impact in the applications, and therefore motivates operators to utilize them, contributing for the adoption of such techniques.

Several current applications utilize elasticity provided by cloud environments in order to ensure that performance metrics are maintained. Such applications, which include streaming video services, map-reduce, and e-commerce, perform scale-out on the resources for performance and therefore are an ideal case for the implementation of e-eco. Therefore, our proposal addresses a cloud environment that offers only one scalable service to customers. Another necessary condition for the implementation of this proposal is a cloud environment where the service provider can manage the technologies in the infrastructure layer in order to modify it without any restrictions. For this, the implementation of e-eco is suitable for private cloud environments, in which the service provider has full control and knowledge of underlying technologies.

This section is divided into two parts. In the first part we discuss the e-eco operation. In the second part, we present the e-eco evaluations.

4.1. E-eco strategy

Applications offered as services on cloud environments have different characteristics from applications supported by other large-scale environments such as clusters and grids. While in these more traditional environments the workload usually uses all the available resources and it has a finite completion time, cloud applications are offered as continuous services indefinitely with variation in the amount of resources required along the time. Therefore, due to the nature of cloud applications and the behavior of the requests performed by customers of this service, there is a fluctuation rate in the resources usage in such environment.

To support this fluctuation and aiming to maintain the quality of service within acceptable levels for the customer, the cloud platform must dynamically offer more resources when necessary, and reduce resources when there is underutilization. This dynamicity is supported by the elasticity of the virtualization layer, which provides an optimal setting for applications while maintaining a low utilization rate most of the time, but which might quickly require a larger set of resources supporting new customer requests.

This behavior is particularly observed in applications such as video streaming (Deng et al., 2013), MapReduce (Malewicz, 2011), and E-Commerce applications (Wee and Liu, 2010). Fig. 9 exemplifies this behavior, in which customers (a) access a cloud service through a listener (b). When there is a need for more resources, replicas of the service are instantiated through a load balancer (c) on new available physical resources (d). When the demand for resources is no longer needed, the instances can be released and the now idle resources are managed, aiming to energy saving. Resources increase or decrease by adding or removing capacity in accordance to the policies and predefined metrics. This feature should support the demands of customers paying only for what they need and use. This elastic scenario enables the utilization of different power saving techniques, as explored in this paper.

The e-eco is based on Zhu et al. (2012), and it divides hosts in the cloud infrastructure in three groups called Sites: Running (ACPI G0), Intermediate (ACPI S3), and Turned Off (ACPI G2). Fig. 10 shows this arrangement. The Running Site (RS) contains the hosts that are

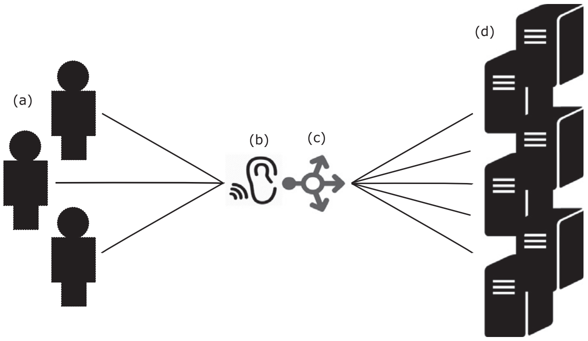


Fig. 9. Horizontal scalability in cloud infrastructures.

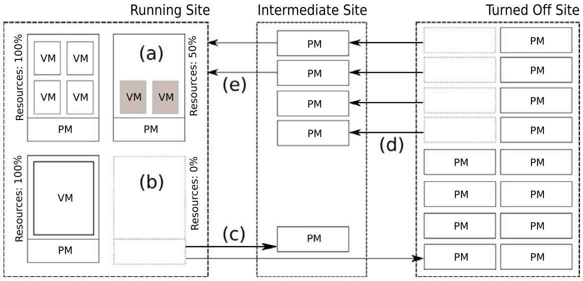


Fig. 10. First part of the e-eco strategy.

allocated by running instances, and/or hosts that are ending their execution and will be released. In the Intermediate Site (IS), there are hosts in suspended state, waiting to meet the demand for new resource allocations. Turned Off Site (TS) is where turned off hosts are kept.

Our assumption when designing e-eco was that decision on the amount of hosts to be kept at the IS has a large impact on energy savings of the system. As our approach is implemented at the IaaS level, e-eco calculates the number of required hosts in the IS based on two information: the amount of hosts running applications in the RS, and the frequency in which more resources are needed. In this context, e-eco *moves* hosts from TS to IS, and keeps them waiting for new demands. When there is no more demand, hosts can be released and transferred again to TS. Thus, the first part of e-eco consists in choosing, based on the number of hosts running and the frequency in which new hosts are required, how many hosts must be placed in an intermediate state, which could quickly meet new demands. This intermediate site increases performance by reducing response time, while the rest of the hosts are shut down to save energy.

When the Load Balancer of the cloud allocates resources to meet new demand from customers, hosts are allocated and maintained with instances running on RS. In this place, there are hosts with different rates of resource usage, as well as hosts that are to be released once they finalize their executions. When hosts are overloaded, new hosts should be allocated. This demand is provided by the IS that keeps hosts in sleep state, i.e., an intermediate state that saves more energy than idle hosts while responding faster than hosts turned off, resulting in energy saving by not performing very deep state transitions.

In our implementation, the demand of application is estimated by the equation below, where U_{ref} is the per-application agreed value of performance expectation, and U_i is the current metric measurement. High-level metrics such as transactions per second is often measured between the customer and the cloud provider. However, the infrastructure that supports cloud environments (IaaS) typically monitors low-level metrics, such as CPU, memory, and network usage. Translation between high-level metrics to low-level metrics is a very complex challenge in today's cloud environments. Moreover, monitoring high-level metrics are quite difficult, and impact performance and privacy of users. In addition, the communication channel between the customer and the cloud listener cannot be monitored by the cloud infrastructure without overhead on the communication channel. For all these reason, we used the model proposed in Rossi et al. (2015) to estimate U_i.

$$U_i = c_0 + c_1 \cdot avg1 + c_2 \cdot avg5 + c_3 \cdot avg15, \qquad (1)$$

Where U_i represents the estimated cloud transactions per second of applications in the entire hosts. The coefficients c_0, c_1 and c_2 are the weights assigned to each variable, in each loadavg times. $avg1$, $avg5$, and $avg15$ are monitored available values in `/proc/loadavg`, for 1, 5, and 15 min, respectively.

We used this estimated value (U_i) to power the model that calculates the required amount of hosts on IS, to meet the demand of RS.

$$\alpha = 1 - \frac{U_{ref} - U_i}{U_{ref}} - \lambda, \quad where\ 0 < \lambda < 1. \qquad (2)$$

Thus, based on the number of hosts in the RS and their historical, the aggressiveness of λ can be calculated as

$$\lambda = \frac{hosts_{run}}{hostsPrevious_{run}} * \Delta, \quad where\ \Delta = 0.5 \qquad (3)$$

and α can calculate the number of hosts in the IS. The handling of states of the hosts is performed over the network, using Wakeup-on-LAN (WoL) (Wakamatsu and Takahashi, 2010) technology, which allows managing different states on the network hosts. This occurs through UDP packets sent directly to the active network adapters of the hosts. These packets are received even by turned (soft) off hosts.

