Expert Systems With Applications 110 (2018) 170–181

Contents lists available at ScienceDirect

Expert Systems With Applications

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

Energy policies for data-center monolithic schedulers

Damián Fernández-Cerero*, Alejandro Fernández-Montes, Juan A. Ortega

Department of Computer Languages and Systems, University of Seville, Av. Reina Mercedes S/N, Seville, Spain

ARTICLE INFO

Article history:
Received 5 December 2017
Revised 15 May 2018
Accepted 3 June 2018
Available online 4 June 2018

Keywords:
Energy policies
Efficiency
Data center
Simulation software
Decision making

ABSTRACT

Cloud computing and data centers that support this paradigm are rapidly evolving in order to satisfy new demands. These ever-growing needs represent an energy-related challenge to achieve sustainability and cost reduction. In this paper, we define an expert and intelligent system that applies various energy policies. These policies are employed to maximize the energy-efficiency of data-center resources by simulating a realistic environment and heterogeneous workload in a trustworthy tool. An environmental and economic impact of around 20% of energy consumption can be saved in high-utilization scenarios without exerting any noticeable impact on data-center performance if an adequate policy is applied.

1. Introduction

Cloud computing and large-scale web services have transformed the data-center scenario and the big-data environment, and have led to a new scenario where these infrastructures are as energy greedy as many factories. The latest estimations claim that data centers account for approximately 1.5% of global energy consumption (Koomey, 2011).

In this new scenario, data centers are in constant evolution towards servicing multiple heterogeneous workloads on the same hardware resources. This strategy enables higher energy-efficiency levels to be achieved by turning off idle resources in low-utilization periods. Decision-support systems are one of the main applications for expert systems. This work presents an automated decision-support system aimed to make the best decisions to improve the energy efficiency of the system through a better management of data-center resources and jobs placement. We develop, apply, and analyze various energy policies based on shutting machines off in order to reduce data-center energy consumption while preserving the cluster performance.

This approach has yet to be widely applied due to various reasons, such as: (a) Natural human behaviour and the fear of any change that could break operational requirements (Fernández-Montes, Fernández-Cerero, González-Abril, Álvarez-García, & Ortega, 2015); (b) the complexity and heterogeneity of all the subsystems involved; and (c) power-off policies, and (d) the fast development of new systems and paradigms that could break the established standards and systems. However, keeping servers underutilized or in idle state is highly inefficient from an energy-efficiency perspective.

On the other hand, the research community has made many efforts in other areas in order to achieve energy proportionality (Jakóbik, Grzonka, Kolodziej, Chis, & González-Vélez, 2017), such as: data-center operating temperature and cooling systems (El-Sayed, Stefanovici, Amvrosiadis, Hwang, & Schroeder, 2012; Sharma, Bash, Patel, Friedrich, & Chase, 2005), hardware energy proportionality (Fan, Weber, & Barroso, 2007; Miyoshi, Lefurgy, Van Hensbergen, Rajamony, & Rajkumar, 2002), upgrading hardware pieces such as HDDs to operate with non-mechanical devices such as SSDs (Andersen & Swanson, 2010), and improving power distribution infrastructures (Femal & Freeh, 2005) that have been put into production in various data centers from top-tier companies such as Google, Microsoft, and Amazon.

The paper is organized as follows. The related work is described in Section 2 and various powering-off resources strategies are shown in Section 3. Section 4 presents the simulation tool adapted and used for the experimentation environment shown in Section 5.

Finally, results are shown and analyzed in Section 6, where we compare energy-saving outcomes and the performance impact for each energy-efficiency policy. Conclusions are drawn in Section 7.

2. Related work

Many efforts have been made in order to increase resource and energy efficiency in data centers. The proposed strategies range from energy-aware scheduling algorithms to power-off heuristics

* Corresponding author.
E-mail addresses: damiancerero@us.es (D. Fernández-Cerero), afdez@us.es (A. Fernández-Montes), jortega@us.es (J.A. Ortega).

https://doi.org/10.1016/j.eswa.2018.06.007
0957-4174/

Table 1
Related work summary.

