CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 0000; **00**:1–18
Published online in Wiley InterScience (www.interscience.wiley.com). DOI: 10.1002/cpe

DVFS-Aware Dynamic Consolidation of Virtual Machines for Energy Efficient Cloud Data Centers

Patricia Arroba[1 2*], José M. Moya[1 2], José L. Ayala[3], Rajkumar Buyya[4]

[1] *Laboratorio de Sistemas Integrados (LSI), Departamento de Ingeniería Electrónica, Universidad Politécnica de Madrid, Spain*
[2] *CCS - Center for Computational Simulation, Universidad Politécnica de Madrid, Spain*
[3] *DACYA, Universidad Complutense de Madrid, Spain*
[4] *Cloud Computing and Distributed Systems (CLOUDS) Laboratory, Department of Computing and Information Systems, The University of Melbourne, Australia*

SUMMARY

Computational demand in data centers is increasing due to the growing popularity of Cloud applications. However, data centers are becoming unsustainable in terms of power consumption and growing energy costs so Cloud providers have to face the major challenge of placing them on a more scalable curve. Also, Cloud services are provided under strict Service Level Agreement conditions, so trade-offs between energy and performance have to be taken into account. Techniques as DVFS and consolidation are commonly used to reduce the energy consumption in data centers, although they are applied independently and their effects on Quality of Service are not always considered. Thus, understanding the relationship between power, DVFS, consolidation and performance is crucial to enable energy-efficient management at the data center level. In this work we propose a DVFS policy that reduces power consumption while preventing performance degradation, and a DVFS-aware consolidation policy that optimizes consumption, considering the DVFS configuration that would be necessary when mapping VMs to maintain QoS. We have performed an extensive evaluation on the CloudSim toolkit using real Cloud traces and an accurate power model based on data gathered from real servers. Our results demonstrate that including DVFS awareness in workload management provides substantial energy savings of up to 41.62% for scenarios under dynamic workload conditions. These outcomes outperforms previous approaches, that do not consider integrated use of DVFS and consolidation strategies. Copyright © 0000 John Wiley & Sons, Ltd.

Received . . .

KEY WORDS: Energy optimization; Green data centers; Cloud computing; DVFS; Dynamic consolidation

1. INTRODUCTION

The trend towards Cloud computing has lead to the proliferation of data centers since they are the infrastructure that provides this new paradigm of computing and storage. Reference companies such as Amazon [1], Google [2], Microsoft [3], and Apple [4] have chosen this computational model where information is stored in the Internet Cloud offering services more quickly and efficiently to the user. Cloud market opportunities in 2013 were supposed to achieve up to $150 billion [5], but the rising price of energy had an impact on the costs of Cloud infrastructures, increasing the Total Cost of Ownership (TCO) and reducing the Return on Investment (ROI).

Nowadays, data centers consume about 2% of the worldwide energy production [6], originating more than 43 million tons of CO_2 per year [7]. Also, the proliferation of urban data centers is

*Correspondence to: Email:parroba@die.upm.es

Prepared using cpeauth.cls *[Version: 2010/05/13 v3.00]*

responsible for the increasing power demand of up to 70% in metropolitan areas, where the power density is becoming too high for the power grid [8]. In two or three years, the 95% of urban data centers will experience partial or total outages incurring in annual costs of about US$2 million per infrastructure. The 28% of these service outages are expected to be due to exceeding the maximum capacity of the grid [9].

Besides the economical impact, the heat and the carbon footprint generated by cooling systems are dramatically increasing and they are expected to overtake the emissions of the airline industry by 2020 [10]. The cooling infrastructure is needed to ensure that IT operates within a safe range of temperatures, ensuring reliability. The energy efficiency of novel cooling technologies, as water-based cooling, heat reusing and free cooling approaches outperform traditional Computer Room Air Conditioning (CRAC) units. However, the implantation rate of these new techniques is still low for typical data centers.

However, the main contributor to the energy consumption in a data center is the IT infrastructure, which consists of servers and other IT equipment. The IT power in the data center is dominated by the power consumption of the enterprise servers, representing up to 60% of the overall data center consumption [7]. The power usage of an enterprise server can be divided into dynamic and static contributions. Dynamic power depends on the switching transistors in electronic devices during workload execution. Static consumption associated to the power dissipation of powered-on servers represents around 70% and is strongly correlated with temperature due to the leakage currents that increase as technology scales down.

The Cloud helps improving energy efficiency, reducing the carbon footprint per executed task and diminishing CO2 emissions [11] by increasing data centers overall utilization. The main reason is that, in this computational model, the computing resources are shared among users and applications so, less powered-on servers are needed, which means less static consumption. In this way, smaller facilities are able to consolidate higher incoming workloads, thus reducing the computing and cooling energy requirements. Cloud computing provided a 17% reduction in energy consumption by 2011 according to the Schneider Electric's report on virtualization and Cloud computing efficiency [12].

To meet the growing demand for their services and ensure minimal costs, Cloud providers need to implement an energy-efficient management of physical resources. Therefore, optimization approaches that rely on accurate power models and optimize the configuration of server parameters (voltage and working frequency, workload assignment, etc.) can be devised. Furthermore, as many applications expect services to be delivered as per Service Level Agreement (SLA), power consumption in data centers may be minimized without violating these requirements whenever it is feasible.

From the application-framework viewpoint, Cloud workloads present additional restrictions as 24/7 availability, and SLA constraints among others. In this computation paradigm, workloads hardly use 100% of CPU resources, and their execution time is strongly constrained by contracts between Cloud providers and clients. These restrictions have to be taken into account when minimizing energy consumption as they impose additional boundaries to efficiency optimization strategies. Users' Quality of Experience (QoE) would be determined by these constraints and it would be impacted by performance degradation.

Also, Cloud scenarios present workloads that vary significantly over time. This fluctuation hinders the optimal allocation of resources, that requires a trade-off between consolidation and performance. Workload variation impacts on the performance of two of the main strategies for energy-efficiency in Cloud data centers: Dynamic Voltage and Frequency Scaling (DVFS) and Consolidation.

DVFS strategies modify frequency according to the variations on the utilization performed by dynamic workload. These policies help to dynamically reduce the consumption of resources as dynamic power is frequency-dependent. DVFS has been traditionally applied to decrease the power consumption of underutilized resources as it may incur on SLA violations. On the other hand, consolidation policies decrease significantly the static consumption by reducing the number of active servers, increasing their utilization. Dynamic workload scenarios would require policies to adapt the operating server set to the workload needs during runtime in order to

DOI: 10.1002/cpe

minimize performance degradation due to overprovisioning. However, both strategies are applied independently, regardless the effects that consolidation have on DVFS and vice versa. Therefore, the implementation of DVFS-aware consolidation policies has the potential to optimize the energy consumption of highly variable workloads in Cloud data centers.

The **key contributions** of our work are 1) a DVFS policy that takes into account the trade-offs between energy consumption and performance degradation; 2) a novel consolidation algorithm that is aware of the frequency that would be necessary when allocating a Cloud workload in order to maintain QoS. Our frequency-aware consolidation strategy reduces the energy consumption of the data center, making use of DVFS to reduce the dynamic power consumption of servers, also ensuring SLA. The algorithm is light and offers an elastic scale-out under varying demand of resources.

The rest of the paper is organized as follows: Section 2 gives further information on the related work on this topic. The proposed DVFS policy that considers both energy consumption and performance degradation is presented in Section 3. This section also provides our Frequency-Aware Optimization strategy for the energy optimization of data centers. The simulation configuration is detailed in Section 4. In Section 5, we describe profusely the performance evaluation and the experimental results. Finally, Section 6 concludes the work with future directions.

2. RELATED WORK

Recently, there has been a growing interest in developing techniques to provide power management for servers operating in a Cloud. The complexity of the power management and workload allocation in servers has been described by Gandhi et al. [13] and Rafique et al. [14] , where the authors show that the optimal power allocation is non-obvious, and, in fact, depends on many factors such as the power-to-frequency relationship in processors, or the arrival rate of jobs. Thus, it is critical to understand quantitatively the relationship between power consumption and DVFS at the system level to optimize the use of the deployed Cloud services.

DVFS is by far the most frequent technique at the architectural-level as well as one of the currently most efficient methods to achieve energy savings. This technique scales power according to the workload in a system by reducing both operating voltage and frequency. Reducing the operating frequency and voltage slows the switching activity achieving energy savings but also decreasing the system performance. DVFS implementation on CPU results in an almost linear relationship between power and frequency, taking into account that the set of states of frequency and voltage of the CPU is limited. Only by applying this technique on a server CPU, up to 34% energy savings in dynamic consumption can be reached as presented in [15].