Algorithm 1 uses α to decide the required amount of hosts to be added or removed from the IS, intending to keep enough hosts to meet a possible demand of applications on RS. As the amount of hosts kept in IS is based on the number of hosts in RS, in case there are no hosts running, 10% of turner off hosts will be kept in IS (this value can be adjusted). This algorithm seeks the amount of currently existing hosts in the IS from Openstack queue via Ceilometer, and based on α, reaches a new value that the IS should have. If there are overuse, the algorithm uses WoL in order to turn on sufficient hosts (β) to complete the necessary amount of hosts in IS. In case of underutilization, the same technique is used to turn off the hosts from IS.

Algorithm 1. Intermediate site size adjustment procedure.

```
if $hosts_{run} = 0$ then
    $hosts_{sleep} \leftarrow hosts_{off} \times 0.1$
else
    $hosts_{sleep} \leftarrow hosts_{run} \times \alpha$
end if
if $hosts_{sleep} > hostsOld_{sleep}$ then
    $\beta \leftarrow hosts_{sleep} - hostsOld_{sleep}$
    for $\forall\ \beta$ do
    send *turning on signal* to $PM \in hosts_{off}$
    send *standby signal* to $\forall\ PM \in hosts_{sleep}$
    end for
end if
if $hosts_{sleep} < hostsOld_{sleep}$ then
    $\beta \leftarrow hostsOld_{sleep} - hosts_{sleep}$
    for $\forall\ \beta$ do
    send *turning off signal* to $PM \in hostsOld_{sleep}$
    end for
end if
```

Then, e-eco decides how deallocated hosts from RS should be managed. When a host is deallocated in RS, the algorithm checks whether the α value is satisfied, and if so, turn off these hosts. Otherwise, this algorithm is able to enhance hosts in IS. Therefore, Algorithm 2 presents a choice based on energy savings and performance. The algorithm can save energy by shutting down hosts at times where there is no need for IS, and it can also bring hosts to IS faster than turning on hosts from Turned Off Site.

Algorithm 2. Running Site PM deallocation procedure.

```
if $\alpha = 0$ then
    send *turning off signal* to $\forall\ hosts_{freed} \in hosts_{run}$
else
    for $\forall\ \beta$ do
    send *standby signal* to $hosts_{freed} \in hosts_{run}$
    end for
    send *turning off signal* to $\forall\ hosts_{freed} \in hosts_{run}$
end if
```

When hosts are released (Fig. 10 (b)), they may be moved to the IS, expecting new resource demands, and avoiding situations where there may be many exchanges either states (Fig. 10 (b)) or moved to TS.

IS keeps the amount of hosts needed to meet the demand for new resources. When e-eco decides that more resources are needed, new hosts are moved from TS (Fig. 10 (d)) and, if necessary, new instances are started over those resources and placed on RS (Fig. 10 (e)). Therefore, IS prevents loss of performance due to the speed to restarting hosts.

TS keeps turning off hosts. When there is a need to add new hosts in the IS, hosts are turned on and kept in a state of suspension ((Fig. 10 (d)). The number of hosts to be moved between TS and IS is decided by e-eco based on the number of hosts required for IS to meet the demand from RS.

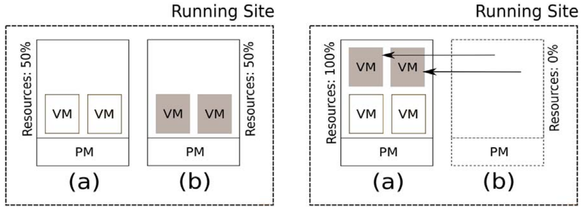
Furthermore, hosts with a low usage rate (Fig. 10 (a)) present an opportunity for management targeting energy savings. Fig. 11 shows the two possibilities that the proposed strategy allows to manipulate. It consists in the decision either allowing VM consolidation or reduction of the processors' frequency, depending VM images storage location and network latency. The first (Fig. 11 (a)) option is to reduce the frequency of the processors of underused hosts. Another alternative is to consolidate virtual machines from host (b) at host (a), allowing host (b) to be turned off or placed in a sleep state, as depicted in Fig. 11 (b).

The choice between these two options necessarily passes through the method employed for storage of VM images. In architectures where VM images are centralized and when migration occurs, only memory pages are transferred through the network. Thus, we believe that this scenario offers a suitable case for performing VM consolidation. On the other hand, when VM images are stored in the same host where they are instantiated, migration becomes costly, because besides the memory pages, VM image files must also be transferred through the network. In this situation, we believe that the most energy-efficient option would be using DVFS to reduce the processors frequency.

Algorithm 3 manages the decision either consolidating VMs or using DVFS. As this decision is based on the existence of a centralized storage for VM images, Algorithm 3 checks if there is an available storage in the infrastructure. This information is provided by the OpenStack Image Service API and accessed via Ceilometer. If there is an available storage, e-eco enables VM consolidation and disables the support for DVFS. If there is no centralized storage in infrastructure, VM consolidation is disabled, and the DVFS becomes enabled. The necessary changes on VM consolidation issues are performed using libvirt (Bolte et al., 2010), an API that provides resource management in virtualized environments. DVFS can be enabled and disabled using Simple Network Management Protocol (SNMP).

Algorithm 3. Running site energy-efficient selection procedure.

```
for $\forall PM \in hosts_{run}$ do
    if exist 'Storage' then
    enable VM Consolidation using libvirt
    disable DVFS Support on $\forall PM$ using SNMP
    else
    enable DVFS Support on $\forall PM$ using SNMP
    disable VM Consolidation using libvirt
    end if
end for
```



(a) CPU Frequency Changes (b) VM Consolidation

Fig. 11. Second part of the e-eco strategy. It consists in the decision either allowing VM consolidation or reduction of the processors' frequency, depending VM images storage location and network latency.

In summary, Algorithm 1 decides the number of hosts that must be kept in an intermediate condition based on the performance metric and the number of hosts running. The Algorithm 2 decides what action should be taken when a host running is freed. Algorithm 3 decides, based on the established architecture, if the environment must perform VM consolidation or offer a reduction in the processors frequency. This set of three algorithms are able to adjust the cloud environment managed by OpenStack in a way that it saves energy while maintaining a number of hosts to meet new demands, thereby maintaining compliance with performance metrics.

Nonetheless, the changes of hosts states among the three sites provided by the e-eco can not be performed at all times, because the α could increase rapidly, causing overload situations. Therefore, the changes applied by e-eco must be carried out in periods where there is greater energy savings, taking into account all factors that can influence that decision.

Thus, we can model the energy consumed during the idle time of the hosts denoted by

$$E_k = P_k T_k \tag{4}$$

Where P_k is the idle state power consumption, and T_k is the host duration time in the state. To reflect our proposal, we need to add to this model the cost during state transitions in terms of power consumption and duration time.

$$E_x = P_k T_{k'} + P_i T_i + P_j T_j + P_n T_n \tag{5}$$

P_i and T_i refer to the power consumption and time to go into a sleep state. P_j and T_j relate to the power consumption and time spent in sleep state. P_n and T_n concern the power consumption and time to exit a sleep state.