Ref.		Title	Savings
Duy et al. (2010)		Title: Performance evaluation of a green scheduling algorithm for energy savings in cloud computing	~45%
	Strategy:	Power off policy based on a neural network predictor	
	Evaluation:	[8–512] nodes cluster simulation	
	Workload:	End user homogeneous requests that follow a day/night pattern	
Lee and Zomaya (2012)		Title: Energy efficient utilization of resources in cloud computing systems	[5–30]%
	Strategy:	Energy-aware task consolidation heuristic based on different cost functions	
	Evaluation:	Simulation of a not stated size cluster	
	Workload:	Synthetic workload in terms of number of tasks, inter arrival time and resource usage	
Juarez et al. (2018)		Title: Dynamic energy-aware scheduling for parallel task-based application in cloud computing	[20–30]%
	Strategy:	Polynomial-time and multi-objective scheduling algorithm for DAG jobs	
	Evaluation:	Experimentation on a 64 nodes cluster	
	Workload:	Synthetic directed acyclic graph-based workload	
Beloglazov and Buyya (2010)		Title: Energy efficient resource management in virtualized cloud data centers	~80%
	Strategy:	VM allocation and migration policies + Always off policy	
	Evaluation:	100 nodes cluster simulation using CloudSim	
	Workload:	Synthetic workload that simulates services that fulfill the capacity of the cluster	
Ricciardi et al. (2011)		Title: Saving energy in data center infrastructures	[20–70]%
	Strategy:	Safety margin power-off policy	
	Evaluation:	100 and 5000 nodes cluster simulation	
	Workload:	Synthetic workload that follows a day/night pattern	

Table 2
Summary of the pros and cons of the energy-aware scheduling algorithms in the related work.

Duy et al. (2010) Performance evaluation of a green scheduling algorithm for energy savings in cloud computing	
Pros	Deeply described neural-network-based algorithm; Empirically measured power consumption
Cons	No focus on overall performance, only in drop rate; Small data-center size ([8–512] nodes)
	Short simulation period (2 days); No evaluation of huge & heterogeneous workload (cloud computing)
Fernández-Cerero et al. (2018) Security supportive energy aware scheduling and scaling for cloud environments	
Pros	Load balancing and VM scaling techniques; Computes security constraints
	Proposal of an energy-aware Genetic Algorithm
Cons	Focused on DVFS, not on shutting-down machines; Only for Independent Batch Scheduling environment
	No evaluation of huge & heterogeneous workload (real-life cloud computing system); Tiny cluster (5 VMs)
Juarez et al. (2018) Dynamic energy-aware scheduling for parallel task-based application in cloud computing	
Pros	DAG and data-aware workload; Multi-heuristic scheduling algorithm
Cons	Small data-center size (64 nodes max.); Only evaluates the makespan and total energy consumed
	No evaluation of huge & heterogeneous workload (real-life cloud computing system)
	Not focused on shutting-down machines, but in various DAG workloads
	Not clear about the cluster utilization (and the theoretical maximal energy efficiency)
Lee and Zomaya (2012) Energy efficient utilization of resources in cloud computing systems	
Pros	Large and detailed experimentation; Allows task migration
Cons	Focused on task scheduling, not on the shut-down of machines.
	No evaluation of huge & heterogeneous workload (real-life cloud computing system)
	No evaluation of the performance impact of the proposed strategies

that aim to minimize the number of idle nodes. A summary of these efforts is presented in Table 1, and a summary of the pros and cons of the related work regarding energy-aware scheduling algorithms, VM scaling and migration, and proposals based on shutting-down idle nodes is presented in Tables 2–4, respectively.