DVFS has been mainly applied to enhance energy efficient scheduling on idle servers, or those performing under light workload conditions [16], and during the execution of noncritical tasks [17]. However, a recent research shows that DVFS can be also used to meet deadlines in mixed-criticality systems [18]. Furthermore, DVFS-based scheduling research on multiprocessor systems shows promising results. Rizvandi et al. [19] achieved considerable energy savings by applying this technique on up to 32-processor systems for HPC workload. However, the effects of DVFS in loaded servers have not been analyzed yet for Cloud scenarios. The Quality of Experience (QoE) offered to the user that depends on the SLA contracted to Cloud providers could be violated under certain frequency-voltage conditions. DVFS-aware approaches could help to reduce the energy consumption of Cloud facilities but new algorithms have to be devised for large scale data center infrastructures also taking into account the SLA considerations of Cloud workloads.

On the other hand, many of the recent research works have focused on reducing power consumption in cluster systems by power-aware Virtual Machine (VM) consolidation techniques, as they help to increase resource utilization in virtualized data centers. Consolidation uses virtualization to share resources, allowing multiple instances of operating systems to run concurrently on a single physical node. Virtualization and consolidation increase hardware utilization (up to 80% [20]) thus improving resource efficiency.

The resource demand variability of Cloud workload is a critical factor in the consolidation problem as the performance degradation boundary has to be considered for both migrating VMs and

DOI: 10.1002/cpe

reducing the active server set [21]. Balancing the resource utilization of servers during consolidation was performed by Calheiros et al. [22] to minimize power consumption and resource wastage. In the research proposed by Hermenier et al. [23], their consolidation manager reduces the VM migration overhead. Also, there exist interesting works that focuses on modeling the energy consumption of the migration process as the research proposed by Haikun et al. [24] and De Maio et al. [25].

However, DVFS-Aware consolidation techniques that maintain QoS in data centers have not been fulfilled yet. Although some combined application of DVFS and consolidation methods for Cloud environments can be found, no one of them are considering performance degradation due to VM migration or resource over-provisioning. In the research presented by Wang et al. [26], the consolidation is performed regardless the frequency impact, and the DVFS is applied separately. The approach presented by Petrucci et al. [27] shows the dependence of power with frequency but the algorithm does not scale for large data centers and SLA violations are not taken into account.

Our work provides a novel DVFS-aware consolidation algorithm that helps to reduce the energy consumption of data centers under dynamic workload conditions. The proposed strategy considers the trade-offs between energy consumption and performance degradation thus maintaining QoS. The work presented in this paper outperforms previous contributions by allowing the optimization of Cloud data centers from a proactive perspective in terms of energy consumption and ensuring the user experience of Cloud-based services.

3. FREQUENCY-AWARE VM CONSOLIDATION

The major challenge that we face in this work is to reduce the energy consumption of the IT infrastructure of data centers, while maintaining QoS, and under dynamic workload conditions. In our previous work [28] we derived a complete accurate model to calculate the total energy consumption of a server $E_{\mathrm{host}}(m,k)$ in $kW \cdot h$ that can be seen in Equation 1:

$$\begin{aligned}
E_{\mathrm{host}}(m,k) &= P_{\mathrm{host}}(m,k)\cdot\Delta t = (P_{\mathrm{dyn}}(m,k)+P_{\mathrm{stat}}(m,k))\cdot\Delta t && (1)\\
T &= \{t_1,...,t_i,...,t_T\} && (2)\\
\Delta t &= t_{i+1}-t_i && (3)\\
P_{\mathrm{dyn}}(m,k) &= \alpha(m)\cdot V_{\mathrm{DD}}^2(m,k)\cdot f_{\mathrm{op}}(m,k)\cdot u_{\mathrm{cpu}}(m) && (4)\\
P_{\mathrm{stat}}(m) &= \beta(m)\cdot T_{\mathrm{mem}}^2(m)+\gamma(m)\cdot FS^3(m) && (5)
\end{aligned}$$

where Δt is the time along which the energy is calculated. In this paper we assume a discrete set of times T in which the algorithm is evaluated in order to optimize the power performance of the system. We define each time t_i as the particular instant in which the system evaluates an incoming batch of workload. Our proposed model estimates the instantaneous electric power of a server in t_i so, the energy is computed for the time interval Δt between two workload evaluations, considering that the power is stable in this time period. For practical reasons, we have selected Δt to be $300s$ in our experiments, which is a realistic assumption for our setup.

$P_{\mathrm{host}}(m,k)$, $P_{\mathrm{dyn}}(m,k)$ and $P_{\mathrm{stat}}(m,k)$ represent total, dynamic and static contributions of the power consumption in Watts of the physical machine m operating in a specific k DVFS mode. $\alpha(m)$, $\beta(m)$ and $\gamma(m)$ define the technological constants of the server in the range of X, $X\cdot 10^{-3}$ and $X\cdot 10^{-11}$.

Our proposed model consists of 5 different variables: $u_{\mathrm{cpu}}(m)$ is the averaged CPU percentage utilization of the specific server m and is proportional to the number of CPU cycles defined in Millions of Instructions Per Second (MIPS) in the range [0,1]. V_{DD} is the CPU supply voltage and f_{op} is the operating frequency in GHz. T_{mem} defines the averaged temperature of the main memory in Kelvin and FS represents the averaged fan speed in RPM. Depending on the target architecture, some factors might have higher impact than others. This model has been validated for Intel architectures achieving accuracy results of about 95%.

Our model allows to obtain power estimations during run-time facilitating the integration of proactive strategies in real scenarios. Power consumption is measured with a current clamp, so we

DOI: 10.1002/cpe

can validate our approach comparing our estimations with real values, obtaining validation errors ranging from 4.22% to 4.94%. *perf* monitoring tool is used to collect the values acquired by different hardware counters in order to monitor different parameters of the CPU and the memory. $u_{\mathrm{cpu}}(m)$ has been measured by monitoring the execution of the virtual machines with the *ps aux* Linux command. *cpufreq-utils* Linux package is used to monitor and modify f_{op} and V_{DD} according to the DVFS states during workload execution. Finally, T_{mem} and FS are collected via *IPMI*.

As shown in Equation 4, the energy consumption due to the dynamic power consumption $P_{\mathrm{dyn}}(m,k)$ depends on the workload profile executed in the server. So, the lower the $u_{\mathrm{cpu}}(m)$, the lower the dynamic energy contribution. On the other hand, the static consumption represents the energy consumed due to power dissipation of a powered-on server, even if it is idle. This energy represents around 70% of the total server consumption. In this context, we can see some observations about the dynamic consolidation problem:

- **DVFS vs SLA.** DVFS can be used to achieve power savings because reducing the frequency and voltage of the CPU ($f_{op}(m,k)$ and $V_{DD}(m,k)$) slows its switching activity. However, it also impacts on the performance of the system by extending tasks duration (t), which can lead to the appearance of SLA violations and to the increase of energy consumption.
- **Underloaded servers.** If the workload is spread over a larger number of servers, the CPU utilization in each server will be lower, so the dynamic power contribution in each server will be also lower. As u_{cpu} is reduced, f_{op} can be scaled thus decreasing the power contribution due to CPU frequency. However, the global energy consumption will be increased disproportionally due to the impact of static consumption of a higher number of servers.
- **Overloaded servers.** On the other hand, if the incoming workload is concentrated in a smaller set of servers, even though the static consumption is reduced, the QoS may be affected. This situation is intensified due to the dynamic variation of workload and, if the maximum server capacity is exceeded during peak loads, it would lead to performance degradation. To avoid overloaded servers, one or more VMs can be migrated from one server to another. However, VM migration has associated costs in terms of energy consumption and time, which could lead to SLA violations.

In this paper we propose a strategy to allow the energy optimization of a Cloud under SLA constraints. As opposed to previous approaches, our work offers a DVFS policy that considers the trade-offs between energy consumption and performance degradation explained in subsection 3.1. Thus, frequency is managed according to the available states depending on the server architecture while ensuring QoS. On the other hand, in subsection 3.2 we provide an energy-aware dynamic placement algorithm that considers the frequency configuration according to the allocation of VMs. Finally, in subsection 3.3 we use both strategies combined to proactively optimize a Cloud under dynamic workload conditions.