Generally, P_j is less than P_k, and P_i and P_n are generally larger than P_k. This means that in order to save energy, T_j must be long enough to offset the increased in energy consumption during T_i and T_n. Based on this, it can be noted that the shortest time interval which is energy efficient occurs when E_k=E_x.

Algorithm 4. Idle checking procedure.

```
for $\forall PM \in hosts_{run}$ do
    if $host idle time >= E_x$ then
    enable e-eco capabilities
    else
    disable e-eco capabilities
    end if
end for
```

e-eco performs state changes only when the time remaining idle hosts are greater than the cost of transitions. Otherwise, no state change takes place. This operation can be seen in Algorithm 4, and it prevents hosts in RS change to IS or TS, and after having to return to a ready state to meet a new demand from RS.

The asymptotic behavior of our strategy, in the worst case presents a linear behavior - $O(n)$. This is also due to the scalability of the management platform that accompanies the growth of the infrastructure components.

5. Evaluation

This section presents the evaluation of the improved trade-off between performance and energy savings enabled by e-eco in a real and a simulated cloud environments.

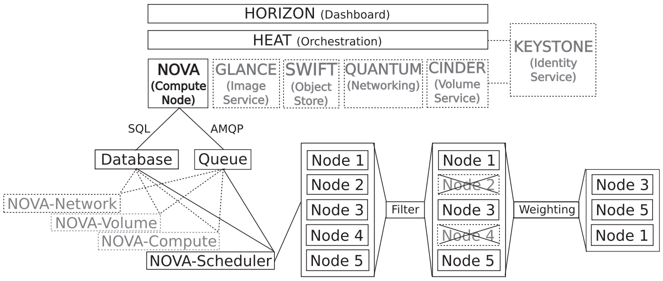


Fig. 12. Openstack scheduling architecture.

5.1. Cloud testbed and workload

The e-eco system was implemented on top of the OpenStack platform, which manages an infrastructure of 10 Citrix XenServer 6.2 hosts (Intel Xeon processors E5-4650 v2 2.40 GHz, 40/80 cores, 25 MB L3 Cache), using a Gigabit Ethernet network among them, and a centralized PowerVault MD3200i SAN Storage Array as a VM images storage. OpenStack performs scale-out operations based on low-level metrics such as use of processor, memory, network, or a combination thereof. As the analysis of the trade-off between performance and energy saving strategies must be carried out based on high-level metrics, we performed measurements where it was possible to estimate that satisfactory use of resources is achieved with different levels of resource usage.

The shaded area in Fig. 12 shows the components involved in the choice of available hosts to receive a VM when it is requested. The component that performs dynamic VM allocations on the resources is called Heat.

Heat manages the pool of available hosts using the OpenStack's Nova service, which maintains a service server in the manager host and client services in each compute host. The manager and the compute hosts communicate with each other to keep their status up-to-date. The horizon panel shows the user interface for managing OpenStack services for both users and administrators. Nova provides a scalable calculation platform supporting the provision and management of large numbers of servers and VMs. This is accomplished by provisioning scheduling algorithms, weighing and filter the hosts available according to pre-determined metrics.

Because the messages conveyed between the hosts are performed via RPC (Remote Procedure Call), they are managed by an AMQP (Advanced Message Queuing Protocol) queue. Therefore, this component consists of a broker that stores information, in addition to acting on the client hosts to manipulate the behavior of the VMs.

When there is a need to deploy new VMs on new hosts, the scheduler checks the pool that contains the most suitable hosts to receive the new instances. OpenStack supports multiple scheduling algorithms, so each has a different policy regarding the decision on which host is the best to receive a VM. The Filter scheduler is the most common and consists of two stages: filtering and weighting. The filter presents the pool of available hosts, and the weight prioritizes them based on policies such as number of processors, memory, disk, network, etc.

The e-eco design is composed of two parts. The first part is responsible for deciding on the number of hosts that must be kept on the hold state at Intermediate Site, in case there is a new demand for resources. Therefore, it is necessary that this demand is monitored. Monitoring can be done through high-level metrics such as response time or transactions per second, or low-level metrics such as utilization rate of resources, such as processor, memory, and network (Emeakaroha et al., 2010).

All necessary information for the e-eco power and performance management are offered by OpenStack through Ceilometer APIs. The relationship between e-eco and the informations provided by OpenStack can be seen in Fig. 13. The e-eco searches for the necessary information to implement decisions aimed at energy saving, such as the number of hosts in several states, the virtualization layer information, and internal events of each host.

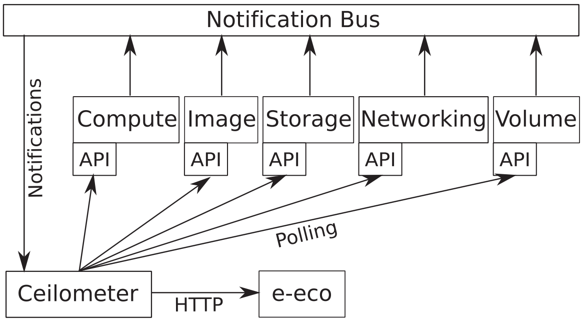


Fig. 13. Ceilometer API.

Tests were performed in different conditions, as follows:

- **Power-Agnostic**: it consists of an environment where there is no concern about energy saving, and the hosts are kept in one of two states: idle or busy. In this scenario, there is neither VM consolidation nor processors' frequency reduction. Fig. 14 illustrates such behavior.
- Alvarruiz et al. (2012): it consists of an environment where hosts are kept in one of two states: running or off. In this scenario, VM consolidation and processors' frequency reduction are also applied to the hosts. Fig. 15 illustrates such behavior.
- **Timeout Strategy** (Augustine et al., 2008; Meisner et al., 2009; Ponciano and Brasileiro, 2010): when the host in the G0 state becomes idle, it enters into the S3 state; the host returns immediately to the G0 state if it is requested; the host enters successively to a lower-power state if the timeout expires (300 sec (Intel, U.E.P. Agency, 2015; Reich et al., 2010; Lammie et al., 2009)).
  Fig. 16 illustrates such behavior.
- **E-eco:** it consists of the management of hosts among execution, standby, or power-off states in order to conserve energy. In addition, it applies VM consolidation or DVFS techniques.

The choice of a cloud workload is a determining factor to evaluate e-eco on a real environment. However, cloud traces are not available. The application behavior used in the test was based on Pucher et al. (2015) trace.

Besides, we followed a literature based methodology and used an Online Transaction Processing (OLTP) database, which is widely used in tests in cloud environments (Scheuner et al., 2015; Li et al., 2013), and Jia et al. (2015). In this way, Sysbench (Kopytov, 2014) was used as a benchmark for conducting evaluations of this work. This tool consists of a benchmark that performs multithread transactional queries on MySQL instances. In our evaluations, we used the complex transactional mode with 75% of read operations and 25% of write operations. As a result, Sysbench provides performance information on the number of TPS achieved by the system. In order to mimic the behavior of scalable cloud applications, we used two test sets (customers and services) simultaneously. In total, customers perform 2000 sessions/connections on a database with 4.5 GB, with an SLO set at 900 TPS. When the amount of TPS is lesser than the agreed, the environment should perform scale-out, launching new database instances on new hosts to supply the growing demand. When the amount

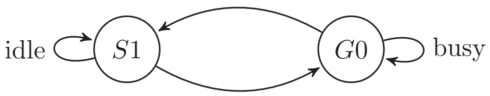


Fig. 14. Power-agnostic strategy's policies employed on one host.