A substantial part of these approaches has been directed towards energy-aware scheduling strategies that could lead to powering off idle nodes, such as Duy, Sato, and Inoguchi (2010), Fernández-Cerero, Jakóbik, Grzonka, Kołodziej, Fernández-Montes (2018), Juarez, Ejarque, and Badia (2018), and Lee and Zomaya (2012). In Duy et al. (2010), a Green Scheduling Algorithm based on neural networks is proposed. This algorithm predicts workload demand in order to apply only one power-off policy to idle servers. These experiments simulate a small data center (512 nodes as a maximum) which serves an homogeneous workload composed of end-user facing tasks which follow a day/night pattern. Lee and Zomaya (2012) present two energy-aware task consolidation heuristics. These strategies aim to maximize resource utilization in order to minimize the wasted energy used by idle resources. To this end, these algorithms compute the total cpu time consumed by the tasks and prevent a task being executed alone. Juarez et al. (2018) propose an algorithm that minimizes a multi-objective function which takes into account the energy-consumption and execution time by combining a set of heuristic rules and a resource allocation technique. This algorithm is evaluated by simulating DAG-based workloads, and energy-savings in the range of [20–30%] are shown. Fernández-Cerero et al. (2018) propose energy-aware scheduling policies and methods based on Dynamic Voltage and Frequency Scaling (DVFS) for scaling the virtual resources while performing security-aware scheduling decisions.

In addition, different techniques of energy conservation such as VM consolidation and migration (Beloglazov, Abawajy, & Buyya, 2012; Beloglazov & Buyya, 2010, 2012; Sohrabi, Tang, Moser, & Aleti, 2016) are also proposed. Beloglazov and Buyya (2010) describe a resource management system for virtualized cloud data centers that aims to lower the energy consumption by applying a set of VM allocation and migration policies in terms of current CPU usage. This work is extended by focusing on SLAs restrictions in Beloglazov et al. (2012) and by developing and compar-

Table 3
Summary of the pros and cons of the VM scaling and migration algorithms in the related work.

Beloglazov and Buyya (2010) Energy efficient resource management in virtualized cloud data centers &	
Beloglazov et al. (2012) Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing	
Pros	VM resizing and migration; Thermal and network considerations.
Cons	Not focused on shutting down machines, but on VM placement; Small data-center size (100 nodes)
	No evaluation of huge & heterogeneous workload (real-life cloud computing system)
	No evaluation of the performance impact of the proposed strategies (only SLA violations)
Sohrabi et al. (2016) Adaptive virtual machine migration mechanism for energy efficiency	
Pros	Machine learning for re-scheduling tasks when hosts become overloaded; Real-life workload
Cons	Not focused on shutting down machines, but in VM placement;Not large data-center size (800 machines)
	No detailed evaluation of the performance impact (only SLA violations & makespan)
Beloglazov and Buyya (2012) Optimal online deterministic algorithms and adaptive heuristics	
for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers	
Pros	Dynamic VM resizing and migration; Dynamic host overloading algorithms; Real-life workload; Extensive experimentation
Cons	Not focused on shutting down machines, but in VM placement; Not large data-center size (800 machines)
	No detailed evaluation of the performance impact (only SLA violations)

Table 4
Summary of the pros and cons of the proposals based on shutting-down idle nodes in the related work.

Ricciardi et al. (2011) Saving energy in data center infrastructures	
Pros	Day-night workload pattern; Two different-sized data centers (5000 and 100 nodes)
Cons	Only one energy-efficiency policy based on a security margin
	No performance impact evaluation; No description of workload and simulation tool
Amur et al. (2010) Robust and flexible power-proportional storage	
Pros	Near optimal power proportionality; Various data-layout policies
	Almost no negative impact in data loss; Good experimental analysis based on standard benchmarks.
Cons	Focused only on cluster storage; Small data center (25 nodes); Read-only workload
Kaushik and Bhandarkar (2010) Greenhdfs: towards an energy-conserving, storage-efficient, hybrid hadoop compute cluster	
Pros	Cold and hot data areas; Real-life HDFS traces workload; Large Yahoo! data center (2600 nodes)
Cons	Focused only on cluster storage; Few details on the simulation tool and performance impact
Luo et al. (2013) Superset: a non-uniform replica placement strategy towards high-performance and cost-effective	
distributed storage service	
Pros	The dynamic replication may improve both energy efficiency and performance
	Extensive experimentation with a comparative with Thereska et al. (2011)
Cons	Focused only on cluster storage; Few details of the simulation tool; Small data center (240 nodes)
Thereska et al. (2011) Sierra: practical power-proportionality for data center storage	
Pros	Real-life workload presenting a day/night pattern; No extra capacity nor migration required
	Read & write workload; Network-aware; Extensive experimentation
Cons	Focused only on cluster storage; Small data center (31 nodes)

