INTERNATIONAL JOURNAL OF PARALLEL, EMERGENT AND DISTRIBUTED SYSTEMS
2019, VOL. 34, NO. 6, 690–705
https://doi.org/10.1080/17445760.2018.1448931

Taylor & Francis
Taylor & Francis Group

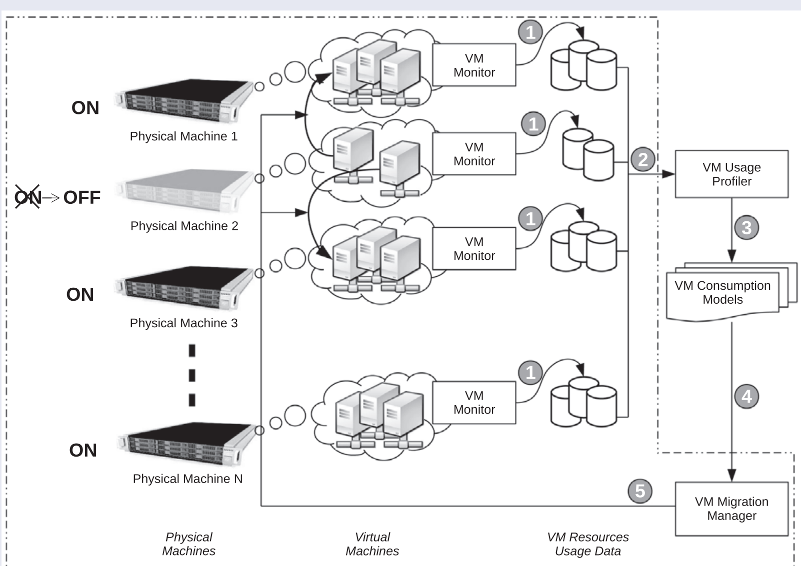
Data analytics for energy-efficient clouds: design, implementation and evaluation

Albino Altomare, Eugenio Cesario and Andrea Vinci

National Research Council of Italy, Institute for High Performance Computing and Networks, Rende, Italy

ABSTRACT

The success of Cloud Computing and the resulting ever growing of large data centers is causing a huge rise in electrical power consumption by hardware facilities and cooling systems. This results in an increment of operational costs of data centres, that is becoming a crucial issue to deal with. Consolidation of virtual machines (VM) is one of the key strategies used to reduce the power consumption of Cloud servers. For this reason, it is extensively studied. Consolidation has the goal of allocating virtual machines on a few physical servers as possible while satisfying the Service Level Agreement established with users. Nevertheless, the effectiveness of a consolidation strategy strongly depends on the forecast of the VM resource needs. Predictive data mining models can be exploited for this purpose. This paper describes the design and development of a system for energy-aware allocation of virtual machines, driven by predictive data mining models. In particular, migrations are driven by the forecast of the future computational needs (CPU, RAM) of each virtual machine, in order to efficiently allocate those on the available servers. The experimental evaluation, performed on real-world Cloud data traces, reports a comparison of performance achieved by exploiting several classification models and shows good benefits in terms of energy saving.



The energy-aware cloud architecture.

ARTICLE HISTORY
Received 15 December 2017
Accepted 2 March 2018

KEYWORDS
Green computing; energy-aware clouds; data mining for energy efficiency

CONTACT Eugenio Cesario eugenio.cesario@icar.cnr.it

1. Introduction

1.1. Motivations

In recent years, an increasing number of companies are migrating their data, software, and services on the Cloud, because this relieves them from the need for setting up basic hardware and software infrastructures, and thus enabling more focus on innovation and creating business value for their services. In addition, the opportunity of using Cloud resources on a pay-as-you-go basis, the availability of powerful data centres and high bandwidth connections is speeding up the success and popularity of Cloud systems, which is making on-demand computing a common practice for many enterprises and scientific communities [1–4].

The success of Cloud Computing is causing, as a natural consequence of it, a notable increase in its global operative costs. In particular, the ever growing of large data centres results in a huge rise of electrical power consumed by hardware facilities and cooling systems. For instance, the energy consumption of the Data Center Networks roughly occupies 1.5% of global power consumption and increases double every five years [3]. Moreover, the rapid diffusion of mobile applications and social networking sites, e.g. Facebook and Twitter, have accelerated data access on the network, especially to share media (photos, videos) contents. It is estimated that video streaming traffic (e.g. YouTube, Netflix) generates more than 50% of the downstream traffic during the peak period and it will increase 16-fold videos during the year 2017 [3]. Thus, according to the current growing speed in data centres, it will consume about 140W billion energy in the United States every year, making the energy consumption a key challenge in the coming years [1].

Unfortunately, the high power utilisation of Cloud data centre has several consequences. First, operational costs of data centre management will become overwhelming for the companies, that will be forced to increase the price of the services offered to the users [1,5]. Second, high power consumption results in reduced system reliability and devices lifetime due to overheating of hardware. Third, the higher the energy request, the higher CO2 emissions to produce it. This contributes to the greenhouse effect, thus affecting the carbon footprint of data centres and aggravating, on the global scale, the problem of global warming [1]. Therefore, the energy consumption reduction is becoming an important issue that is attracting a huge attention in cloud data centres management.

1.2. Objectives and contributions

One of the major reasons for the huge amount of consumed power is the inefficiency of data centres, which are often under-utilised: several research works estimate that servers in many organisations typically run at less than 30% of their full capacity [6,7]. In addition, it has to be considered that an idle server consumes approximately 65–70% of the power consumed when it is fully utilised [8]. As a consequence of that, having servers idle or whose computational facilities are not properly used, generates relevant waste of energy.

At the software level, one of the most popular ways to reduce the power consumption of a data centre consists in the *consolidation of virtual machines*. By exploiting the virtualization mechanism, user processes are not assigned directly to servers, but are first associated with Virtual Machine (VM) instances, which, in turn, are run by servers. A virtual machine is a software black box encapsulating one (or more) virtualized service(s), that can be moved, copied, created and deleted depending on management decisions. In addition, the use of virtualization allows heterogeneous platforms to be executed on any kind of hardware facility. The basic idea of the consolidation approach is to allocate the virtual machines on a few physical servers as possible, while satisfying various constraints specified as part of the system requirements. In this way, by assigning the virtual machines to a minimal number of physical servers, some of the nodes will get a higher load, but others will be unused and can be turned off (or hibernated). These operations, de facto, reduce the number of physical servers needed to be active (i.e. turned on) and reach the goal of saving energy.

The optimal mapping of VMs to servers, so as to minimise energy consumption, is a NP-hard problem and requires a full knowledge of the server loads. From one side, it is profitable to maximise the number of virtual machines allocated on each physical node, so as to minimise the number of servers turned on. On the other hand, an aggressive consolidation of VMs could excessively load physical servers and can lead to performance loss. Nevertheless, it is essential for Cloud providers to offer reliable Quality of Service (QoS) for the customers that is negotiated in terms of Service Level Agreements (SLA), i.e. throughput, response time. For such a reason, consolidation algorithms have to deal with a power-performance trade-off, which reduces the energy consumption while satisfying performance constraints at the same time.