3.1. DVFS-Performance Management

DVFS scales the power of the system varying both CPU frequency and voltage. Reducing the operating frequency and voltage slows the switching activity to achieve energy savings; however, it also impacts negatively on the performance of the system.

The CPU performance of a physical machine m is characterized by the maximum MIPS that can be provided by its CPU ($MIPS_{MAX}(m)$) at its maximum frequency ($f_{MAX}(m)$). Moreover, the real available cycles offered by the system ($MIPS_{available}(m,k)$) depend on the current operating frequency of the CPU ($f_{op}(m,k)$). As we present in Equation 6, $f_{op}(m,k)$ can only take a value from a specific set of valid frequencies where k represents the operating DVFS mode. It is important to note that not all frequencies from 0 to $f_{MAX}(m)$ are available, as the set of states of frequency and voltage of the CPU is limited and it may be different depending on the architecture of the physical machine m. We define the available CPU utilization percentage $u_{cpu_{available}}(m,k)$ as the maximum CPU utilization that could be used by the workload without performance degradation. We present the relationship between these parameters in Equation 8.

DOI: 10.1002/cpe

$$f_{op}(m,k) \in \{f_1(m), f_2(m), \cdots, f_k(m), \cdots, f_{MAX}(m)\} \tag{6}$$

$$MIPS_{available}(m,k) = \frac{f_{op}(m,k)}{f_{MAX}(m)} \cdot MIPS_{MAX}(m) \tag{7}$$

$$u_{cpu_{available}}(m,k) = \frac{MIPS_{available}(m,k)}{MIPS_{MAX}(m)} = \frac{f_{op}(m,k)}{f_{MAX}(m)} \tag{8}$$

In order to motivate these metrics we provide the following case of use for the Fujitsu RX300 S6 server. The maximum frequency for this type of physical machine is $f_{MAX}(Fujitsu) = 2.4GHz$, so the maximum number of instructions per second is $MIPS_{MAX}(Fujitsu) = 2400 \cdot CPI$, where CPI defines the number of cycles per required per instruction. One of the available operating frequencies for this server is $f_{op}(Fujitsu, 1) = f_1(Fujitsu) = 1.73GHz$. Assuming that the server is operating at $1.73GHz$, $MIPS_{available}(Fujitsu, 1) = 1.73/2.4 \cdot 2400 \cdot CPI = 1730 \cdot CPI$, and according to Equation 8, the available CPU utilization percentage takes the value $u_{cpu_{available}}(Fujitsu, 1) = 0.72$. Thus, if the utilization of the Fujitsu server, running at $1.73GHz$, exceeds the 72% of its total capacity, the required number of instructions to be executed will be higher than the available MIPS than can be provided, provoking a delay. On the other hand, when the utilization is kept below this threshold, no performance degradation occurs due to DVFS. These quality and performance metrics will be considered by the proposed energy optimization algorithm, so that they are not degraded (as it will be confirmed by the experimental results).

Our proposed DVFS management policy (DVFS-perf), presented in Algorithm 1 takes into account the previous relationships in order to improve energy efficiency, avoiding performance degradation. As inputs, we consider the complete set of hosts (*hostList*), and the set of valid frequencies (*frequenciesList*). For each host, the current value of CPU *utilization* is acquired in step 4. This variable, which depends on the workload that is already hosted and running in the server, is monitored during runtime by using calls to the system utility (e.g. *Linux top*). Then, $u_{cpu_{available}}(m,k)$ (*availableUtilization*) is calculated for the different frequencies in *frequenciesList* in steps 5 to 9. The algorithm selects the minimum frequency that offers a suitable $u_{cpu_{available}}(m,k)$ value that is greater or equal to the current *utilization* in the host in step 7. Finally, the DVFS configuration for the entire set of hosts is provided by *frequencyConfiguration*.

Algorithm 1 DVFS-perf configuration

```
Input: hostsList, frequenciesList
Output: frequencyConfiguration
 1: frequenciesList.sortIncreasingFrequency()
 2: maxFrequency ← frequenciesList.getMax()
 3: foreach host in hostList do
 4:    utilization ← host.getUtilization()
 5:    foreach frequency in frequenciesList do
 6:       availableUtilization ← frequency / maxFrequency
 7:       if availableUtilization ≥ utilization then
 8:          frequencyConfiguration.add(frequency)
 9:          break
10: return frequencyConfiguration
```

As dynamic power is reduced with frequency, our algorithm sets the operating frequency of each host to the lowest available value that provides sufficient CPU resources according to Equation 8. This ensures that the server offers sufficient MIPS based on the amount demanded by the allocated workload satisfying QoS.

We motivate these metrics by providing a case of use based on a Fujitsu RX300 S6 server, whose *maxFrequency* is $2.4GHz$. The operating frequencies set (*frequenciesList*) in GHz is $f_{op}(Fujitsu, k) = \{1.73, 1.86, 2.13, 2.26, 2.39, 2.40\}$. Our aim is to find the best *frequencyConfiguration* for a current *utilization* of the server of 80%. First, we calculate the *availableUtilization* for the minimum frequency according to equation 8 obtaining $u_{cpu_{available}}(Fujitsu, 1) = 1.73/2.4 = 0.721$. As 72% is lower than 80%, this frequency is discarded so the algorithm check the next one in an increasing order. $u_{cpu_{available}}(Fujitsu, 2) = 1.86/2.4 = 0.775$ is also lower so the next frequency is evaluated. For $f_{op}(Fujitsu) = 2.13GHz$, we calculate $u_{cpu_{available}}(Fujitsu, 3) = 2.13/2.4 = 0.887$ obtaining an available CPU utilization of 88.7%, that is higher than the 80% required by the workload allocated in it. Thus, our algorithm sets the frequency of the Fujitsu server running at 80% to 2.13GHz, as it is the minimum frequency that allows running the workload without performance degradation due to DVFS.

This policy allows servers to execute the workload in a more efficient way in terms of energy as frequency is scaled depending on the CPU requirements of the workload, while maintaining QoS.

3.2. *Frequency-Aware Dynamic Consolidation*

As an alternative to previous approaches, in this research we provide an energy-aware consolidation strategy that considers the frequency configuration according to the allocation of VMs. We use this approach to proactively optimize a Cloud under dynamic workload conditions.

3.2.1. Dynamic Consolidation Outlook. In this context, the dynamic consolidation problem can be split into four different phases, as proposed by Beloglazov et al. [29]. Each phase considers (i) detection of overloaded and (ii) underloaded hosts, (iii) selection of VMs to be migrated from these hosts, and (iv) VM placement after migrations respectively. Their research also present different algorithms for optimizing phases (i)-(iii) that we use during performance evaluation (see Subsections 4.4.1 and 4.4.2). So our work will be focused on finding new placements to host VMs after their migration from underloaded and overloaded hosts. In this work, we aim to optimize VM placement taking into account the frequency variations caused by the workload allocation together with the estimation of its impact in the overall consumption. This premise is incorporated in our policy, and evaluated lately in terms of energy efficiency and performance.

3.2.2. Algorithm Considerations. One of the main challenges when designing data center optimizations is to implement fast algorithms that can be evaluated for each workload batch during run-time. For this reason, the present research is focused on the design of an optimization algorithm that is simple in terms of computational requirements, in which both decision making and its implementation in a real infrastructure are fast. Instead of developing an algorithm for searching the optimal solution, we propose a sequential heuristic approach because it requires lower computational complexity. Our solution scales properly in accordance with large numbers of servers as explained in Subsection 3.2.3.

Minimizing the overall power consumption of the data center as a whole by only considering the consumption of each server separately may drive to some inefficiencies. The dynamic power of a host depends linearly on the CPU utilization, while the static remains constant (see Equation 1). So, when the reduction in consumption is performed individually, server by server, it results in the allocation of less workload on each physical machine, leading to the *underloaded server*-issue. This situation increases the number of active servers, that become underutilized, regardless the increase in the global static consumption. Otherwise, if the total energy consumed by the infrastructure is considered to be optimized, increasing the CPU utilization will reduce the number of servers required to execute the workload thus decreasing the overall static consumption but leading to an *overloaded server*-scenario. Therefore, both QoS and energy consumption could be affected as a consequence of VM migrations.