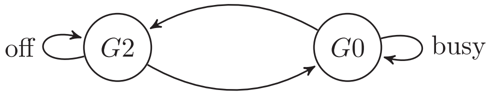


Fig. 15. Alvarruiz et al. energy-saving strategy's policies employed on one host.

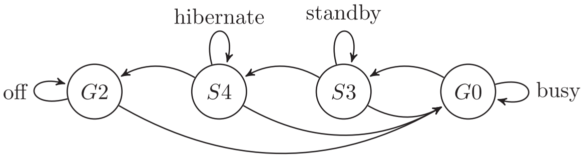


Fig. 16. Timeout strategy's policies employed on one host.

of TPS is higher than the established, the environment must perform scale-in for resource-efficient.

Fig. 17 shows the workload behavior (c), and defines three main phases: Regular Operation (c), SLA Saturation (a), and SLA Violation (b). Regular Operation is the maintenance of SLO set to 500 TPS. When there is under-utilization of resources (e.g., by reduction of customer requests), the spare resources make the TPS exceeds the SLA required by the application. This case is called SLA Saturation, and the environment must be adjusted to the correct amount of resources so that again there is a Regular Operation, avoiding waste of resources. On the other hand, when the amount of TPS is reduced due to the increasing customer requests, then more resources must be added to the environment to supply this new demand and return the environment to a Regular Operation. We call this time interval SLA Violation.

5.2. E-eco evaluations on real cloud environments

Table 6 shows that the Alvarruiz et al. strategy saves more energy, but it has a greater impact on application performance. The energy savings shown is due to the fact that all the idle hosts are powered off. The performance impact occurs for the same reason, because while hosts are restarting, the SLA remains in violation. The SLA saturation does not impact directly on the performance, but the shutdown time impacts on the resource usage and on the energy consumption consequently.

On the other hand, the strategy with the least impact on application performance is the Power-Agnostic. This is because the hosts are always powered on and ready to supply new demands immediately. However, as the hosts remain connected, the power consumption is the highest among all strategies. The impact of these strategies on the performance are shown in Table 6, where we described the test runtime, and how much time was spent during Regular Operation, SLA Violation, and SLA Saturation. It shows that Alvarruiz et al. reduces the performance of applications because they incur SLA violations for a longer time and have a long adjustment time of the environment during SLA saturations, which adds to the overhead on regular operation of the application.

The main goal of e-eco is to improve the trade-off between performance and energy savings. Results showed that for cloud environments, e-eco can save more energy than the Power-Agnostic strategy, and have less impact on application performance. This is because e-eco maintains a set of intermediate hosts that respond quickly to new demands, reducing the impact on performance, at the same time it saves energy while keeping the rest of hosts turned off. When compared to the timeout strategy, e-eco shows better performance, greater energy savings addition to demonstrating fewer SLA violation, enabling an increased regular operating time on cloud resources.

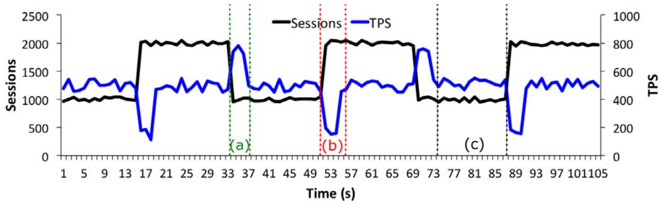


Fig. 17. Workload behavior phases.

Table 6
Evaluation of e-eco conducted in a real cloud environment on 10 Xen hosts managed by Openstack, compared to other strategies.

Strategies	Execution time	Energy consumption	Cloud usage		
			Regular operation	SLA violation	SLA saturation
Power-agnostic	5502 s	646 Wh	85%	3%	2%
Alvarruiz et al.	5917 s	455 Wh	50%	24%	26%
Timeout	5715 s	531 Wh	70%	14%	16%
E-eco	5590 s	485 Wh	82%	3%	5%

Besides, the EDP (Energy Delay Product) metric (Blem et al., 2013) considers the latency together with the energy consumption through

$$EDP = Energy(Joules)xDelay(Seconds) \tag{6}$$

Such metric was used and it can clarify, summarize and corroborate our findings correlating performance with energy consumption (the lower the EDP, the better the correlation between performance and energy saving). The results of such well-known metric can be seen in Fig. 18, where e-eco obtained the best results when compared to the other strategies.

The scale of a cloud environment supplied by a service provider is not easily reproduced in academic environments (Barker et al., 2014). For example, it is estimated that Amazon EC2 holds more than 450,000 servers (Netcraft, 2015). Although the tests have been carried out in a small-scale cloud (10 hosts), we believe that the performance gains and power savings provided by the e-eco remains for private cloud environments with greater number of hosts. The works by Cameron et al. (2007) and Zomaya (Zomaya and Lee, 2012) make reference to the growth of energy consumption in relation to the scale of hosts in a near-linear or linear rate. Performance may also be supported by the scalability of cloud environment (Chieu et al., 2011). This mean that the energy savings and performance obtained by e-eco are expected to be proportional, for each cloud usage rate, to a higher number of hosts.

5.3. E-eco evaluations on simulated cloud environments

In addition to the empirical experiments, we also analyzed the scalability of our approach with an enhanced version of the CloudSim simulator (Calheiros et al., 2011; Xavier et al., 2016). Since in our work we target a very specific type of provisioning infrastructure, namely Cloud Data Centers, several of the public available traces do not represent our target workload and therefore do not include the information we need to validate our orchestration strategy. They contain mostly infrastructure data, like the CPU usage rate of each

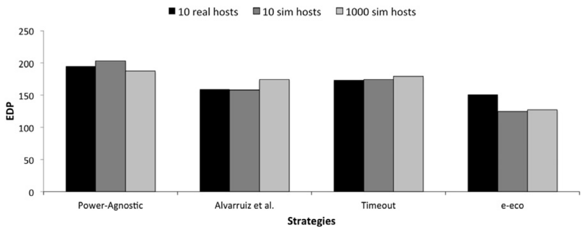


Fig. 18. Energy-delay product - real and simulated environments.

Table 7
CloudSim simulation using 10 hosts.

Strategies	Execution time	Energy consumption	SLA violation/ saturation
Power-agnostic	5505 s	673 Wh	4%
Alvarruiz et al.	5764 s	477 Wh	41%
Timeout	5704 s	537 Wh	28%
E-eco	5714 s	381 Wh	4%

VM, and we need application level traces. Because of that, we decided to use the Pucher et al. (2015) trace that reproduce the behavior of real cloud applications. The tested strategies were the same evaluated in the previous section, with the addition of a virtual machine migration heuristic (Beloglazov et al., 2012) known as Minimum Migration Time (MMT) in cases when VM consolidation is needed. Such heuristic migrates a VM that wants the minimum time to whole a migration comparatively to the other VMs billed to the host.

The first experiment was to simulate the real test of 10 hosts in order to calibrate the CloudSim simulator. The results can be seen in the Table 7. The latency – the time spent during transitions – is compatible with the SLA Violation and Saturation values previously presented in Table 6. The results showed a smaller standard deviation up to 2%. Such results show that the simulator can achieve the same results presented in tests on the real environment, keeping a very significant accuracy. Fig. 18 showed this behavior.