This paper describes the design and development of a system for energy-aware allocation of virtual machines on Cloud physical nodes. In particular, the migration is driven by the prediction of the future computational needs (CPU, RAM) of each virtual machine, in order to efficiently allocate those on the available servers. The approach is composed of three main steps. First, virtual and physical machines are monitored, in order to collect data on their real resource utilisation. Second, such data are analysed by a data mining algorithm, to discover knowledge models that are descriptive of the resource needs of each virtual machine; in this way, each model should reflect the characteristics of resource usage of the individual VM accurately. Third, at runtime, once resource demands are predicted by such knowledge models, the allocation of virtual machines on the available servers is periodically done, by minimising the number of busy servers while satisfying performance constraints and Service Level Agreement established with users. We apply the proposed methodology to a real-world dataset (data collected through a private Cloud system), exploiting several data-driven classification models to perform resource-usage predictions. Experimental evaluation shows that, due to complexity and large data involved in the application scenario, the proposed energy-aware framework achieves good benefits in terms of energy saving, number of migrations and SLA violations.

For the sake of clarity, this paper extends the works presented in [9,10] and it provides several original contributions with respect to the previous ones. In particular: (i) the problem formulation and the description of the proposed framework have been enhanced; (ii) the experimental evaluation has been extended, by reporting several additional experimental results; (iii) finally, a comparison of performance achieved by the framework exploiting several classification models is reported.

1.3. Plan of the paper

The rest of the paper is organised as follows. Section 2 reports the state-of-art of the approaches for energy-aware virtual machines allocation proposed in literature. Section 3 describes our approach, where migrations of virtual machines are driven by resource-demand predictive models. Section 4 reports experimental results carried on real-world resource-usage traces and evaluates the effectiveness and efficiency of the proposed strategy. Finally, Section 5 concludes the paper and plans further research works.

2. Related work

As the Cloud computing paradigm and its applications are rapidly emerging, many studies focus on algorithms and procedures aimed at reducing the power consumption of data centres and improving the 'green' characteristics of Cloud environments. We briefly report in the following the most interesting software-based approaches for virtualized data centres, oriented to optimise task scheduling, resource allocation and virtual machines consolidation.

A greedy algorithm for an efficient allocation of virtual machines on physical servers is proposed in [1]. To address this issue, authors propose a virtual machine placement scheme meeting multiple resource constraints, such as the physical server size (CPU, memory, storage, bandwidth, etc.) and network link capacity. The final aim is to improve resource utilisation and reduce both the number of active physical servers and network elements, so as to finally reduce energy consumption. The virtual machine placement problem is abstracted as a combination of bin packing problem and quadratic assignment problem, and a greedy algorithm by combining minimum cut with the best-fit has been described and experimented in a simulated scenario.

In [11] authors have studied the problem of request scheduling for multi-tiered web-applications in virtualized heterogeneous systems to minimise energy consumption, while meeting performance requirements. In particular, the paper analyses the effect of performance degradation due to high utilisation of different resources when the workload is consolidated. The paper shows that the energy consumption per transaction results in a U-shaped curve, and it is possible to determine the optimal utilisation point. To handle the optimization over multiple resources, authors model the workload consolidation problem as a multidimensional bin packing problem and propose a heuristic to solve it.

In [12] a new architecture for cloud resource allocation is proposed, which maps groups of tasks to customised virtual machine types. The mapping is based on task usage patterns detected through the analysis of utilisation trace data. To do that, tasks using a similar quantity of resources are grouped together by clustering algorithms. Then the clustering output is used for the determination of customised virtual machine types. The proposed solution results to be effective in decreasing the resource wastage in data centres via virtualization and efficient resource allocation policies.

A new virtual machine placement algorithm is proposed in [13], which is aimed at handling deployment, migration, and cancellation of virtual machines. Specifically, a mathematical model is used to build a CPU utilisation estimator that can be used to forecast current and future energy efficiency at different levels (virtual machine, node, infrastructure, and service levels). Experimental results show that when running web workloads, estimators focused on noise filtering provide the best precision even if they react slowly to changes, whereas reactive predictors are desirable for batch workloads.

A power-aware algorithm for VM allocations in virtualized heterogeneous computing environments is proposed in [14]. The approach is aimed at leveraging the min, max and shares parameters of VM monitor, which represent minimum, maximum and proportion of the CPU allocated to VMs sharing the same resource. The key strategy is to adopt a shares based mechanism for the hypervisor to distribute spare resources among contending VMs. The running tasks are assigned to the VMs (that have been configured by the aforementioned parameters), in order to minimise the power usage. A limitation of the approach is that the allocation of VMs is static (and not adapted at runtime) and no other resources except for the CPU are considered during the VM reallocation.

A bio-inspired approach for adaptive assignment of VMs to servers and their dynamic migration is presented in [5], with a twofold goal: reduce the energy consumption and meet the Service Level Agreements established with users. The main goal is to cluster VMs in as few servers as possible, using statistical procedures for the assignment and the migration of VMs. Specifically, a new VM is assigned to one of the available servers through statistical Bernoulli trials for which the success probability depends on the current utilisation of the servers. On the other hand, the migration procedure fosters the relocation of VMs from servers in which the current utilisation is either too high or too low, that is, above or below two defined thresholds. In the first case, the migration of a VM helps to prevent a possible overload of the server, which may lead to Service Level Agreement (SLA) violations. In the second case, the objective of the migration is to take VMs away from lightly loaded servers, and then power off these servers.

A study and evaluation of some energy-aware allocation policies, aimed at minimising energy costs of Cloud platforms, is provided in [15]. In this case, the power consumption is minimised by switching off/on some servers whenever it is convenient and thus improving the utilisation of the server farm. To do that, two heuristics are proposed. The first one, named adaptive, takes decision at runtime with respect to the current computational load of the servers. The second one, named predictive, is based on the assumption that the load typically follows certain patterns (daily, weekly, etc.) and thus estimates the future arrival rate of jobs. Authors claim that such a heuristic has experimentally shown good results under different workload conditions.

In [16] an algorithm for energy-efficient management of homogeneous resources in Internet hosting centres is proposed. The main idea is to determine the resource demand of each application at its current request load level, and to allocate resources in the most efficient way. In particular, the system maintains an active set of servers selected to serve requests for each service. Energy consumption is reduced by switching idle servers to power saving modes (e.g. sleep, hibernation), through the application of statistical approaches that reduce the number of unproductive reallocations and lead to a more stable and efficient control.

In [17] an approach for power-efficient resource management of Web servers, satisfying a fixed SLA (response time) and load balancing, is provided. In detail, authors propose two power saving techniques: (i) switching power of computing nodes on/off and (ii) Dynamic Voltage and Frequency Scaling (DVFS). The main idea of the policy is to estimate the total CPU frequency required to provide the necessary response time, determine the optimal number of physical nodes and set the proportional frequency to all the nodes. A similar technique has been proposed in [18], where the load balancing is handled by an external system, which is driven by a centralised algorithm.