The proposed power and performance considerations, in Equations 1-5 and 6-8 respectively, provide a better understanding of how the system's behavior varies depending on frequency and utilization simultaneously. According to this, a more proactive allocation policy could be devised using DVFS to dynamically constrain aggressive consolidation scenarios to preserve QoS. To this purpose, the trade-offs between CPU utilization and frequency have to be analyzed in terms of energy. An increase in the resource demand of a host in terms of MIPS could represent an increment

DOI: 10.1002/cpe

in its frequency depending on the available set of frequencies to maintain QoS. If frequency needs to be risen, the power consumption will be increased due to the frequency contribution (see Equation 6). So, we propose a VM placement policy that estimates the frequency increment during workload consolidation. Our strategy decides to allocate workload in those servers that have a higher utilization (but still have resources left to accommodate the incoming VM) and that impact less on the frequency contribution. Consequently, the policy uses more efficiently the ranges of utilization in which the frequency is not increased.

3.2.3. DVFS-Aware Dynamic Placement. The policy proposed in this research is not only aware of the utilization of the incoming workload to be assigned, but also is conscious of the impact of its allocation on servers working at different frequencies. DVFS-awareness allows to predict operating frequencies depending on VM allocation, thus helping to estimate future energy contributions. The presented approach takes advantage of this knowledge to optimize VM placement within the Cloud infrastructure under QoS and energy constraints.

Our algorithm is based on the bin packing problem, where servers are represented as bins with variable sizes due to the frequency scaling. To solve this NP-hard problem we use a Best Fit Decreasing (BFD)-based algorithm as BFDs are shown to use no more than $11/9 \cdot OPT + 1$ bins [30], being OPT the bins provided by the optimal solution. The bin packing approach under similar conditions has been proved to work well for this type of problems with large server sets of 800 hosts [29].

The allocation of a VM in a specific host provokes an increase in its CPU utilization and, according to our proposed *DVFS-perf configuration* algorithm, may increase or not its operating frequency. According to our previous considerations, a tradeoff between servers' utilization and frequency may be inferred to reduce energy consumption of dynamic workload scenarios. Typically, the frequency span in which a CPU ranges is of about 1GHz. So, the difference between a frequency of the set of valid frequencies and the next one is 0.X, being more common steps of about 0.1-0.5 GHz. On the other hand, average cloud workload utilization ranges from 16%-59% [31]. As we define utilization of CPU as a value that ranges from 0 to 1, average cloud workload utilization would be in the range 0.16-0.59. Thus utilization and frequency increments originated by the allocation of VMs have the same orders of magnitude. So, in order to maximize servers' utilization while minimizing frequency increment, we propose to maximize the difference between these two parameters as can be seen in Equation 9. We avoid the use of normalization, providing a light algorithm. We mean that our proposed algorithm is light because, compared to tests that we have conducted with metaheuristics as Simulated Annealing and Grammatical Evolution, we achieve simulation times that are about 160 times lower.

$$Placement_{host,vm} = u_{host,vm} - \Delta f_{host,vm} \tag{9}$$
$$u_{host,vm} = u_{host} + u_{vm} \tag{10}$$
$$\Delta f_{host,vm} = f_{host,vm} - f_{host} \tag{11}$$

$u_{host,vm}$ is the estimated CPU utilization resulting from adding both host and vm utilizations (u_{host} and u_{vm}). $\Delta f_{host,vm}$ provides the estimated difference between the host frequency after ($f_{host,vm}$) and before (f_{host}) the VM allocation calculated for the new estimated utilization. Algorithm 2 presents our DVFS-Aware Dynamic Placement proposal.

The input *vmList* represents the VMs that have to be migrated according to the stages (i), (ii) and (iii) of the consolidation process, while *hostsList* is the entire set of servers in the data center that are not considered overutilized. First, VMs are sorted in a decreasing order of their CPU requirements. Steps 3 and 4 initialize *bestPlacement* and *bestHost*, which are the best placement value for each iteration and the best host to allocate the VM respectively. Then, each VM in *vmList* will be allocated in a server that belongs to the list of hosts that are not overutilized (*hostList*) and have enough resources to host it.

In steps 7 and 8, the algorithm calculates the value of the estimated CPU *utilization* ($u_{host,vm}$) and *freqIncrement* ($\Delta f_{host,vm}$) after vm allocation using Equations 10 and 11. According to our

Algorithm 2 Frequency-Aware Placement

```
Input: hostsList, vmList
Output: FreqAwarePlacement of VMs
1: vmList.sortDecreasingUtilization()
2: foreach vm in vmList do
3:   bestPlacement ← MIN
4:   bestHost ← NULL
5:   foreach host in hostList do
6:     if host has enough resources for vm then
7:       utilization ← estimateUtilization(host, vm)
8:       frequencyIncrement ← estimateFrequencyIncrement(host,vm)
9:       placement ← utilization - frequencyIncrement
10:      if placement > bestPlacement then
11:        bestHost ← host
12:        bestPlacement ← placement
13:  if bestHost ≠ NULL then
14:    FreqAwarePlacement.add(vm, bestHost)
15: return FreqAwarePlacement
```

allocation strategy, derived from the above considerations, the *placement* value ($Placement_{host,vm}$) obtained when a VM is allocated in a specific host is calculated in step 9 using Equation 9.

As can be seen in steps 10, 11 and 12, the VM is allocated in the host that has a higher *placement* value, that means a high CPU utilization but, on the contrary, it represents a low increase in frequency due to the utilization increment. This approach minimizes the number of bins used by this combinatorial NP-hard problem while taking full advantage of the range of CPU utilization available for each frequency. The output of this algorithm is the frequency-aware placement (*FreqAwarePlacement*) of the VMs that have to be mapped according to the under/overloaded detection and VM selection policies.

3.3. Frequency-Aware Optimization

Our Frequency-Aware Optimization combining the DVFS-perf policy with the Freq-Aware Placement algorithm is shown in listing of Algorithm 3. First, it finds the optimized placement of the VMs (*optimizedPlacement*) that have to be migrated due to dynamic workload variations. This is calculated in Algorithm 2, taking care of the frequency requirements. In step 2, the function *consolidateVM* allocates the VMs according to this mapping, performing VM migrations and updating utilization requirements for each host. Then in steps 3 and 4, the DVFS-perf configuration is obtained using Algorithm 1 with current utilization values. Finally the data center status is updated according to the optimized allocation and frequency configuration. Our DVFS-Aware strategy provides an elastic scale out that is adapted to the varying demand of resources. Also, the algorithm is light, making it suitable for quickly adaptation to workload fluctuations in the data center and run-time execution.

Algorithm 3 Frequency-Aware Optimization

```
Input: hostsList, vmList, frequenciesList
Output: optimizedConfiguration of the data center
1: optimizedPlacement ← frequencyAwarePlacement (hostList, vmList)
2: consolidateVM (optimizedPlacement)
3: optimizedFrequencies ← DVFS-perfConfiguration (hostList, frequenciesList)
4: setFrequencyConfiguration (optimizedFrequencies)
5: optimizedConfiguration ← configureDC (optimizedAllocation, optimizedFrequencies)
6: return optimizedConfiguration
```

4. SIMULATION CONFIGURATION

In this section, we present the impact of our frequency-aware policies in energy consumption because of the improved management of the workload and the frequency assignment in servers. However, large-scale experiments and their evaluations are difficult to replicate in a real data center infrastructure because it is difficult to maintain the same experimental system condition necessary for comparing different user and application scenarios. This can be achieved in simulation environment as simulators helps in setting up repeatable and controllable experiments.

For that reason, we have chosen the CloudSim toolkit [32] to simulate a Infrastructure as a Service (IaaS) Cloud computing environment. In contrast to other simulators, CloudSim provides the management of on-demand resource provisioning, representing accurately the models of virtualized data centers. The software version 2.0 that we have chosen supports the energy consumption accounting as well as the execution of service applications with workloads that vary along time [29]. For this work, we have provided frequency-awareness to the CloudSim simulator, also incorporating the ability to modify the frequency of servers. This frequency management policy allows to evaluate the performance of the algorithms proposed in Sections 3.1 and 3.2. Our code also supports switching the VM placement policy to compare our strategy with other approaches.

Simulations by our frequency-aware version of CloudSim have been executed in a 64-bit Windows 7 Operating System running on an Intel Core i5-2400 3.10GHz Dell Workstation with four cores and 4 GB of RAM. Moreover, the simulations are configured according to the following considerations:

4.1. Workload

We conduct our experiments using real data from PlanetLab, that comprises more than a thousand servers located at 645 sites around the world. The workload consists of 5 days of data with different resource demand profiles obtained from the CoMon monitoring project [33]. The data traces are available and fully operative in CloudSim as this workload is commonly used by researchers using this simulator. By using these traces we can compare our approach with published and future research works.