ing various adaptive heuristics for dynamic consolidation of VMs in terms of resource usage in Beloglazov and Buyya (2012). These migration policies are evaluated by simulating a 100-node cluster. Energy reductions up to approximately 80% are shown with low impact on quality of service and SLAs. In Sohrabi et al. (2016), a Bayesian Belief Network-based algorithm that aims to allocate and migrate VMs is presented. This algorithm uses the data gathered during the execution of the tasks in addition to the information provided at submission time in order to decide which of the virtual machines are to be migrated when a node is overloaded. In Ricciardi et al. (2011), a different approach is proposed. In this work, Ricciardi et al. present a data center energy manager that relies on day/night workload patterns in order to aggregate traffic during night periods and therefore turn off idle nodes. The authors apply a power-off policy based on a safety margin in order to minimize the negative impact on performance. To evaluate this strategy, two different data centers of 5000 and 100 nodes are simulated. In this kind of scenario, potential energy reductions between approximately 20 and 70% are shown.

The application of these techniques together presents a major opportunity in various large-scale scenarios, such as Grid 5000 (De Assuncao, Gelas, Lefevre, & Orgerie, 2012).

In order to achieve energy proportionality, many efforts (Amur et al., 2010; Kaushik & Bhandarkar, 2010; Luo, Wang, Zhang, & Wang, 2013; Thereska, Donnelly, & Narayanan, 2011) have been made in only one subset of all the systems, since these represented the main bottleneck when they were written. In Amur et al. (2010), a power-proportional distributed file system that stores replicas of data on non-overlapping subsets of nodes is proposed. These subsets of different sizes contain one replica for each file. This partitioning strategy lets the administrator decide the number of datasets to be kept turned on to serve incoming requests, and therefore it gives the administrator the opportunity to control the trade-off between energy consumption and performance. Kaushik and Bhandarkar (2010) present a variant of Hadoop Distributed File System that divides the cluster in two zones in terms of data usage pattern. The first zone, called the *Hot Zone*, contains the subset of fresh data that is more likely to be accessed short term. The second zone, called the *Cold Zone*, contains the set of files with low spatial or temporal popularity with few to rare accesses. Once the cluster is divided in these two zones, an aggressive power-off policy is applied to the *Cold Zone*. This energy-efficiency strategy achieves approximately 26% energy reduction without notably worsening the overall performance and reliability in a three-month simulation based on a Yahoo! cluster configuration. In Thereska et al. (2011), the cluster is partitioned in order to create different non-overlapping data zones. Each of these zones contains one replica of the cluster data. Once the cluster is partitioned, the system lets the administrator power off the desired number of zones, depending on the aggressiveness of the energy-efficiency strategy. Luo et al. (2013) propose a non-uniform replica placement strategy in terms of data popularity. This strategy aims to increase the number of available parallel replicas for data that is very likely to be accessed, and to lower the number of replicas of the low-used data that is rarely accessed in order to power off the maximum number of nodes without affecting the overall performance. In order to evaluate this strategy, a Zipf distribution-based workload and a real trace of Youku is executed in a 240-nodes simulated cluster.