A software consolidation solution for saving both energy and cost is proposed in [19], which suits the Software-as-a-Service (Saas) cloud model. Instead of trying to consolidate only virtual machines, the approach relies on the dynamic consolidation of software applications, so as to reduce the total number of VMs used. The algorithm takes into account issues like software isolation and migration, and shows better performance when software consolidation is combined with virtual machine consolidation.

A solution for the management of resources aims at coping with traffic burst is presented in [20]. The main contribute of the work consists in a novel model for workload bursts forecasting, which is exploitable to create, migrate or destroy VMs so as to both guaranty QoS and save costs. However, such a solution does not explicitly consider energy efficiency as a goal.

Most of cited approaches try to reduce energy consumption by virtual machine consolidation, which is currently considered as the most effective way exploited to decrease the number of active servers. Assignments and migrations of virtual machines are made by some scheduling algorithm or statistic technique. However, resource needs of virtual machines may change over time due to the workload variability, so such techniques sometimes are not able to well model the variability over time of the VM resource needs. Differently from the approaches described above, in this paper we propose a VM data-driven consolidation approach to forecast resource needs of each machine in the future. The approach is based on data mining models. Such knowledge models are trained off-line from historic data and are used at runtime to drive energy-efficient allocation of virtual machines.

3. The proposed approach

This section presents the high-level architecture we designed for supporting energy-aware allocations of virtual machines, based on predictive data mining models. The architecture, depicted in Figure 1, includes some components working on-line (to plan and execute live VM migrations) and other ones acting off-line (to analyse log usage data and extract predictive models). The system includes the following modules:

- *Physical and Virtual Machines*. Physical machines are the underlying physical computing servers which constitute the hardware infrastructure of the Cloud data centre. On each server, multiple virtual machines can be dynamically started and stopped according to incoming requests, and they can be migrated from one to another machine with respect to the decisions taken by the VM Migrations Manager module. Whenever a server results with no virtual machine running, it can be turned off or switched to a low-power mode for saving energy.
- *VM Monitor*. Virtual machines resource needs (CPU, RAM, storage) are monitored by the *VM Monitor* module, which is a software component running on the server and in charge of storing VM resource usage data over a time horizon. It is usually implemented as a low-priority running process that collects resource usage data including CPU, memory and I/O bandwidth of all virtual

machines and physical machines. The most important virtualization suites provide a module to collect performance data (e.g. vCenter Server of VMware [21]), that are usually stored in text files (labeled in Figure 1 as *VM Resource Usage Data*).

- *consumption modeller*. This component analyses resource usage data of virtual machines by executing a data mining algorithm, with the goal of discovering patterns or trends that are generally followed. In particular, the *consumption modeller* extracts a specific model for each virtual machine, aimed at having an accurate model for each one. Different kinds of models can be exploited, i.e. classification tree, numeric regression, neural network, etc. as well as more complex ones (obtained by ensemble-learning or meta-learning approaches). The data mining algorithms adopted in this work are described in Section 4.2.
- *VM Migrations Manager*. This module handles virtual machine migrations, with the goal of minimising the number of running servers. Synchronized by a periodic trigger, each T time the *VM Migrations Manager* obtains, by the consumption model, a prediction of the resource sizes that will be used by the virtual machines in the next T-time window. According to that, it plans a new relocation, aimed at reducing the number of busy physical machines while satisfying SLA constraints. Obviously, servers with no virtual machines running on it are switched off. Moreover, to avoid that CPU/RAM demands (at runtime) exceed resources actually available on the physical servers, the consolidation policy allocates the virtual machines by cautiously considering as upper bound the δ ratio (with δ assuming values in $[0, 1]$) of the server resource availabilities, thus saving the other $1 - \delta$ ratio for unpredicted needs.

We describe here by an example the steps composing the whole process, by pointing out details of the interactions between entities composing the architecture. Let us suppose N physical servers, where several virtual machines in charge of running client tasks are allocated on. At runtime, CPU and RAM needs of each virtual machine are monitored by the *VM Monitor* module and suitably logged as VM Resource Usage Data (step 1). Periodically, such data are analysed off-line by the *consumption modeller* (step 2), with the goal of discovering usage models for CPU and RAM resources (step 3), labeled in Figure 1 as f_{CPU} and f_{RAM} functions. Such models are used on-line by the *VM Migrations Manager*, with a period T, to forecast resource sizes used by virtual machines in the next T-time window (step 4). Based on this information, it plans a new relocation of the virtual machines, aimed at managing their consolidation in a minimum number of servers while satisfying SLA constraints (step 5). As a consequence of that, some servers will get a higher load, but others will be unused and can be hibernated or turned off. For example, Figure 1 shows that virtual machines of the server 2 are moved to other servers (1 and 3 respectively) and it can be switched off.

4. Experimental evaluation

To evaluate effectiveness and performance of the proposed energy-aware approach, we carried out several tests considering various scenarios. The main goal of our experiments is to quantify how much energy can be saved when VM migrations are driven by data mining models with respect to a no energy-aware case, while satisfying performance constraints and offering a reliable quality of service to users. The experimental evaluation, performed on real data, shows how our approach can be applied on a concrete private Cloud scenario and how it can provide benefits in terms of energy saving.

We also present a comparative evaluation of several algorithms used for usage resource predictions. Details about tests and results on real data are reported in the following subsections. In particular, Section 4.1 presents the experimental setting used for the experiments, Section 4.2 describes the data mining algorithms exploited to discover predictive usage models, Section 4.3 introduces the evaluation metrics adopted to assess system performances, and Section 4.4 reports the experimental results achieved by running tests in different scenarios.