The main features of each of the 5 sets, as the number of VMs and both the mean and standard deviation values of the CPU utilization, are shown in Table I. Each of the five data sets includes CPU utilization values of around a thousand VMs with a monitoring interval of 300 seconds. We have chosen this collection because each independent workload can be executed for the same data center's initial size. Also, the usage of traces from a real system makes our simulation-based analysis applicable to real scenarios.

Table I. PlanetLab workload main features

Date	VMs	CPU mean utilization	CPU utilization SD
2011.03.03	1052	12.31 %	17.09 %
2011.03.06	898	11.44 %	16.83 %
2011.03.09	1061	10.70 %	15.57 %
2011.04.12	1054	11.54 %	15.15 %
2011.04.20	1033	10.43 %	15.21 %

4.2. Physical Nodes

The simulation consists of a set of 400 hosts conforming a data center. This is the minimum amount of resources required by the CloudSim initial provisioning policy to manage the number of VMs for the different workloads that we have selected. During simulations, the number of servers will be significantly reduced as oversubscription is enabled. Hosts are modeled as a Fujitsu RX300 S6 server based on an Intel Xeon E5620 Quad Core processor @2.4GHz, RAM memory of 16GB and storage of 1GB, running a 64bit CentOS 6.4 OS virtualized by the QEMU-KVM hypervisor.

DOI: 10.1002/cpe

4.2.1. DVFS Governors. The DVFS system of our Fujitsu server operates at 1.73, 1.86, 2.13, 2.26, 2.39 and 2.40 GHz respectively. For our experiments, we define two different governors to dynamically manage the CPU frequency. Both of them are fully available in our CloudSim modified version. For this work, we have provided frequency-awareness to the CloudSim simulator, also incorporating the ability to modify the frequency of servers according to our new DVFS-perf policy.

- **Performance**. The CPUfreq governor *performance*† is a typical governor available in the Linux Kernel. It sets the CPU to the highest frequency of the system.
- **DVFS-perf**. This governor dynamically modifies the CPU frequency according to Algorithm 1 so, it is set to the minimum frequency that ensures QoS depending on the workload.

4.2.2. Server Power Modeling. The power model used to estimate the energy consumed by these servers was proposed in our previous work [28] and can be seen in Equation 12. Then the energy consumption is obtained using Equation 13 where t determines the time in which the energy value is required. The operating frequencies set (in GHz) is provided in 14.

$$P_{\text{Fujitsu,k}} = 3.32 \cdot V_{\text{DD}}^2(k) \cdot f_{op}(k) \cdot u_{\text{cpu}} + 1.63 \cdot 10^{-3} \cdot T_{\text{mem}}^2 + 4.88 \cdot 10^{-11} \cdot FS^3 \quad (12)$$

$$E_{\text{Fujitsu}} = P_{\text{Fujitsu}} \cdot t \quad (13)$$

$$f_{op}(k) = \{1.73, 1.86, 2.13, 2.26, 2.39, 2.40\}(GHz) \quad (14)$$

$$V_{DD}(k) = \{1.29, 1.39, 1.59, 1.69, 1.79, 1.8\}(V) \quad (15)$$

This model presents a validation error of 4.46% when comparing power estimation to real measurements of the actual power. We used applications that can be commonly found in nowadays' Cloud data centers (including web search engines, and intensive applications) for training and validation stages. We assume a thermal management that allows memory temperature and fan speed to remain constant as we are interested in analyzing the power variations only due to utilization and DVFS management provided by our Freq-Aware optimization. The temperature of the memory T_{mem} and the fan speed FS are considered constant at 308 K and 5370 RPM respectively. Both parameters take their average values from the exhaustive experimental evaluation for this type of server that has been performed in our aforementioned previous work. This approach is valid since current models usually take into account only the variation of the dynamic consumption, as seen in Section 2. By including our power model in the CloudSim toolkit we are able to evaluate the power consumption in a more accurate way, as both the dynamic (depending on CPU utilization and frequency) and the static contributions are now considered. Thus, the impact of DVFS and consolidation-aware optimizations on the data center IT energy consumption is more likely to be measured by including our proposed models.

4.2.3. Active Server Set. In this work we assume that a server is switched off when it is idle, so no power is consumed when there is not any any running workload. Also, servers are turned on when needed, if the system is overloaded. We take into account the booting energy consumption required by a server to be fully operative as seen in Equation 18.

$$P_{boot} = 1.63 \cdot 10^{-3} \cdot 308^2 + 4.88 \cdot 10^{-11} \cdot 5370^3 = 162.1768W \quad (16)$$

$$t_{boot} = 300s \quad (17)$$

$$E_{boot} = P_{boot} \cdot t_{boot} = 13.514 \cdot 10^{-3} kW \cdot h \quad (18)$$

where P_{boot} is the server booting power working at 308 K and 5370 RPM as defined above and t_{boot} is the booting time obtained experimentally.

† www.kernel.org/doc/Documentation/cpu-freq/governors.txt

Prepared using cpeauth.cls
DOI: 10.1002/cpe

4.3. Virtual Machines

4.3.1. VM types. The simulation uses heterogeneous VM instances that correspond to existing types of the Amazon EC2 Cloud provider. The Extra Large Instance (2000 MIPS, 1.7 GB RAM), the Small Instance (1000 MIPS, 1.7 GB RAM) and the Micro Instance (500 MIPS, 613 MB RAM) are available for all the scenarios. All the VM are forced to be single-core to meet the PlanetLab data set requirements.

4.3.2. Migration policy. In all our scenarios we allow online migration, where VMs follow a straightforward load migration policy. During migration, another VM, which has the same configuration as the one that is going to be migrated, is created in the target server. Then the cloudlets are migrated from the source VM to the target VM. Finally, when the migration is finished the source VM is removed. Live migration has two different overheads that affect to energy consumption and performance degradation. Therefore, it is crucial to minimize the number migrations in order to optimize energy efficiency while maintaining QoS.

Energy overhead. A migration takes a time known as *migration time* ($t_{migration}$), that is defined in Equation 19. Migration delay depends on the network bandwidth (*BW*) and the *RAM* memory used by the VM. We consider that only half of the bandwidth is used for migration purposes, as the other half is for communication. Thus, migrations have an energy overhead because, during migration time, two identical VMs are running, consuming the same power in both servers.

$$t_{migration} = \frac{RAM}{BW/2} \tag{19}$$

Performance overhead. Performance degradation occurs when the workload demand in a host exceeds its resource capacity. In this work we model that oversubscription is enabled in all servers. So, if the VMs hosted in one physical machine simultaneously request their maximum CPU performance, the total CPU demand could exceed its available capacity. This situation may lead to performance degradation due to host overloading. The impact on SLA can be calculated as the SLA violation time per active host (SLA_{TAH}) that can be seen in Equation 20.

On the other hand, when overloading situations are detected, VMs are migrated to better placements, thus provoking performance degradation due to migration (PDM) as seen in Equation 21. The metric used in this work to determine the SLA violation ($SLA_{violation}$) [29] combines SLA_{TAH} and PDM as shown in Equation 23:

$$SLA_{TAH} = \frac{1}{M}\sum_{i=1}^{M} \frac{t_{100\%_i}}{t_{active_i}} \tag{20}$$

$$PDM = \frac{1}{V}\sum_{j=1}^{V} \frac{pdm_j}{C_{demand_j}} \tag{21}$$

$$pdm_j = 0.1 \cdot \int_{t_0}^{t_0+t_{migration}} u_j dt \tag{22}$$

$$SLA_{violation} = SLA_{TAH} \cdot PDM \tag{23}$$

where M is the number of servers; $t_{100\%_i}$ and t_{active_i} are the time in which the CPU utilization of the host i is 100% and the total time in which it is active respectively. V is the number of VMs and C_{demand_j} represents the CPU demand of the VM during its lifetime. pdm_j defines the performance degradation per VM during $t_{migration}$. In our experiments it is estimated as the 10% of the CPU utilization in MIPS during the migration time of VM j. Finally, t_0 is the time in which the migration starts and u_j is the CPU utilization of VM j.

4.4. Dynamic Consolidation Configuration

The present work aims to evaluate the performance of DVFS-aware dynamic consolidation. Consolidation phases (i), (ii) and (iii) are able to use the algorithms for the detection of overloaded or underloaded hosts and for the selection of VMs to be migrated that are available in CloudSim 2.0 [29]. We have simulated all the possible combinations for both types of algorithms with the default configuration of internal parameters, resulting in 15 different tests. The internal parameters for each option are set to those values that provide better performance according to Beloglazov et al [29]. Finally, consolidation phase (iv) is able to use two different power-aware placement algorithms.