Table 8 shows the simulation results testing a cloud environment with 1000 hosts using the Pucher et al. (2015) trace. The results show that even in larger cloud environments, e-eco is able to maintain the best relationship between performance and power savings when compared to other strategies. Generally, when the environment is very large, with lots of hosts and physical links, multiple platform controllers are used to segment the management. In the same direction, various OpenStack replicas can manage different groups of hosts on the same infrastructure. Therefore, our proposal maintains the same levels of quality, even with increased scalability.

E-eco presents these results due to the fast response of hosts in IS when there is a new demand from RS. This occurs because hosts in IS respond faster than hosts in TS, such as the proposed by Alvarruiz et al. (2012). Furthermore, hosts in IS are in a less deep state of energy-saving, yet consuming less energy than idle hosts on a Power-Agnostic strategy. The balance between performance and energy-saving by e-eco strategies is translated by EDP results in Fig. 18, where again, for larger data center environments, e-eco is able to maintain a better relationship between the two metrics than the Timeout strategy.

6. Related work

Modern operating systems and virtualized platforms bring, through the ACPI, opportunities for power management with the use of energy-aware strategies on idle hosts (Chase et al., 2001; Heath et al., 2005; Zong et al., 2007). The question of minimization of operational costs through the reduction of power consumption in cloud environments is widely discussed in current research, as shown by Gao et al. (2013). Isci et al. (2013) show that there is an opportunity for energy-savings strategies in these environments using the concept of sleep states. Sleep states refer to the S-states of the ACPI specification and are adopted by strategies that “wake up” hosts from sleeping or hibernation states, bringing the system to an operational state.

Table 8
Simulation results among evaluated strategies on a cloud with 1000 hosts, taking into account the execution time in seconds, the energy consumed in watts.

Strategies	Execution time	Energy consumption	SLA violation/ saturation
Power-agnostic	5715 s	57,248 Wh	6%
Alvarruiz et al.	5864 s	50,720 Wh	39%
Timeout	5785 s	53,609 Wh	30%
E-eco	5720 s	39,114 Wh	5%

Min et al. (2014) present a framework that selects the best sleep state and processor frequency based on typicalworkloads for smart phones. To switch from an idle state to another with lower power consumption, some thresholds (such as idle time and time in each sleep state) were used 1020 along with a heuristic that is applied to different states on the device. Results showed energy-savings of up to 50%.

Niyato et al. (2009) proposed a power management approach based on a Markov model to adjust the number of active servers for maximum energy-savings. Although a considerable amount of power can be saved by shutdown and restart operations on hosts, the main goal was to perform the configuration of autonomous resources and enable online regulation according to the service behavior, power consumption, and SLA requirements. Results showed an increase of energy efficiency by up to 30%, minimally impacting performance.

Beloglazov and Buyya (2010) presented a heuristic for virtual machines allocation in the cloud, with the goal of saving power. The heuristic determines when and what virtual machines should be allocated on available resources with a minimum amount of migration to reduce the overhead and to avoid violation of SLAs. When virtual machines are migrated, idle hosts may enter into a sleep state, reducing thereby the overall power consumption. Results showed power savings of up to 83% compared to energy-agnostic scenarios, although they showed a minimal violation of SLAs.

Alvarruiz et al. (2012) proposed a management system for clusters and clouds that saves power by turning off idle hosts across the network. When a host is deallocated by the task, a timer checks the time of host in idle state. This timer prevents the host to shut down when there is a possibility for a new job to run. When the timeout is reached, the host is turned off. Results showed energy-savings of 38% for cluster environments and 16% for cloud environments, respectively.

Duy et al. (2010) presented the design, implementation, and evaluation of a scheduling algorithm integrated with a predictor that uses neural networks to optimize the power consumption of servers in a cloud. The future workload prediction is based on historical usage. According to the prediction, the algorithm turns off unused hosts with the intention of minimizing the number of running servers, thus also minimizing power consumption of the hosts. Evaluations showed that this model is able to reduce power consumption in up to 46% compared to energy-agnostic environments.

Zhu et al. (2012) proposed splitting a cloud in four areas: busy, active idle, sleep, and shutdown. In the first area, hosts are allocated to running applications. The second area retains a certain amount of hosts in the idle state, waiting to meet any possible demand. At the next level, hosts are kept in a state of suspension, ending with a level in which the hosts are turned off. This division provides an environment that classifies hosts into categories related to the environment usage. Results show that this organization can reduce power consumption of idle hosts in up to 84%, with an impact on the runtime of up to 8.85%.

Lefèvre and Orgerie (2010) showed a cloud architecture that saves power due to several factors, such as startup and shutdown hosts, control of the usage rate of resources, and migration of virtual machines. An algorithm to predict the behavior of the workload has been proposed. The experimental results showed differences in power consumption among the various scenarios (a heuristic that turns on/off hosts, a heuristic that migrates virtual machines, or a mixed heuristic). In these experiments, energy-savings of up to 25% was achieved when compared to an energy-agnostic environment. Moreover, results showed that the best option may vary depending on the type of resource.

Santana et al. (2010) proposed a model for predicting the behavior of applications on web clusters, aiming to apply DVFS policies and turn idle hosts off, trying to keep up the quality of service. The metric assessed was the rate of processor usage. Results showed an energy-saving of up to 59%, trying to keep up the quality of service in 95%. On several occasions, this quality of service could not be maintained precisely due to the action of turning off and restarting hosts.

Feller et al. (2012) proposed a consolidation model for workloads coupled with dynamic adaptation of sleep states and changes in processor frequency. The paper presents the proposed model, whose main goal was to minimize the number of hosts that run applications, doing this by adjusting the workload on the hosts and setting limits to control the transition between idle and turned off hosts. An evaluation of the proposed model is not presented.

Krioukov et al. (2010) proposed a manager for heterogeneous clusters with a focus on saving power and the least possible impact on the response time of the tasks. Three different architectures were simulated: Nehalem, Atom, and BeagleBoard, and as a workload trace, 7 days of HTTP traffic of Wikipedia were used. DVFS techniques have been used over underutilized hosts, sleep states on idle hosts, besides a slice of shutdown unused hosts in the cluster. The decision on the ideal amount of hosts to meet the tasks is based on combinatorial optimization (knapsack problem) and the results showed an energy-saving of up to 27%, with less than 0.03% of lost requests. In clusters, the use of DVFS is not recommended because when the frequency of the processor is reduced, the number of instructions that can be performed are also reduced. The used trace had a low usage rate, and probably because of this, results were so satisfactory. With workloads with higher rates of use or a queue of tasks with dynamic arrival rates enough, not only the response time is likely to be greatly affected but also an increase in power consumption is expected.

Ding et al. (2015) presents a VM allocation algorithm on cores with different frequencies. Within certain periods, this organization is performed again, making the environment to self-adjust always aiming energy savings. Through simulation, the authors claim that its strategy can save up to 20% energy. However, the paper assumes that frequency changes are performed on cores individually, however this assumption do not hold in modern processors.

Table 9
Summary of related work.