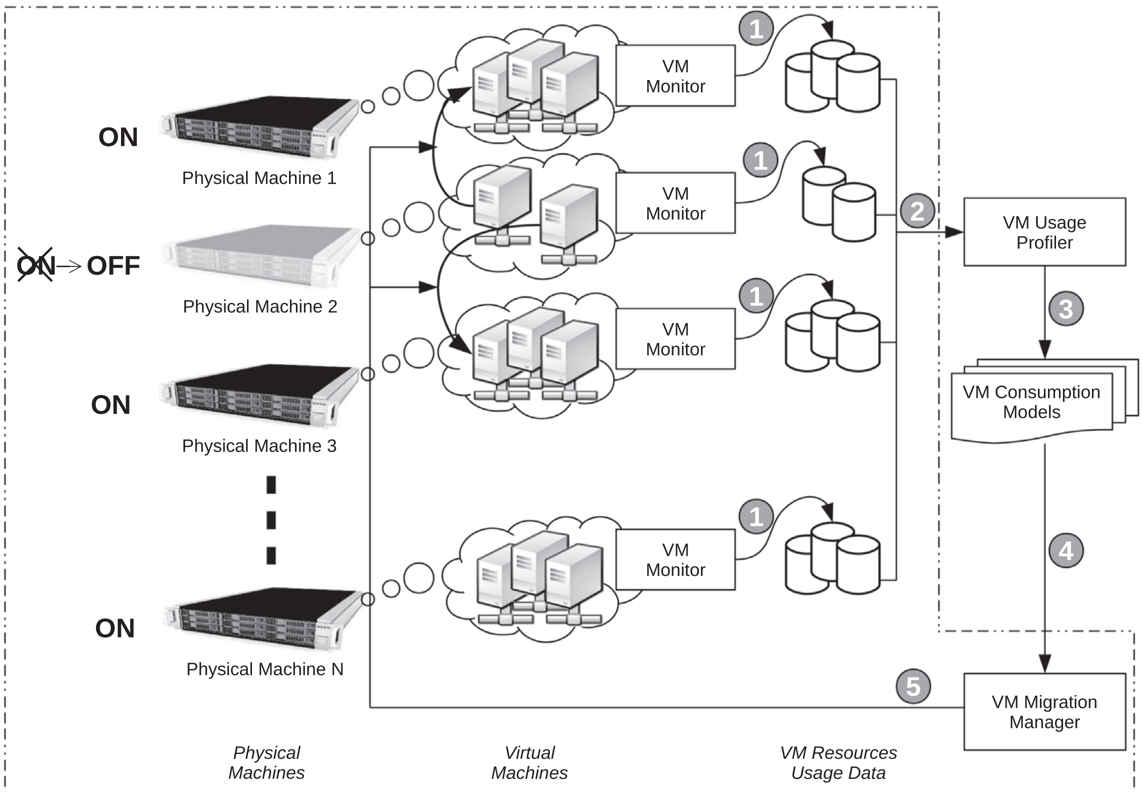


Figure 1. The energy-aware cloud architecture.

4.1. Experimental setting and VM usage data

The experimental scenario of our tests is a real private Cloud system composed of 10 servers, running *VMWare* as virtualization software, with 75 virtual machines running on the servers. Each server is equipped with 8 cores, all having 3 GHz CPU frequency; 4 servers have 32 GB and 6 servers have 24 GB of RAM. Such machines have been monitored for 80 days during their operations, by collecting data on resources used to execute running tasks. The tweaking of resources has been done by a module implemented on *vSphere*. In particular, performance data have been stored in a file, populated by records reporting CPU and RAM needs (sampled every 5 min) of the virtual machines.

As the targeted system is a generic Cloud computing environment, it is essential to evaluate it on a large-scale virtualized data centre infrastructure. However, it is extremely difficult to conduct repeatable large-scale experiments on a real infrastructure, which is required to evaluate and compare the proposed resource management algorithms. Therefore, to ensure the repeatability of experiments, simulations have been chosen as a way to evaluate the performance of the proposed heuristics. Specifically, the experimental evaluation has been carried out by running an ad hoc discrete-event simulator developed in Java. Our goal was to reproduce the behaviour of the architecture described in Section 3 by simulating virtual machine migrations, server switching on/off, the energy consumed by servers, etc. An event queue is used to exchange messages and data among Java objects associated with the system components, i.e. physical server, virtual machines, VM migration manager, etc. To make the simulator behaviour the most adhering to the real case, we implemented it by taking into account the time needed to perform a live migration of a virtual machine from a server to another one, the additive CPU overhead to do that, the capacity of each physical machine (in terms of CPU power and memory), etc. In particular, the time elapsed for migrations has been computed as the size of its memory divided by the available network bandwidth [22], while the CPU overhead has been

Table 1. Bin boundaries for CPU usage.

Bin	Level consumption	CPU usage range (GHz)	RAM usage range (GB)
1	Very low	0.18–1.26	0.052–1.63
2	Low	1.27–2.34	1.63–3.21
3	Medium	2.35–3.43	3.21–4.79
4	High	3.44–4.51	4.79–6.37
5	Very high	4.52–5.60	6.37–7.95

considered equal to almost 1% of the nominal CPU power, in accordance with the results reported in [23].

The experimental evaluation has been carried on as follows. First, we collected usage data from a real private Cloud (above described) and partitioned it in training and test sets. Second, we analysed the training set to discover predictive models (f_{CPU} and f_{RAM} functions) for each virtual machine. Third, by exploiting the simulator developed to reproduce the behaviour of the architecture described in the paper, we performed several runs on the test data set and evaluated energy consumed, number of migrations, SLA violations, etc. to assess the effectiveness of the energy-aware approach. We also present a comparative evaluation of several algorithms exploited for usage resource predictions.

4.2. Discovery of predictive usage models

As described in Section 3, the consumption modeller analyses virtual machine usage data with the goal of discovering their resource utilisation patterns and trends. Such data-driven knowledge is used at runtime to plan an energy-efficient allocation of virtual machines across the available servers. In particular, in our approach the discovery of predictive usage models has been modelled as a classification task and has been implemented through several algorithms, as described in the following.

A classification task can be defined as follows. Let us suppose a collection of records, each one modelled as a tuple $\langle \mathbf{x}, y \rangle$, where $\mathbf{x}$ is the *attribute set* (features of the record) and y is the class label (category or target attribute). The classification goal consists in learning a *target function* f, i.e. the classification model, which maps each possible attribute set $\mathbf{x}$ to one of the predefined class labels y. The target function can be used to classify future data for which the class labels are unknown.

For the purposes of this work, the input data set consists of $\langle vm, month, day, dayOfWeek, hour,$ *minute*,*cpu-usage*,*ram-usage*$\rangle$ instances, representing that the virtual machine *vm* has requested *cpu-usage* and *ram-usage* resource sizes at the timestamp identified by the $\langle month, day, dayOfWeek, hour,$ *minute*$\rangle$ attribute set. In order to be suitable for a classification task, *cpu-usage* and *ram-usage* class attributes have been discretized in 5 bins, as reported in Table 1, where each bin represents a specific value range. This has been performed by adopting a discretization approach [24], which tries to put the same number of objects into each bin (or interval) and well works when there are outlier values. Such intervals correspond to several levels of usage resources, that can be seen as interval labels (i.e. 1.27–2.34 GHz, 2.35–3.43 GHz, 3.44–4.51 GHz) or conceptual labels (i.e. *low*, *medium*, *high* usage). The goal of the consumption modeller is to learn (for each virtual machine *vm*) two *target functions*, f_{CPU} and f_{RAM}, which can be used to predict future resource needs of *vm* in terms of computation and memory usage. Specifically, when a new attribute set $\langle month, day, dayOfWeek, hour, minute \rangle$ is given in input, f_{CPU} and f_{RAM} return the estimated CPU and RAM needs of a virtual machine at the given timestamp. The whole dataset, composed of almost 1728 millions of tuples, has been split into two parts, whose sizes are circa $2/3$ (53 days) and $1/3$ (27 days) of the whole file, respectively. The first one has been used as training set to discover predictive models, while the second one (or testing set) has been exploited as a test bed to validate the approach.

For what concerns the classification algorithms, in this work we exploit *J48*, *RandomForest*, and *JRip*. Specifically, *J48* is an implementation of the C4.5 classification algorithm [25], which builds decision trees from a training set using the concept of information entropy. During the execution, at each node

Table 2. Average error prediction of the classifiers, computed by ten-fold cross-validation.