4.4.1. Over/Underloading detection algorithms. We consider the detection of overloaded or underloaded hosts using five specific policies that belong to three different detection strategies.

- **Adaptive Utilization Threshold Methods**. Includes the *Interquartile Range* (IQR) and the *Median Absolute Deviation* (MAD) algorithms, and offers an adaptive threshold based on the workload utilization to detect overloaded or underloaded hosts. The internal safety parameters take the value 1.5 and 2.5 respectively, and define how aggressively the consolidation is considered in this stage.
- **Regression Methods**. Both the *Local Regression* (LR) and the *Local Regression Robust* (LRR) are Regression Methods based on the Loess method and have the same internal parameter of 1.2.
- **Static Threshold Method**. The *Static threshold* (THR) sets a fixed value to consider when a host is overloaded or underloaded. The internal parameter is 0.8.

4.4.2. VM Selection Algorithms. The selection of the VMs that have to be migrated from overloaded or underloaded hosts is performed by three different algorithms.

- **Maximum correlation** (MC). The system migrates the VM that presents a higher correlation of CPU utilization with other VMs so, the peak loads would occur at the same time.
- **Minimum migration time** (MMT). The algorithm selects the VM that takes less time to be migrated when compared to the rest of VMs hosted in the same server.
- **Random choice** (RS). The VM is randomly selected.

4.4.3. VM Placement Algorithms.

- **Power Aware Best Fit Decreasing** (PABFD). This placement policy for Cloud infrastructures takes into account the power consumption of the servers when finding an optimal placement under dynamic workload conditions [29]. It works well for SLA-constrained systems, maintaining QoS while reducing energy consumption. This solution does not take into account frequency increments due to workload allocation.
- **Frequency-Aware Placement** (Freq-Aware Placement). This is the DVFS-aware placement policy that we propose in Algorithm 2. This solution allows a dynamic consolidation that is aware of both power and frequency also taking into account QoS.

4.5. Scenarios

We provide three different scenarios to evaluate the performance of our frequency-aware optimization. For this purpose, we will compare our work with two different approaches. All the proposed scenarios are able to power on/off servers when needed as can be seen in section 4.2.3

- **Baseline** scenario represents the default performance of CloudSim. The performance governor is active so, the servers always operate at the maximum frequency. PABFD placement is used to perform VM allocation.
- **DVFS-only** scenario uses our DVFS-perf governor combined with PABFD placement. Thus, the frequency of each server is reduced to the lowest value that allows the system to meet QoS. However, the mapping is not aware of the allocation impact on CPU frequency that also impacts on the power consumption.
- **Freq-Aware Optimization** scenario combines our DVFS-perf governor with our Freq-Aware Placement as shown in Algorithm 3. Both utilization and frequency estimations are considered to find the optimized allocation. It aims to evaluate our proposed optimization strategy.

DOI: 10.1002/cpe

5. EXPERIMENTAL RESULTS

We have simulated the 3 different scenarios for each of the 5 different PlanetLab workloads presented in Table I, and tested the 15 different combinations of algorithms for overloading detection and VM selection aforementioned. Therefore, for each of the daily workloads we are able to present the following results per test (under/overload detection-VM selection) and per scenario, in order to compare our Freq-Aware optimization with the other two alternatives.

5.1. Performance Analysis

We consider the following metrics to analyze the obtained results. The number of VM migrations is considered as a figure of merit because migrations may cause SLA violations due to performance degradation, also impacting on energy consumption. Additionally we have included the overall SLA violations provided by the metric $SLA_{violation}$ to simultaneously verify if our policies meet QoS requirements. As CloudSim allows turning machines on when needed, we have included the additional booting energy consumption of the servers to the simulation. The number of *Power on events* is our proposed metric to evaluate its impact because, reducing the number of these events would decrease the overall data center energy. Service outages are experienced when the power density exceeds the maximum capacity of the grid. So, we evaluate the peak power during the simulation in order to analyze the system's performance under critic situations in terms of electricity supply. Finally, the energy signature is obtained in order to evaluate the efficiency introduced by our strategy.

Table II shows the average values of these metrics when comparing the baseline with the DVFS-only policy and with the Freq-Aware optimization. For each PlanetLab workload (represented as the date when it was obtained), the table shows the averaged values that result from their execution under every possible combination of the overloading detection and the VM selection algorithms. An average of 3.35% energy savings is achieved just including the DVFS capabilities to the simulation infrastructure for all the workloads. The savings in energy consumption come from the combined reduction of the VM migrations and the Power on events. In this scenario, QoS is maintained but the peak power is not improved when compared with the baseline.

The proposed Frequency-Aware Placement combined with the DVFS management significantly reduces both the number of power on events and VM migrations. The minimization of the times that a server is powered on has several benefits, not only reducing the energy consumption but also extending its lifetime. However, its impact on the total energy consumption represents only about 5.31%. So, the energy savings are obtained mainly due to the reduction of the VM migrations as, during each migration, an identical VM is simultaneously running in the source and in the target hosts. Our proposed Freq-Aware optimization policy outperforms the baseline obtaining average energy savings of 37.86% significantly reducing peak power consumption around 66.14% while maintaining the QoS, as can be seen in the peak power reduction column and in the SLA violations reduction column respectively.

Table II. Average values per day for baseline comparison

Optim. Policy	Date (yy.mm.dd)	VM migrations reduction	Power on events reduction	SLA violations reduction	Peak power reduction	Energy savings
DVFS-only	2011.03.03	4.40 %	13.63 %	0 %	-6.08 %	4.64 %
	2011.03.06	4.81 %	9.60 %	0.01 %	-2.87 %	3.45 %
	2011.03.09	3.63 %	5.16 %	0 %	-7.27 %	3.44 %
	2011.04.12	1.44 %	1.49 %	0 %	0.1 %	2.36 %
	2011.04.20	1.82 %	-3.72 %	0.01 %	5.81 %	2.59 %
Freq-Aware	2011.03.03	23.44 %	86.10 %	0 %	68.16 %	34.82 %
	2011.03.06	19.38 %	79.16 %	0.01 %	64.29 %	34.64 %
	2011.03.09	19.53 %	85.41 %	0 %	64.34 %	39.14 %
	2011.04.12	26.77 %	88.03 %	0.01 %	66.19 %	38.88 %
	2011.04.20	19.55 %	85.81 %	0 %	69.09 %	41.62 %

The different tests, each of them representing a specific combination of overloading detection and VM selection algorithms, perform differently. However, the performance pattern for each test is repeated for every considered PlanetLab workload in Table I. Thus, we are able to analyze the system's performance for every test, shown in Figure 1, which presents the averaged values of each metric for all the workloads. As shown in 1.e, both policies achieve energy savings for each test but the Freq-Aware optimization reduces the data center energy consumption to an average value of 69.16 kWh for all the workloads regardless the combination of algorithms. This means an average savings of 37.86%. In 1.d we obtain a similar pattern in the overall peak power of the IT infrastructure, achieving a reduction of about 66.14%.