Paper	Power techniques	Refers to the performance
Min et al. (2014)	sleep states + DVFS	No
Niyato et al. (2009)	sleep states	Yes
Beloglazov and Buyya (2010)	VM consolidation + sleep states	No
Alvarruiz et al. (2012)	sleep states	No
Duy et al. (2010)	sleep states	No
Zhu et al. (2012)	sleep states	Yes
Lefèvre and Orgerie (2010)	VM consolidation + sleep states	No
Santana et al. (2010)	DVFS + sleep states	Yes
Feller et al. (2012)	VM consolidation + sleep states	No
Krioukov et al. (2010)	VM consolidation + sleep states + DVFS	Yes
Ding et al. (2015)	VM consolidation + DVFS	No
Maccio and Down (2015)	sleep states	Yes
Shen et al. (2015)	VM consolidation	Yes
Dong et al. (2015)	VM consolidation	No
E-eco	VM consolidation + sleep states + DVFS	Yes

Maccio and Down (2015) proposed modeling some sleep states for servers based on Markov Chains. The model was supported by four states: off, setup, busy, and idle. Through an incoming jobs guided by Poisson, the model optimizes the states on multiple hosts to meet SLA restrictions. In the same way, Shen et al. (2015) used a Markov model to allocate virtual machines on hosts in order to save energy, aiming to improve the trade-off between performance and energy savings. Compared with the state-of-the-art suggested at work, the proposal achieves 23% energy savings.

Dong et al. (2015) proposed an algorithm that scales in a multi-dimensional manner the VMs on a homogeneous mobile cloud depending on two factors: the rate of CPU usage and the bandwidth among available hosts. Based on previous analysis, the minimum energy consumption and the number of physical machines in operation are derived. Results enable the development of an algorithm for virtual machines placement in order to save power.

Table 9 summarizes the studies presented in this section to allow better visualization of the applied energy saving techniques, as well as identification of any bias aiming application performance. We can see that only the Krioukov et al. (2010) work manages VM consolidation, sleep states, and DVFS the same time, although the focus of the work is HPC clusters. In addition, some studies analyze application performance, although they prioritize energy saving over application performance.

Some discussed works provided a basis for the development of our proposed e-eco system. Initially, based on Alvarruiz et al. (2012), we evaluated the cost of transitions between different states of suspension, given that the proposal presented by the researchers only turn on/off the hosts, which affected the execution time of applications, especially in HPC clusters. The work of Zhu et al. (2012) proposed splitting the cloud environment in several sites with different states of suspension. However, our evaluations demonstrate that the cost of transition between states and energy savings between some of them were very close, and thus there is no direct benefit in using them. The work of Santana et al. (2010) introduced the DVFS as a proposal for energy savings in clouds, which raised one of the questions addressed in our work, which is the decision between reducing the processors frequency and keeping the environment as it is or consolidate VMs and turning off idle hosts. Feller et al. (2012) addresses the trade-off between energy savings and performance through sleep states and DVFS, but it does not handle virtualized environment in order to manage the migration ability of VMs and it does not take into account the overhead imposed by the transitions between states, which may suggest that the simulated results still need a more fine-tuning.

In other areas such as sensor networks, the placement of resources is widespread. He et al. (2012) considers the quality of sensing as a utility function, proposing a greedy algorithm to perform placement and scheduling through the activation or deactivation of the sensors. In the same direction, Mo and Xu (2015) uses a Kalman filter algorithm for scheduling sensors, based on current events, in order to develop a deployment strategy that enhances the coverage of the sensors. E-eco also manages the state of the participating hosts, but these hosts have considerable delay time between transitions. This delay does not exist

Table 10
E-eco evaluations in an environment without server consolidation, where e-eco applied DVFS on processors.

Size	Testbed		Execution time	Energy consumption	SLA violation/saturation	EDP
	Real	Simulated				
10 hosts	x		5608 s	493 Wh	7%	155 (0.10^{8})
10 hosts		x	5615 s	499 Wh	6%	157 (0.10^{8})
1000 hosts		x	5729 s	50,826 Wh	6%	166 (0.10^{11})

or is not as significant in sensors. Therefore, this is a concern that E-eco takes into consideration, given the fact that this delay directly affects the performance metrics.

Although all these solutions address applications performance issues and energy-saving possibilities in computing environments, neither assesses their mutual impact, or present a solution to improve such a trade-off. Thus, the Energy-Efficient Cloud Orchestrator (e-eco) manages the two discussed energy-saving options, bounding its utilization to the applications performance. On contrary of previous work, e-eco was developed with specific focus on the trade-off between saving power and application performance.

7. Conclusions and future work

The advantages brought by cloud computing has been promoting the establishment of data centers that support many different applications. Among the cloud advantages, intelligent use of resources is a key factor, as through server virtualization, services can be scaled as they need. This management of resources impacts on operating costs for the provider of services, and among them, one of the most significant costs is the one with power consumption. Besides the consolidation of virtual machines intrinsically enabled by virtualized environments, several energy-saving techniques are used on cloud environments.

These techniques have different levels of intrusiveness on the cloud environment, and offer many energy saving levels. The problem with its use is the overhead they impose on others equally important metrics in an enterprise environment. One of the most important affected metrics, which directly involves customers' experience of quality, is the performance of applications. As cloud services offer access through the Internet, application performance becomes a determining factor on consumer loyalty to the service offered.

In this direction, this paper presents e-eco, an Energy-Efficient Cloud Orchestrator that improves the trade-off between energy savings and application performance through a smart management of a set of power-saving techniques. A prototype has been implemented on real and simulated cloud environments, and tests have shown that e-eco maintains the balance between energy saving promoted with minimal impact on performance. Results of our evaluation demonstrated that e-eco is able to reduce energy consumption in up to 25% compared to power-agnostic approaches at a cost of only 6% of extra SLA violations. When compared to existing power-aware approaches, e-eco achieved the best relationship between performance and energy-saving, as expressed by EDP. These results showed that e-eco improves the trade-off between power savings and applications performance in order to enable a cloud environment that is at the same time economical and responsive.

The tests presented in Section 5 were applied on cloud environments that allow server consolidation via migration of virtual machine (using a centralized storage to store virtual machine images). However, several large scale cloud environments do not perform server consolidation. Table 10 shows the results of e-eco on such environments, using the same amount of hosts, traces, and strategies of the previous tests. Although this option offers increased EDP than the previously one, it still presents a better EDP than other strategies.

As future work, we will improve e-eco capabilities through new energy-efficient algorithms for VM placement in the case of VM consolidation. In addition, we showed in this paper that in cloud environments without centralized storage for VM images, the most energy-efficient decision is not to perform VM consolidation. However, most of today's data centers have redundant network paths that may support link aggregation techniques to increase the communication channel between hosts, and enable faster traffic of such VM images. Thus, we will investigate how such environments may affect the decisions made by e-eco. Finally, new sleep states as hibernate can be incorporated to the model, given that new hardware components, such SSDs, have smaller delays for state savings and thus do not impose extra overhead for hibernation compared to suspension.

References

Alvarruiz, F., de Alfonso, C., Caballer, M., Hernandez, V., 2012. An energy manager for high performance computer clusters. In: Proceedings of the IEEE10th International Symposium on Parallel and Distributed Processing with Applications (ISPA), pp. 231–238.