Classification algorithm	Error prediction	
	CPU (%)	RAM (%)
J48	8.1	21.1
JRip	6.7	18.9
RandomForest	10.5	25.7

of the tree J48 chooses the attribute that most effectively splits the current sample set into subsets w.r.t. the normalised information gain. Once the attribute is chosen, a value is select for splitting the current set into subsets, and thus for making the decision. *JRip* is an implementation of the RIPPER algorithm (Repeated Incremental Pruning to Produce Error Reduction) [26], which is a propositional rule learner adopting rule pruning policies aimed at avoiding data overfitting. Finally, *RandomForest* [27] is an ensemble classification algorithm that builds a collection of different decision trees, and exploits them for classifying new objects by aggregating predictions of all trees through a voting technique. The average error rate (i.e. number of wrong predictions on the total number of predictions) of classifiers on the test set is reported in Table 2, which has been computed by a ten-fold cross validation method. Error predictions are quite low for CPU usage, while show higher values for RAM. In particular, we can notice that *JRip* achieves generally an higher accuracy than the other two algorithms.

4.3. Evaluation metrics

The goal of the experimental evaluation is to assess the energy saving effectiveness of the proposed framework, as well as to evaluate how the number of virtual machine migrations and SLA violations are affected by it. In particular, we evaluated the results by exploiting the following performance metrics:

- *Total Energy Consumption E*. To analyse the amount of consumed energy, we adopt a general model described by several studies ([8,28–31]). Specifically, the power consumed in data centres is mostly determined by the CPU, memory, disk storage, and network interfaces. Nevertheless, in comparison to other system resources, the CPU consumes the main part of the energy, hence the consumed power is generally computed by taking into account only the CPU utilisation. From these studies results that the power consumption at time t of a server i is expressed by the following mathematical model:

$$P_i(t) = P_{MAX} \cdot (0.7 + 0.3 \cdot u_i(t))$$

  where P_{MAX} is the maximum power consumed when the server is fully utilised (it is usually equal to 250W for modern servers) and $u_i(t)$ is the CPU utilisation ratio at time t in the server i. From the formula, it clearly results that on average an idle server consumes approximately 70% of the power consumed by a server running at the fully utilised CPU. This fact justifies the technique of switching idle servers to the sleep mode to reduce the total power consumption. In our experiments, the total energy consumption E_i of a physical node in the time interval $[t_0, t_1]$ has been computed as an integral of the power consumption function over the given period of time. Finally, the total energy E consumed in a data centre composed of N servers is computed as:

$$E = \sum_{i=1}^{N} E_i = \sum_{i=1}^{N} \left[\int_{t_0}^{t_1} P_i(t)\, \mathrm{d}t \right]$$

- *Service Level Agreement Violations*. Meeting Quality of Service (QoS) requirements is extremely important for Cloud computing environments. Generally, QoS requirements are commonly formalised in the form of SLAs violations, which can be determined as the number of times

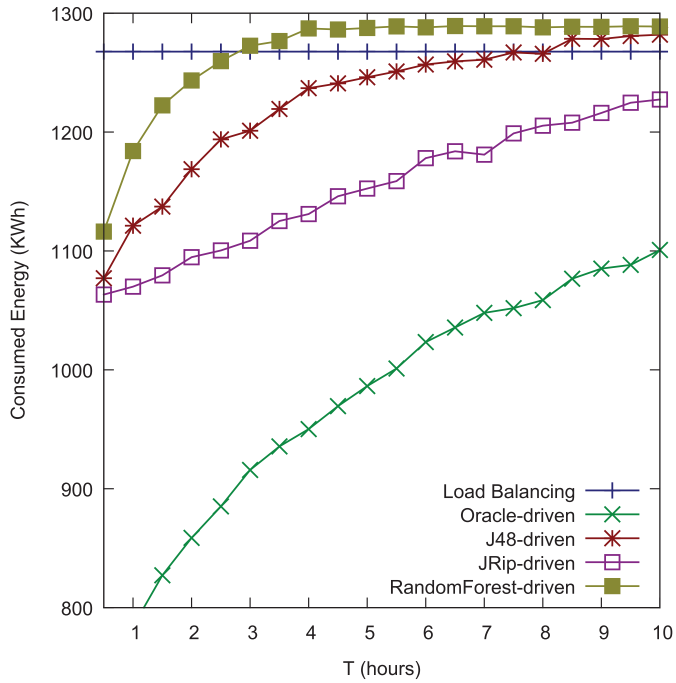


Figure 2. Consumed energy vs. T (hours), varying predictive algorithms, considering $\delta = 0.6$.

the cumulative CPU demand (at runtime) over a server exceeds that physically available on it, thus causing a computational overload and some performance degradation.

- *Virtual Machine Migrations*. Live migration of VMs allows transferring a VM between physical nodes without suspension and with a short downtime. However, live migration has a negative impact on the performance of applications running in a VM during a migration, because VM images must be transferred from source to destination nodes, and several memory pages (depending on the application) must be updated at both nodes during the course of migration.

In the next section, we will describe the results obtained through an extensive evaluation carried out in various experimental scenarios.

4.4. Results

To validate the energy-aware method presented in Section 3, we adopted as benchmark the testing dataset above described (see Section 4.2) and we performed our tests in different scenarios:

- *no-energy aware case*: this scenario corresponds to the case where no energy-aware policy is adopted; to do that, we ran the simulator in the hypothesis that the VM Migration Manager is disabled, thus reproducing resource needs and migrations among hosts of virtual machines as contained in the input file; in particular, the relocation policy adopted in the no-energy aware case is the *load balancing* strategy, which represents an useful *baseline* for a performance comparison with other approaches;
- *energy-aware case*: this scenario corresponds to the case where migrations of virtual machines are forced by the VM Migration Manager module of the architecture, on the basis of VM usage predictions; various tests have been carried out by varying the *consolidation period* T and the *threshold load* δ; moreover, as introduced in Section 4.2, several predictive algorithms have been used (JRip, J48, RandomForest), in order to provide a comparative analysis in terms of effectiveness and performance of the proposed approach;
- *oracle*: this scenario corresponds to the case where we suppose that the VM Migration Manager can query an oracle to have advance knowledge about resource sizes (cpu, ram) will be requested