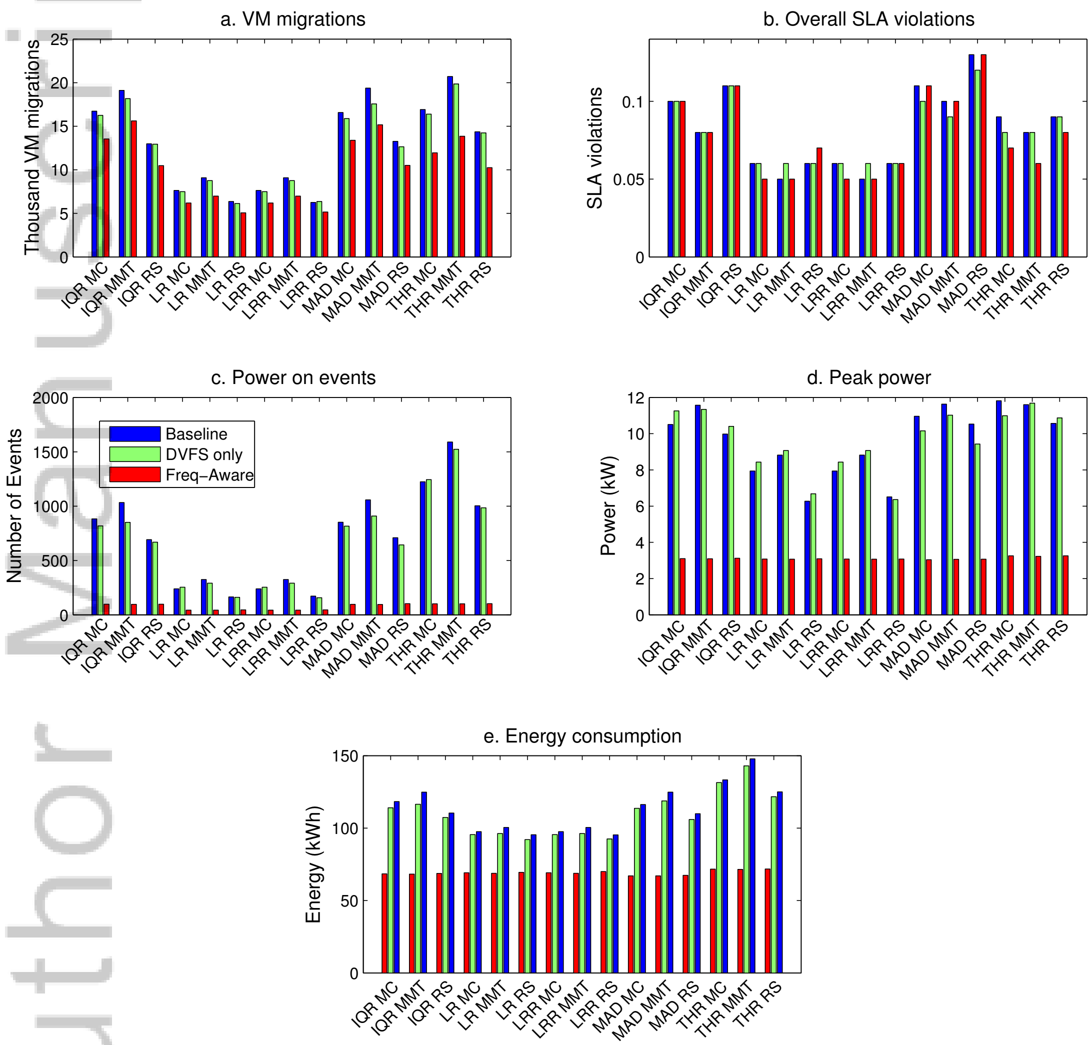

Figure 1. Average metrics per test

The same occurs in 1.c for the number of power on events that is reduced to about 76.53 events, showing average savings of 86.03%. However, not every test performs the same in terms of SLA violations. Overall SLA violation for local regression methods combined with MC and MMT algorithms present better values of about 0.05% as can be seen in 1.b. Also in 1.a, average VM migrations vary considerably from one test to another. So, the SLA violations and VM migrations metrics may be determining factors when selecting a combination of overloading detection and VM selection algorithms.

DOI: 10.1002/cpe

5.2. Run-time Evaluation

Moreover, to deeply understand the performance of the Freq-Aware optimization during run-time, we have selected one of the workloads for its simulated execution under the conditions of a specific test. Figure 2 presents the temporal evolution of the test that combines the MAD and MMT algorithms as it achieved the lowest total energy consumption. The test runs the 1052 VMs of the workload dated on 2011.03.03 because it achieves the highest CPU utilization and standard deviation (see Table I).

In this framework, we evaluate additional metrics to compare both baseline and Freq-Aware scenarios. Figure 2.a shows the global resource demand of this workload in terms of MIPS. The global utilization represents the average CPU utilization of all the servers in the data center. The number of active hosts within the total facility is also analyzed because, as this value increases, the global utilization will be reduced. Finally the cumulative energy consumption of the IT infrastructure is presented to study its deviation between both scenarios during a 24 hours-workload.

For the baseline policy, the number of active hosts is highly increased during peaks of workload demand, consequently reducing the data center global utilization, as can be seen in Figures 2.b and 2.c respectively. The decrease on the overall utilization also reduces each server energy consumption, as its power depends linearly on CPU demand. However, the static consumption (which accounts for about 70% of total consumption in each physical machine) due to the additional servers that are required to execute the workload with this utilization, highly increases the total energy budget. On the other hand, for the Freq-Aware optimization policy, both values remain more constant, as shown in Figures 2.b and 2.e respectively.

The DVFS configuration of the active server set during run-time can be seen in Figure 2.d. The DVFS mode operating at $2.13GHz$ is the most selected, as it offers a wider range of utilizations in which the frequency remains constant. This frequency allows a sufficiently high utilization (from 77.5% to 88.75%) that helps to minimize the number of servers. The rest of DVFS modes are also used but mainly to absorb load peaks as dynamic workload fluctuates during run-time.

Our algorithm speeds up both the consolidation into a lower number of active servers and the elastic scale out of the IT infrastructure, increasing the global utilization in a 23.46% while reducing the number of active hosts around a 44.91%. Table III presents the averaged values for these results. Figure 2.f shows how this behavior impacts on the energy usage of the data center where the baseline consumption grows at a higher rate during dynamic workload variations than for the optimized scenario, achieving total energy savings of 45.76%.

Table III. Average results for MAD-MMT test running workload 2011.03.03.

Scenario	Global Utilization	Active Hosts	Total Energy
Baseline	60 %	35.49	125.45 kWh
Freq-Aware	83 %	19.55	76.72 kWh

6. CONCLUSIONS AND FUTURE WORK

The contribution of Cloud data centers in the overall consumption of modern cities is growing dramatically, so minimizing their energy consumption is a critical challenge to reduce economical and environmental impact. Cloud workloads significantly vary over time, thus achieving an optimal allocation of resources while preserving performance is not trivial.

The work presented in this paper makes relevant contributions on the optimization of Cloud data centers from a proactive perspective. In this work we present the Freq-Aware optimization that combines a novel reactive DVFS policy with our proactive Frequency-aware Consolidation technique. We have achieved competitive energy savings of up to 41.62% at the data center level maintaining QoS, even improving slightly the SLA violations around 0.01%, for real workload traces in a realistic Cloud scenario. According to our results, our algorithm enhances the

DOI: 10.1002/cpe

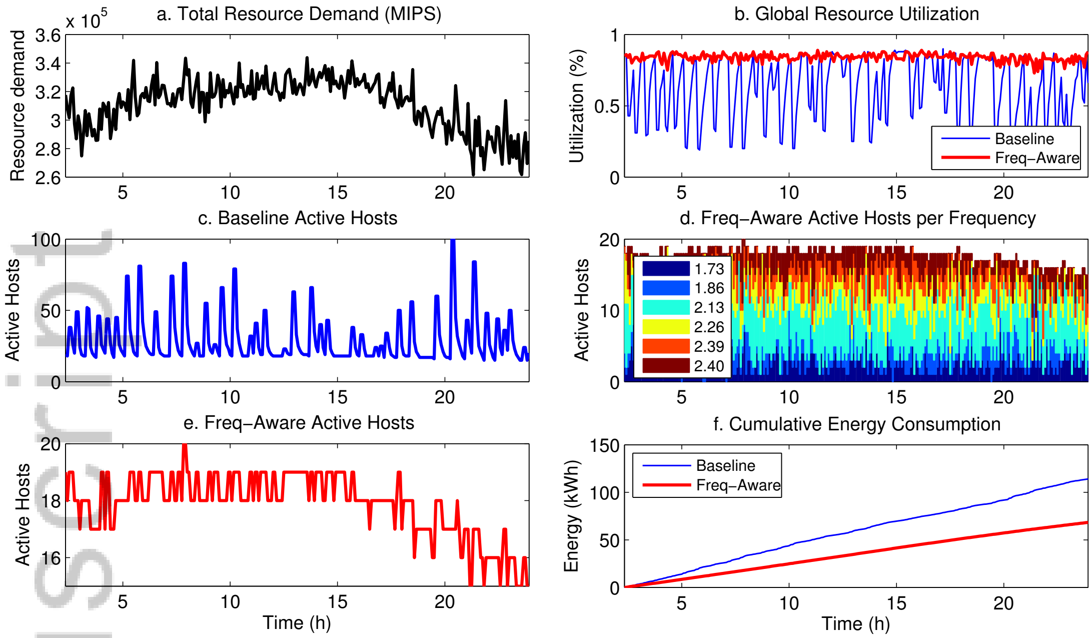

Figure 2. Temporal evolution for MAD-MMT test running workload 2011.03.03

consolidation process and speeds up the elastic scale out, reducing the global peak power demand about a 66.14% while improving the energy efficiency.