Augustine, J., Irani, S., Swamy, C., 2008. Optimal power-down strategies. SIAM J. Comput. 37 (5), 1499–1516.

Barker, A., Varghese, B., Ward, J.S., Sommerville, I., 2014. Academic cloud computing research: Five pitfalls and five opportunities. In: Proceedings of the 6th USENIX Conference on Hot Topics in Cloud Computing, HotCloud' 14, USENIX Association, Berkeley, CA, USA, pp. 2–2.

Beloglazov, A., Buyya, R., 2012. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. Concurr. Comput.: Pract. Exp. 24 (13), 1397–1420.

Beloglazov, A., Abawajy, J., Buyya, R., 2012. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Gener. Comput. Syst. 28 (5), 755–768.

Beloglazov, A., Buyya, R., 2010. Energy efficient resource management in virtualized cloud data centers. In: Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, CCGRID' 10, IEEE Computer Society, Washington, DC, USA, pp. 826–831.

Berl, A., Gelenbe, E., Di Girolamo, M., Giuliani, G., De Meer, H., Dang, M.Q., Pentikousis, K., 2010. Energy-efficient cloud computing. Comput. J. 53 (7), 1045–1051.

Bhuiyan, M.F.H., Wang, C., 2013. Energy-efficient virtual machine management in heterogeneous environment: challenges, approaches and opportunities. In: Proceedings of the 2013 IEEE International Conference on Systems, Man, and Cybernetics, SMC' 13, IEEE Computer Society, Washington, DC, USA, pp. 4116–4121.

Blem, E., Menon, J., Sankaralingam, K., 2013. Power struggles: Revisiting the risc vs. cisc debate on contemporary arm and x86 architectures. In: IEEE Proceedings of the 19th International Symposium on High Performance Computer Architecture (HPCA2013), pp. 1–12.

Bolte, M., Sievers, M., Birkenheuer, G., Niehörster, O., Brinkmann, A., 2010. Non-intrusive virtualization management using libvirt. In: Proceedings of the Conference on Design, Automation and Test in Europe, DATE' 10, European Design and Automation Association, 3001 Leuven, Belgium, Belgium, pp. 574–579.

Buyya, R., Broberg, J., Goscinski, A.M., 2011. Cloud Computing Principles and Paradigms. Wiley Publishing.

Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A.F., Buyya, R., 2011. Cloudsima tool kit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Softw. Pract. Exp. 41 (1), 23–50.

Cameron, K.W., 2014. Energy efficiency in the wild why datacenters fear power management. Computer 47 (11), 89–92.

Cameron, K.W., Ge, R., Feng, X., 2007. Designing computational clusters for performance and power. In: Zelkowitz M.V., (Ed.), Architectural Issues, vol. 69 of Advances in Computers, Elsevier, pp. 89–153.

Chase, J.S., Anderson, D.C., Thakar, P.N., Vahdat, A.M., Doyle, R.P., 2001. Managing energy and server resources in hosting centers. In: Proceedings of the eighteenth ACM symposium on Operating systems principles, SOSP' 01, ACM, New York, NY, USA, pp. 103–116.

Chieu, T., Mohindra, A., Karve, A., 2011. Scalability and performance of web applications in a compute cloud. In: IEEE Proceedings of the 8th International Conference on e-Business Engineering (ICEBE), pp. 317–323.

Deng, D., Lu, Z., Fang, W., Wu, J., 2013. Cloudstreammedia: a cloud assistant global video on demand leasing scheme. In: Proceedings of the 2013 IEEE International Conference on Services Computing, SCC' 13, IEEE Computer Society, Washington, DC, USA, pp. 486–493.

Ding, Y., Qin, X., Liu, L., Wang, T., 2015. Energy efficient scheduling of virtual machines in cloud with deadline constraint. Future Gener. Comput. Syst. 50 (C), 62–74.

Dong, Y., Zhou, L., Jin, Y., Wen, Y., 2015. Improving energy efficiency for mobile media cloud via virtual machine consolidation. Mob. Netw. Appl. 20 (3), 370–379.

Duy, T.V.T., Sato, Y., Inoguchi, Y., 2010. Performance evaluation of a green scheduling algorithm for energy savings in cloud computing. In: IEEE International Symposium on Parallel Distributed Processing, Workshops and Phd Forum (IPDPSW), pp. 1–8.

Emeakaroha, V., Brandic, I., Maurer, M., Dustdar, S., 2010. Low level metrics to high level slas - lom2his framework: Bridging the gap between monitored metrics and sla parameters in cloud environments. In: International Conference on High Performance Computing and Simulation (HPCS), pp. 48–54.

Eyerman, S., Eeckhout, L., 2011. Fine-grained dvfs using on-chip regulators. ACM Trans. Archit. Code Optim. 8 (1), 1:1–1:24.

Feller, E., Rilling, L., Morin, C., 2012. Snooze: a scalable and autonomic virtual machine management framework for private clouds. In: Cluster, Cloud and Grid Computing (CCGrid), 2012 Proceedings of the 12th IEEE/ACM International Symposium on, pp. 482–489.

Freeh, V.W., Lowenthal, D.K., Pan, F., Kappiah, N., Springer, R., Rountree, B.L., Femal, M.E., 2007. Analyzing the energy-time trade-off in high-performance computing applications. IEEE Trans. Parallel Distrib. Syst. 18 (6), 835–848.

Gao, Y., Wang, Y., Gupta, S.K., Pedram, M., 2013. An energy and deadline aware resource provisioning, scheduling and optimization framework for cloud systems. In: Proceedings of the Ninth IEEE/ACM/IFIP International Conference on Hardware/Software Codesign and System Synthesis, CODES+ISSS' 13, IEEE Press, Piscataway, NJ, USA, pp. 31:1–31:10.

Grover, A., 2003. Modern system power management. Queue 1 (7), 66–72.

He, S., Chen, J., Cheng, P., Gu, Y., He, T., Sun, Y., 2012. Maintaining quality of sensing with actors in wireless sensor networks. IEEE Trans. Parallel Distrib. Syst. 23 (9), 1657–1667.

He, L., Zou, D., Zhang, Z., Yang, K., Jin, H., Jarvis, S.A., 2011. Optimizing resource consumptions in clouds. In: Proceedings of the 2011 IEEE/ACM 12th International Conference on Grid Computing, IEEE Computer Society, pp. 42–49.

Heath, T., Diniz, B., Carrera, E.V., Meira, W., Jr., Bianchini, R., 2005. Energy conservation in heterogeneous server clusters. In: Proceedings of the tenth ACM SIGPLAN symposium on Principles and practice of parallel programming, PPoPP' 05, ACM, New York, NY, USA, pp. 186–195.

Intel, U.E.P. Agency, 2015. Energy star* version 5.0 system implementation whitepaper, ⟨http://www.energystar.gov/ia/partners/product_specs/program_reqs/Computers_Intel_Whitepaper_Spec5.pdf⟩

Isci, C., McIntosh, S., Kephart, J., Das, R., Hanson, J., Piper, S., Wolford, R., Brey, T., Kantner, R., Ng, A., Norris, J., Traore, A., Frissora, M., 2013. Agile, efficient virtualization power management with low-latency server power states. ACM SIGARCH Comput. Archit. News 41 (3), 96–107.