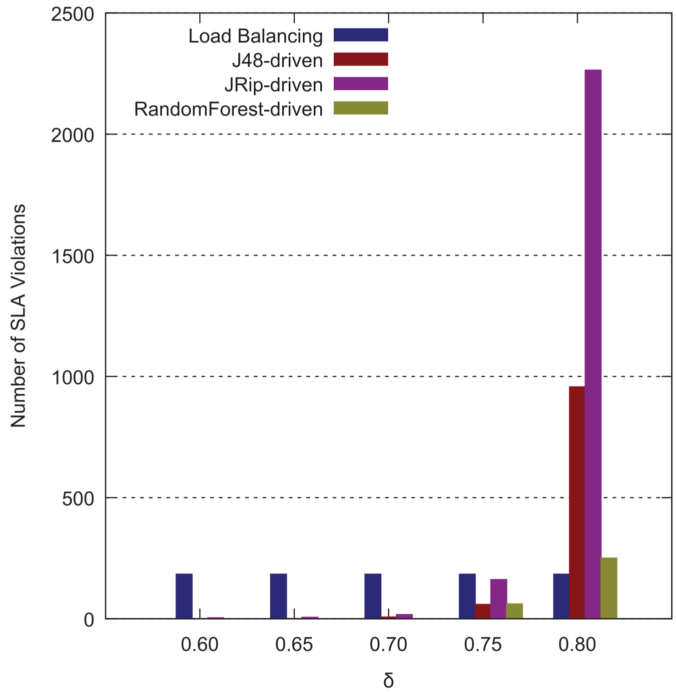
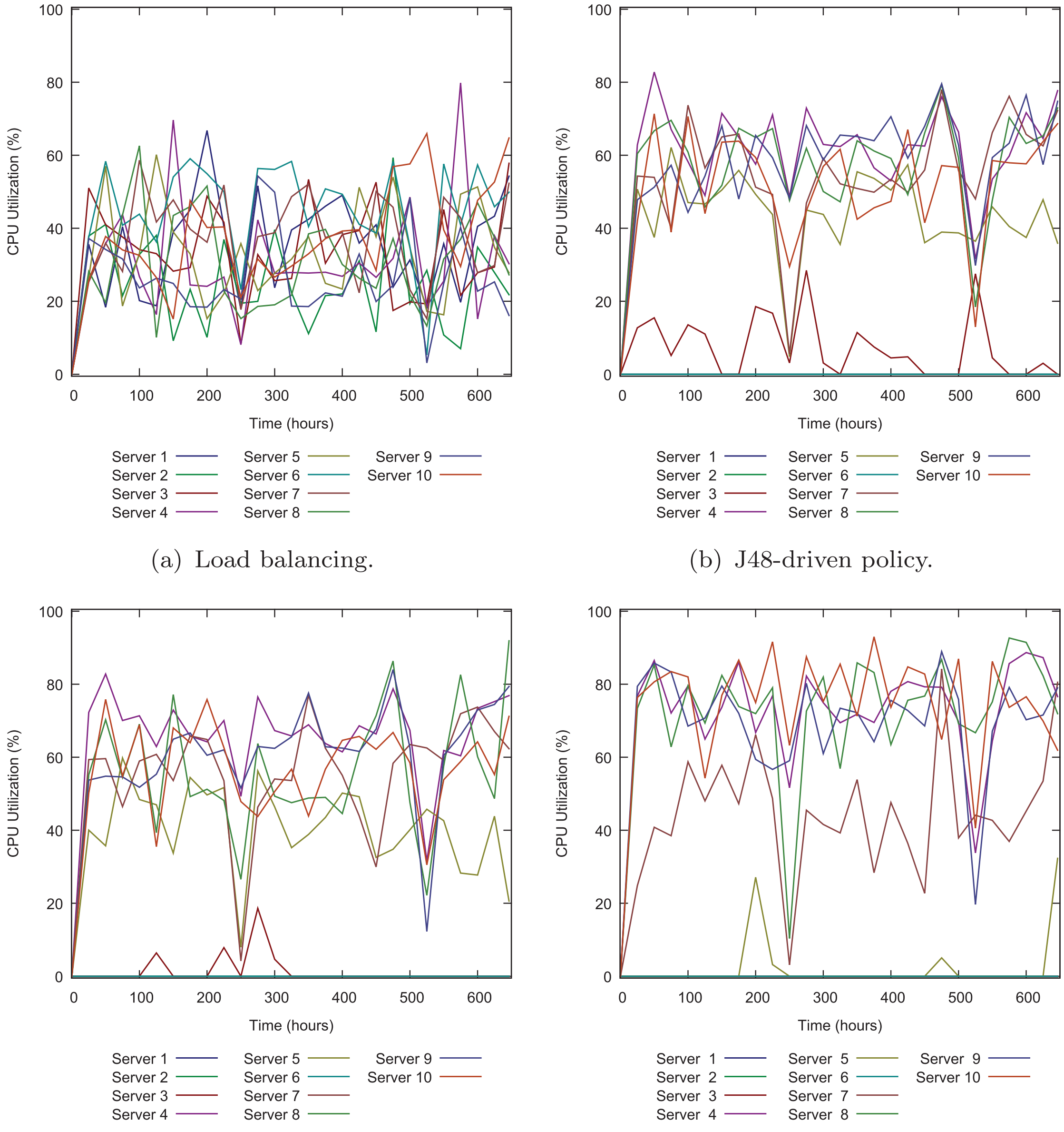


Figure 3. Number of SLA violations vs. resource limit (δ), considering $T = 0.5$.

in the future by each virtual machine, and can handle their relocations on the basis of such information; this case is obviously not applicable to reality, but it is a reference scenario showing the *best* result that we could achieve (in theory).

As a first result, let us show how the *consolidation period T* influences the energy consumption. Figure 2 plots the total energy consumed by the data centre over the whole simulation time, for reallocation periods T values ranging from 0.5 to 10 h. By observing the chart, we can make two main observations. First, the adopted energy-aware approaches obtain better results for shorter reallocation periods; in particular the lowest energy consumption is achieved for $T = 0.5$ h. This is a reasonable result, because shorter T values guarantee more frequent predictions of VM resource needs and, thus, more reactive VM reallocations to changing resource demands at runtime. In addition, this means that predictions of the consumption modeller have an effective impact on the overall system efficiency. Second, the chart shows that (as expected) the *load balancing* strategy (i.e. *no-energy aware* scenario) is not influenced by T, and that *oracle*-driven relocations achieve the best results with respect to the other policies. Finally, for small values of T the energy consumed by adopting *J48*-driven, *JRip*-driven and *RandomForest*-driven policies is lower than the *load balancing* case, thus showing the effectiveness of the data-driven relocation policies. In particular, since $T = 0.5$ h is the best relocation period value in terms of consumption in all cases, we will show further results considering this value setting.

Now, let us consider how the proposed data-driven approaches influence SLA violation events. Since virtual machine allocations performed in the energy-aware scenarios are based on resource predictions, it is possible that some of them are incorrect (e.g. the virtual machine CPU/RAM needs forecasted by the classifier is less than that will be used at runtime); this could lead to consolidate on a server a set of virtual machines whose cumulative CPU demand (at runtime) will exceed that physically available on it, by causing a SLA violation (and a consequent degradation of the QoS). Such a problem can be handled by cautiously considering a δ fraction of server resource availabilities as upper bound, thus saving the other $1 - \delta$ for unpredicted needs. Figure 3 reports the number of SLA violations vs. δ, for different predictive algorithms (fixed $T = 0,5$ h). As expected, the higher δ the higher SLA violations. In particular, the number of violations computed in the *load balancing*



(a) Load balancing.
(b) J48-driven policy.
(c) JRip-driven policy.
(d) Oracle-driven policy.

Figure 4. CPU utilisation (%) of each server vs. time (fixed $T = 0.5$ h and $\delta = 0.75$).