For future research, we plan to extend the techniques proposed in this paper (Freq-Aware optimization) for other application programming models supporting Web, HPC, Big Data, enterprise and transactions on mobile applications, as well as for different power models that also include memory performance. We also envision further energy optimization techniques, thus considering the combined effect of workload consolidation, DVFS and temperature. This research will help to optimize not only the computing resources of the data center but also the cooling contribution.

ACKNOWLEDGMENT

Visit of Patricia Arroba to CLOUDS lab, Melbourne University, has been supported by a collaboration fellowship of the European Network of Excellence on High Performance and Embedded Architecture and Compilation (HiPEAC). This project has been partially supported by the Spanish Ministry of Economy and Competitiveness, under contracts TEC2012-33892, IPT-2012-1041-430000 and RTC-2014-2717-3. This work is partially supported by a project grant from the Australian Research Council (ARC). We thank members of the CLOUDS Lab especially Rodrigo Calheiros and Amir Vahid for their comments and support.

REFERENCES

1. J. Varia, *Best Practices in Architecting Cloud Applications in the AWS Cloud*. John Wiley & Sons, Inc., Wiley Press, New York, USA, February 2011, pp. 457–490.
2. G. A. for Business, “Top ten advantages of Google’s cloud,” http://www.google.com/apps/intl/en/business/cloud.html, 2012, [Online; accessed 1-March-2012].
3. Microsoft News Center, “Microsoft on Cloud Computing,” http://www.microsoft.com/presspass/presskits/cloud/, 2012, [Online; accessed 1-March-2012].
4. iCloud, “What is iCloud?” http://www.apple.com/icloud/what-is.html, 2012, [Online; accessed 1-March-2012].
5. M. Raskino, J. Mahoney, and P. Meehan, “Gartner Research Report ID Number G00164278,” 5 January 2009, [CIO New Year’s Resolutions, 2009].
6. N. Engbers and E. Taen, “Green Data Net. Report to IT Room INFRA,” November 2014, european Commision. FP7 ICT 2013.6.2.

7. J. Koomey, "Growth in data center electricity use 2005 to 2010," Analytics Press, Oakland, CA, Tech. Rep., 2011.
8. A. Donoghue, P. Inglesant, and A. Lawrence, "The EU dreams of renewable-powered datacenters with smart-city addresses," "https://451research.com", October 2013.
9. Ponemon Institute, "2013 Study on Data Center Outages," Sponsored by Emerson Network Power, Tech. Rep., September 2013.
10. J. Hartley, "The truth about data centre cooling," *eCool Solutions*, 2010.
11. A. Berl, E. Gelenbe, M. Di Girolamo, G. Giuliani, H. De Meer, M. Q. Dang, and K. Pentikousis, "Energy-efficient cloud computing," *The Computer Journal*, vol. 53, no. 7, pp. 1045–1051, Sep. 2010.
12. P. Niles and P. Donovan, "Virtualization and Cloud Computing: Optimized Power, Cooling, and Management Maximizes Benefits. White paper 118. Revision 3," Schneider Electric, Tech. Rep., 2011.
13. A. Gandhi, M. Harchol-Balter, R. Das, and C. Lefurgy, "Optimal power allocation in server farms," *ACM SIGMETRICS Performance Evaluation Review*, vol. 37, no. 1, pp. 157–168, Jun. 2009.
14. M. M. Rafique, N. Ravi, S. Cadambi, A. R. Butt, and S. Chakradhar, "Power management for heterogeneous clusters: An experimental study," in *Proceedings of the 2011 International Green Computing Conference and Workshops*, ser. IGCC '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 1–8.
15. E. Le Sueur and G. Heiser, "Dynamic voltage and frequency scaling: the laws of diminishing returns," in *Proceedings of the 2010 international conference on Power aware computing and systems*, ser. HotPower'10. Berkeley, CA, USA: USENIX Association, 2010, pp. 1–8.
16. C.-M. Wu, R.-S. Chang, and H.-Y. Chan, "A green energy-efficient scheduling algorithm using the dvfs technique for cloud datacenters," *Future Generation Computer Systems*, vol. 37, no. 0, pp. 141 – 147, 2014.
17. L. Wang, G. von Laszewski, J. Dayal, and F. Wang, "Towards energy aware scheduling for precedence constrained parallel tasks in a cluster with dvfs," in *Proceedings of the 2010 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing*, ser. CCGRID '10, May 2010, pp. 368–377.
18. P. Huang, P. Kumar, G. Giannopoulou, and L. Thiele, "Energy efficient dvfs scheduling for mixed-criticality systems," in *Proceedings of the 14th International Conference on Embedded Software*, ser. EMSOFT '14. NY, USA: ACM, 2014, pp. 11:1–11:10.
19. N. Rizvandi, J. Taheri, A. Zomaya, and Y. C. Lee, "Linear combinations of dvfs-enabled processor frequencies to modify the energy-aware scheduling algorithms," in *Proceedings of the 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid2010)*, May 2010, pp. 388–397.
20. R. H. Katz, "Tech Titans Building Boom," http://spectrum.ieee.org/green-tech/buildings/tech-titans-building-boom/0, February 2009, [Online; accessed 9-March-2012].
21. R. Buyya, A. Beloglazov, and J. Abawajy, "Energy-efficient management of data center resources for cloud computing: A vision, architectural elements, and open challenges," in *Proceedings of the 2010 International Conference on Parallel and Distributed Processing Techniques and Applications, Las Vegas, USA*, ser. PDPTA 2010, CSREA Press, July 12-15 2010.
22. M. Ferdaus, M. Murshed, R. Calheiros, and R. Buyya, "Virtual machine consolidation in cloud data centers using aco metaheuristic," in *Proceedings of the 20th International European Conference on Parallel Processing (Euro-Par'14)*, ser. Lecture Notes in Computer Science, F. Silva, I. Dutra, and V. Santos Costa, Eds. Springer, 2014, vol. 8632, pp. 306–317.
23. F. Hermenier, X. Lorca, J.-M. Menaud, G. Muller, and J. Lawall, "Entropy: A consolidation manager for clusters," in *Proceedings of the 2009 ACM SIGPLAN/SIGOPS VEE'09*. NY, USA: ACM, 2009, pp. 41–50.
24. H. Liu, C.-Z. Xu, H. Jin, J. Gong, and X. Liao, "Performance and energy modeling for live migration of virtual machines," in *Proceedings of the 20th International Symposium on High Performance Distributed Computing*, ser. HPDC '11. New York, NY, USA: ACM, 2011, pp. 171–182.
25. V. D. Maio, R. Prodan, S. Benedict, and G. Kecskemeti, "Modelling energy consumption of network transfers and virtual machine migration," *Future Generation Computer Systems*, vol. 56, pp. 388 – 406, 2016.
26. Y. Wang and X. Wang, "Performance-controlled server consolidation for virtualized data centers with multi-tier applications," *Sustainable Computing: Informatics and Systems*, vol. 4, no. 1, pp. 52 – 65, 2014.
27. V. Petrucci, O. Loques, and D. Mossé, "A dynamic optimization model for power and performance management of virtualized clusters," in *Proceedings of the 1st International Conference e-Energy '10*. New York, NY, USA: ACM, 2010, pp. 225–233.
28. P. Arroba, J. L. Risco-Martín, M. Zapater, J. M. Moya, J. L. Ayala, and K. Olcoz, "Server power modeling for run-time energy optimization of cloud computing facilities," *Energy Procedia*, vol. 62, no. 0, pp. 401 – 410, 2014, SEB-14.
29. A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency and Computation: Practice & Experience*, vol. 24, no. 13, pp. 1397–1420, Sep. 2012.
30. Y. Minyi, "A simple proof of the inequality FFD(L) $<$ 11/9 OPT (L) + 1, for all L for the FFD bin-packing algorithm ," *Acta Mathematicae Applicatae Sinica (English Series)*, vol. 7, no. 4, pp. 321 – 331, 1991.
31. S. Frey, *Conformance Checking and Simulation-based Evolutionary Optimization for Deployment and Reconfiguration of Software in the Cloud*. Books on Demand GmbH University of Kiel, 2014.
32. R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya, "Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and Experience*, vol. 41, no. 1, pp. 23–50, Jan. 2011.
33. K. Park and V. S. Pai, "Comon: A mostly-scalable monitoring system for planetlab," *ACM SIGOPS Operating Systems Review*, vol. 40, no. 1, pp. 65–74, Jan. 2006.