Jia, Q., Shen, Z., Song, W., van Renesse, R., Weatherspoon, H., 2015. Supercloud opportunities and challenges. SIGOPS Oper. Syst. Rev. 49 (1), 137–141.

Kolpe, T., Zhai, A., Sapatnekar, S.S., 2011. Enabling improved power management in multicore processors through clustered dvfs. In: Proceedings of the Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, pp. 1–6.

Kopytov, A., 2014. Sysbench: a system performance benchmark, ⟨https://launchpad.net/sysbench⟩.

Krioukov, A., Mohan, P., Alspaugh, S., Keys, L., Culler, D., Katz, R.H., 2010. Napsac: design and implementation of a power-proportional web cluster. In: Proceedings of the First ACM SIGCOMM Workshop on Green Networking, Green Networking' 10, ACM, New York, NY, USA, pp. 15–22.

Lammie, M., Brenner, P., Thain, D., 2009. Scheduling grid workloads on multicore clusters to minimize energy and maximize performance. In: Proceedings of the Grid Computing, 2009 10th IEEE/ACM International Conference on, pp. 145–152.

Lefèvre, L., Orgerie, A.-C., 2010. Designing and evaluating an energy efficient cloud. J. Supercomput. 51 (3), 352–373.

Li, Z., Zhang, H., O'Brien, L., Cai, R., Flint, S., 2013. On evaluating commercial cloud services. J. Syst. Softw. 86 (9), 2371–2393.

Maccio, V., Down, D., 2015. On optimal policies for energy-aware servers. Perform. Eval. 90 (C), 36–52.

Malewicz, G., 2011. Beyond mapreduce. In: Proceedings of the Second International Workshop on MapReduce and Its Applications, MapReduce' 11, ACM, New York, NY, USA, pp. 25–26.

Meisner, D., Gold, B.T., Wenisch, T.F., 2009. Powernap: eliminating server idle power. In: Proceedings of the 14th International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS XIV, ACM, New York, NY, USA, pp. 205–216.

Min, A.W., Wang, R., Tsai, J., Tai, T.Y.C., 2014. Joint optimization of dvfs and low-power sleep-state selection for mobile platforms. In: Proceedings of the 2014 IEEE International Conference on Communications (ICC), pp. 3541–3546.

Mo, L., Xu, B., 2015. Coordination mechanism based on mobile actuator design for wireless sensor and actuator networks. Wirel. Commun. Mob. Comput. 15 (8), 1274–1289.

Netcraft, 2015. Amazon web services' growth unrelenting, ⟨http://news.netcraft.com/archives/2013/05/20/amazon-web-services-growth-unrelenting.html⟩.

Niyato, D., Chaisiri, S., Sung, L.B., 2009. Optimal power management for server farm to support green computing. In: Proceedings of the 2009 9th IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGRID '09, IEEE Computer Society, Washington, DC, USA, pp. 84–91.

NRDC, 2014. Data center efficiency assessment, ⟨http://www.nrdc.org/energy/files/data-center-efficiency-assessment-IP.pdf⟩.

Ponciano, L., Brasileiro, F., 2010. On the impact of energy-saving strategies in opportunistic grids. In: Grid Computing (GRID). Proceedings of the 11th IEEE/ACM International Conference on, pp. 282–289.

Pucher, A., Gul, E., Wolski, R., Krintz, C., 2015. Using trustworthy simulation to engineer cloud schedulers. In: Cloud Engineering (IC2E), 2015 IEEE International Conference on, pp. 256–265.

Reich, J., Goraczko, M., Kansal, A., Padhye, J., 2010. Sleepless in seattle no longer. In: Proceedings of the 2010 USENIX Conference on USENIX Annual Technical Conference, USENIXATC' 10, USENIX Association, Berkeley, CA, USA, pp. 17–17.

Rossi, F.D., Conterato, M., Ferreto, T.C., De Rose, C.A.F., 2014. Evaluating the trade-off between dvfs energy-savings and virtual networks performance. In: Proceedings of the Thirteenth International Conference on Networks, ICN' 14, IARIA, Nice, France, pp. 01–06.

Rossi, I.C.R.B.R., De Oliveira, F., De Rose, C., 2015. Non-invasive estimation of cloud applications performance via hypervisor's operating systems counters. In: Proceedings of the Fourteenth International Conference on Networks (ICN), 2015, pp. 177–184.

Santana, C., Leite, J.C.B., Mossé, D., 2010. Load forecasting applied to soft real-time web clusters. In: Proceedings of the 2010 ACM Symposium on Applied Computing, SAC' 10, ACM, New York, NY, USA, pp. 346–350.

Schatz, R., Hossfeld, T., Janowski, L., Egger, S., 2013. Datatraffic monitoring and analysis. Springer-Verlag, Berlin, Heidelberg, Ch. From Packets to People: Quality of Experience As a New Measurement Challenge, pp. 219–263.

Scheuner, J., Cito, J., Leitner, P., Gall, H., 2015. Cloud workbench: Benchmarking iaas providers based on infrastructure-as-code. In: Proceedings of the 24th International Conference on World Wide Web Companion, WWW' 15 Companion, International World Wide Web Conferences Steering Committee, Republic and Canton of Geneva, Switzerland, pp. 239–242.

Shen, D., Luo, J., Dong, F., Fei, X., Wang, W., Jin, G., Li, W., 2015. Stochastic modeling of dynamic right-sizing for energy-efficiency in cloud data centers. Future Gener. Comput. Syst. 48 (C), 82–95.

Wakamatsu, H., Takahashi, T., 2010. Communication system with automatic start-up and reply from a distance in the internet. In: Proceedings of the 10th WSEAS International Conference On Applied Informatics And Communications, And 3rd Wseas International Conference On Biomedical Electronics And Biomedical Informatics, AIC'10/BEBI'10, World Scientific and Engineering Academy and Society (WSEAS), Stevens Point, Wisconsin, USA, pp. 109–114.

Wee, S., Liu, H., 2010. Client-side load balancer using cloud. In: Proceedings of the 2010 ACM Symposium on Applied Computing, SAC' 10, ACM, New York, NY, USA, pp. 399–405.

Xavier, M.G., Rossi, F.D., De Rose, C.A.F., Calheiros, R.N., Gomes, D.G., 2016. Modeling and simulation of global and sleep states in acpi-compliant energy-efficient cloud environments. Concurr. Comput.: Pract. Exp.. http://dx.doi.org/10.1002/cpe.3839, (n/a–n/aCpe.3839).

Zhu, H., Liu, Y., Lu, K., Wang, X., 2012. Self-adaptive management of the sleep depths of idle nodes in large scale systems to balance between energy consumption and response times. In: Proceedings of the 2012 IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom), CLOUDCOM' 12, IEEE Computer Society, Washington, DC, USA, pp. 633–639.

Zomaya, A.Y., Lee, Y.C., 2012. Energy Efficient Distributed Computing Systems, 1st Edition, Wiley-IEEE Computer Society Pr.

Zong, Z., Qin, X., Ruan, X., Bellam, K., Nijim, M., Alghamdi, M., 2007. Energy-efficient scheduling for parallel applications running on heterogeneous clusters. In: Proceedings of the 2007 International Conference on Parallel Processing, ICPP' 07, IEEE Computer Society, Washington, DC, USA, pp. 91–113.