(i.e. no-energy aware) scenario is 184, and such value is comparable with the number of violations computed in the energy-aware scenarios when δ is fixed to 0.75 (59, 162, and 61 violations for J48, JRip and RandomForest, respectively). Thus, in order to provide results in similar QoS conditions for a fair performance comparison, in the following we will show further experimental results for $\delta = 0.75$ (i.e. 75% as resource usage limit).

As explained in Section 3, the architecture presented in the paper achieves energy saving by a more efficient CPU usage of the servers and by allocating all tasks on a minimum number of machines. In this way, some servers will get a higher load, but others will be unused for some periods and can be switched off, finally reducing the energy consumption. This phenomenon is clearly pointed out in Figure 4, which shows the CPU utilisation rate of the servers in the load balancing, oracle, and energy-aware cases, for $T = 0.5$ h and $\delta = 0.75$. We can observe that in the first scenario (see Figure 4(a)) all servers are turned on for the whole observed time period, with a CPU usage that is balanced

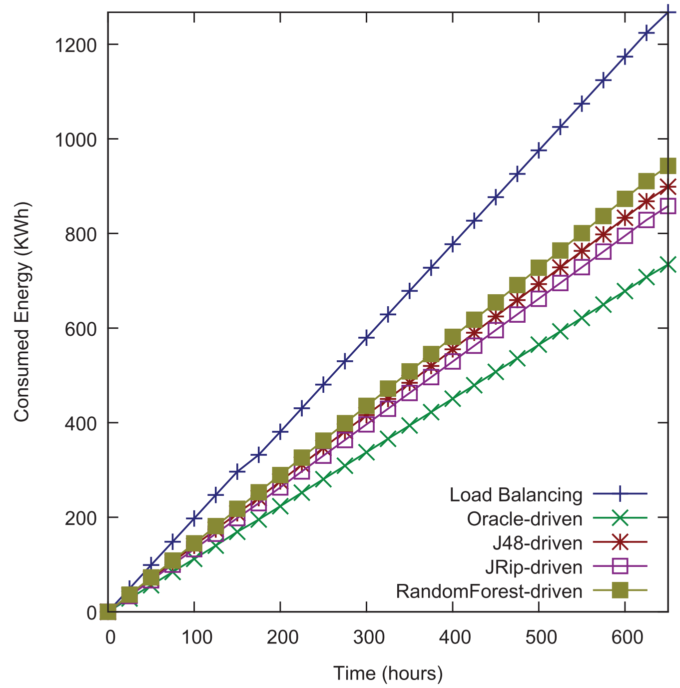


Figure 5. Cumulative consumed energy vs. time (fixed $T = 0.5$ h and $\delta = 0.75$).

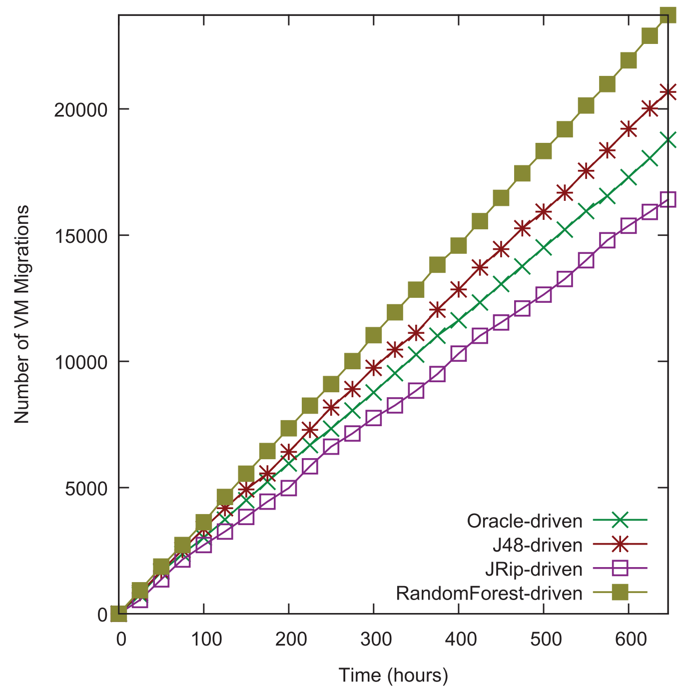


Figure 6. Cumulative migration number vs. time (fixed $T = 0.5$ h and $\delta = 0.75$).

between them and with some peaks depending on the variability of the running tasks needs. In this case, the average CPU usage is about 34.4%. In the second and third scenarios (see Figure 4(b) and (c)), where the J48-driven and JRip-driven energy-aware policies are adopted, it clearly appears that all the computational work is done by seven and six servers (respectively) which get higher loads with respect to the previous case, while the remaining three and four machines (respectively) are inactive. Indeed, this is the positive consequence of virtual machine consolidations, that is, reducing the number of physical servers turned on for all the time. In this case, considering the active servers, the average CPU usage is about 49.7% for J48 and 57.3% for JRip. Finally, Figure 4(d) shows the CPU utilisation rate of the servers, when migrations are planned in the oracle case. We can observe some differences with respect to the energy-aware case, that is worth noting more in detail. First, the CPU usage rate in the energy-aware case ranges from low values, i.e. 8%, to high values (one server uses even the 92% of its cpu power). In the oracle case, we can observe a more balanced cpu usage, where the most of servers (with some exceptions) exploit between 60 and 80% of their computational power, with an average CPU usage about 68.8%. Considering such results, we observe that the oracle strategy achieves two goals: (i) the lowest number of machines are running (only five machines) w.r.t. the other cases and (ii) the load capacity threshold of the servers is always respected (since migrations are driven by an oracle, there are no errors in predictions and no SLA violations).

The energetic effectiveness of the proposed energy-aware approach can be inferred from Figure 5, which shows the energy consumed by the data centre over the whole simulation time (646.5 h, corresponding to circa 27 days) in the different scenarios. The energy consumption has been computed by summing up the energy consumed by all the hosts. In particular, the figure shows consumed energy values for different algorithms exploited for relocation. We can observe that the *Load Balancing* strategy achieves the highest energy consumption, while the *Oracle* approach obtains the lowest energy consumption. The other policies (*J48*, *JRip*, and *RandomForest*) assess an energy consumption lower than the load balancing scenario. In particular, at the end of the experiments, the energy consumption is reduced from 1267.69 KWh (no energy-aware scenario) to 857.91 KWh (adopting the JRip-driven relocation, for $T = 0.5$ h), corresponding to circa 32.32% of saving in 27 days. In theory, by applying the oracle strategy, the energy consumption would amount to 734.41 KWh, corresponding to circa 42.07% of saving, which represents the (theoretical) best result that could be achieved.

Finally, Figure 6 shows the cumulative number of virtual machine migrations with respect to the time, fixed T = 0.5 h and for different data-driven relocation policies. Interestingly, the number of migrations ranges from 16,414 (JRip) to 23,718 (RandomForest). This means that, given $\delta = 0.75$ and $T = 0.5$, the JRip-driven relocation policy assesses the best results both in terms of number of migrations and energy consumption, at expense of only slightly more SLA violations (See Figure 3).

5. Conclusion

This paper presented a data-driven system for energy-aware allocation of virtual machines in a Cloud framework. Specifically, migrations of virtual machines are planned by predicting future computational needs (CPU, RAM) of each virtual machine through data mining algorithms, in order to plan their efficient allocations across the available servers. Experimental evaluation of the approach, performed on real-world Cloud data, shows encouraging benefits in terms of energy saving and SLA violations.

In future work, several research issues will be investigated. First, while this paper shows results related to a set of well-known classification algorithms, more advanced data-driven algorithms will be considered for the discovery of usage models, such as deep-learning models, Markov models, and sequential patterns. Moreover, it will be interesting to study some heuristic approaches that automatically fix the most effective values for the consolidation period T and load threshold δ. Finally, an extension of the presented approach, which discovers clusters of virtual machines having similar resource utilisation patterns and performs the consolidation task on the basis of this knowledge, will be investigated.

Disclosure statement

No potential conflict of interest was reported by the authors.

References

[1] Dong J, Jin X, Wang H, et al. Energy-saving virtual machine placement in cloud data centers. In: Proceedings of the 2013 13th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, CCGRID '13. Delft, Netherlands: IEEE Computer Society; 2013. p. 618–624.

[2] Cesario E, Talia D. Distributed data mining patterns and services: an architecture and experiments. Concurr Comput Pract Exp. 2012;24:1751–1774.

[3] Tseng HW, Yang TT, Yang KC, et al. An energy efficient vm management scheme with power-law characteristic in video streaming data centers. IEEE Trans Parallel Distrib Syst. 2017;PP:1–1.

[4] Cicirelli F, Guerrieri A, Spezzano G, et al. An edge-based platform for dynamic smart city applications. Future Gener Comput Syst. 2017;76:106–118.

[5] Mastroianni C, Meo M, Papuzzo G. Self-economy in cloud data centers: statistical assignment and migration of virtual machines. In: Proceedings of the 17th International Conference on Parallel Processing - Volume Part I, Euro-Par'11. Berlin, Heidelberg: Springer-Verlag; 2011. p. 407–418.

[6] Barroso LA, Hölzle U. The case for energy-proportional computing. Computer. 2007;40:33–37.

[7] Dong D, Herbert J. Energy efficient vm placement supported by data analytic service. In: Proceedings of the 2013 13th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, CCGRID '13; IEEE Computer Society; 2013. p. 648–655.

[8] Greenberg A, Hamilton J, Maltz DA, et al. The cost of a cloud: research problems in data center networks. SIGCOMM Comput Commun Rev. 2008;39:68–73.

[9] Altomare A, Cesario E, Talia D. Energy-aware migration of virtual machines driven by predictive data mining models. In: Proceedings of the 23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, CCGRID '10. Turku, Finlan: IEEE Computer Society; 2015. p. 549–553.

[10] Altomare A, Cesario E. A data-driven approach based on auto-regressive models for energy-efficient clouds. In: Proceedings of the 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGRID 2017. Madrid, Spain: IEEE; 2017. p. 1062–1069.

[11] Srikantaiah S, Kansal A, Zhao F. Energy aware consolidation for cloud computing. In: Proceedings of the 2008 Conference on Power Aware Computing and Systems, HotPower'08. San Diego (CA): USENIX Association; 2008. p. 10–10.

[12] Piraghaj SF, Calheiros RN, Chan J, et al. Virtual machine customization and task mapping architecture for efficient allocation of cloud data center resources. Comput J. 2016;59:208–224.

[13] Subirats J, Guitart J. Assessing and forecasting energy efficiency on cloud computing platforms. Future Gener Comput Syst. 2015;45:70–94.

[14] Cardosa M, Korupolu MR, Singh A. Shares and utilities based power consolidation in virtualized server environments. In: Proceedings of the 11th IFIP/IEEE International Conference on Symposium on Integrated Network Management, IM'09. Long Island (NY): IEEE Press; 2009. p. 327–334.

[15] Mazzucco M, Dyachuk D, Deters R. Maximizing cloud providers' revenues via energy aware allocation policies. In: Proceedings of the 2010 IEEE 3rd International Conference on Cloud Computing, CLOUD '10; IEEE Computer Society; 2010. p. 131–138.

[16] Chase JS, Anderson DC, Thakar PN, et al. Managing energy and server resources in hosting centers. In: Proceedings of the Eighteenth ACM Symposium on Operating Systems Principles, SOSP '01. New York (NY): ACM; 2001. p. 103–116.

[17] Elnozahy EN, Kistler M, Rajamony R. Energy-efficient server clusters. In: Proceedings of the 2nd International Conference on Power-aware Computer Systems, PACS'02. Berlin, Heidelberg: Springer-Verlag; 2003. p. 179–197.

[18] Pinheiro E, Bianchini R, Carrera EV, et al. Load balancing and unbalancing for power and performance in cluster-based systems. In: Proceedings of the Workshop on Compilers and Operating Systems for Low Power, COLP'01; 2001. p. 182–195.

[19] Tchana A, Palma ND, Safieddine I, et al. Software consolidation as an efficient energy and cost saving solution. Future Gener Comput Syst. 2016;58:1–12.

[20] Zhang Q, Chen H, Shen Y, et al. Optimization of virtual resource management for cloud applications to cope with traffic burst. Future Gener Comput Syst. 2016;58:42–55.

[21] VMware vCenter Server [Online; accessed 2014 Aug 30]. Available from: http://www.vmware.com/products/vcenter-server/

[22] Beloglazov A, Abawajy J, Buyya R. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future Gener Comput Syst. 2012;28:755–768.

[23] Voorsluys W, Broberg J, Venugopal S, et al. Cost of virtual machine live migration in clouds: a performance evaluation. In: Proceedings of the 1st International Conference on Cloud Computing, CloudCom '09. Berlin, Heidelberg: Springer-Verlag; 2009. p. 254–265.

[24] Tan PN, Steinbach M, Kumar V. Introduction to data mining. Boston: Addison-Wesley; 2005.

[25] Quinlan JR. C4.5: programs for machine learning. San Francisco (CA): Morgan Kaufmann Publishers Inc.; 1993.

[26] Cohen WW. Fast effective rule induction. In: Twelfth International Conference on Machine Learning. Tahoe City (CA): Morgan Kaufmann; 1995. p. 115–123.

[27] Breiman L. Random forests. Mach Learn. 2001;45:5–32.
[28] Beloglazov A, Buyya R. Energy efficient allocation of virtual machines in cloud data centers. In: Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, CCGRID '10. Washington (DC): IEEE Computer Society; 2010. p. 577–578.
[29] Gandhi A, Harchol-Balter M, Das R, et al. Optimal power allocation in server farms. SIGMETRICS Perform Eval Rev. 2009;37:157–168.
[30] Kusic D, Kephart JO, Hanson JE, et al. Power and performance management of virtualized computing environments via lookahead control. In: Proceedings of the 2008 International Conference on Autonomic Computing, ICAC '08; 2008. p. 3–12.
[31] Verma A, Ahuja P, Neogi A. pMapper: power and migration cost aware application placement in virtualized systems. In: Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware, Middleware '08; Springer-Verlag New York Inc; 2008. p. 243–264.