This paper follows a different approach: to deeply describe the impact of 6 different power-off policies in terms of performance and energy consumption on a well-defined, rich and realistic heterogeneous workload that follows the trends present in Google Traces by running a huge amount of experiments for centralized monolithic scheduling frameworks. In order to better characterize the impact of these power-off policies and unlike the presented related work, this paper does not focus on developing energy-aware VM allocation or migration policies, but the authors use a Best-fit-like VM allocation heuristic and does not apply VM migration strategies as stated in Section 5. In addition, these power-off policies are applied at the data center operating system / resource manager level, not to a framework or subsystem like some of the related work presented. This difference makes it possible to apply the proposed power-off policies to any framework that can run as a VM / Linux container on the data center.

3. Power-off policies

In this work, we have developed several deterministic and probabilistic power-off decision policies. These power-off decision policies form the core of the work since they have much more impact on data-center efficiency and performance than anything else.

From among the *deterministic* policies, the following policies have been developed:

- *Never power off*: This power-off decision policy disables the power-off process, and therefore represents the current scenario.
- *Always power off*: This power-off decision policy will shut down every machine after freeing all the resources under use, whenever possible.
- *Maximum load*: This power-off decision policy takes into account the maximum resource pressure of the data-center load and compares it to a given threshold μ. If the current load is less than this given threshold μ, then the machine will be powered off.
- *Minimum free-capacity margin*: This power-off decision policy assures that at least a given percentage of resources μ is turned on, free, and available in order to respond to peak loads.

Regarding among the *probabilistic* policies, the following policies have been implemented:

- *Random*: This policy switches off and randomly leaves the resources idle by following a Bernoulli distribution whose parameter is equal to 0.5. This policy is useful to ascertain the accuracy of the predictions made by the following probabilistic policies.
- *Exponential*: The Exponential distribution, denoted by $Exp(\lambda)$, describes the time between events in a Poisson process, that is, a process in which events occur continuously and independently at a constant average rate $(1/\lambda)$. Under the hypothesis that the arrival of new jobs follows an Exponential distribution, this energy policy attempts to predict the arrival of new jobs that can harm the data-center performance due to the lack of sufficient resources for their execution.
  To compute the λ parameter, the most recent jobs are taken into account. The size of these last jobs is denoted as *Window size*. Thus, every time a shut-down process is executed, the mean time between these last jobs that could not be served at the time of making the decision is computed, and denoted by δ. Hence, $\lambda = 1/\delta$ by using the method of maximum likelihood. The probability of the arrival of a new job can then be computed by means of the exponential cumulative density function (cdf), as $cdf(T_s)$[1] $= 1 - e^{-T_s/\delta}$. Therefore, given a decision threshold μ value, the following conditions are imposed:
  $$\begin{cases} \text{if } cdf(T_s) >= \mu & \text{then leave resources } Idle \\ \text{if } cdf(T_s) < \mu & \text{then switch resources } Off \end{cases}$$
- *Gamma*: The Gamma distribution, denoted by $\Gamma(\alpha, \beta)$, is frequently used as a probability model for waiting times and presents a more general model than the Exponential distribution. Under the hypothesis that the arrival of new jobs follows a Gamma distribution, this energy policy attempts to predict the arrival of the amount of new jobs required to oversubscribe the available resources.
  and takes into account the *Lost factor* described in the *Exponential* policy. are:
  – $mem_{available}$: memory in *Idle* state.
  – $cpu_{available}$: computational resources in *Idle* state.
  – mem_{mean}: mean RAM used by last jobs.
  – cpu_{mean}: mean computational resources used by last jobs.
  – δ: mean inter-arrival time of last jobs.
  – α_{cpu}: as $cpu_{available}/cpu_{mean}$.
  – α_{mem}: as $mem_{available}/mem_{mean}$.
  The parameters of the Gamma distribution are then estimated as: $\alpha = Min(\alpha_{cpu}, \alpha_{mem})$ and $\beta = \delta$. Finally the probability of the arrival of new jobs is computed by means of the cumulative density function (cdf) with:
  $$cdf(T_s) = \frac{\gamma(\alpha, \beta x)}{\Gamma(\alpha)}$$
  Hence, given a decision threshold μ value, the following conditions are imposed:
  $$\begin{cases} \text{if } cdf(T_s) >= \mu, & \text{then leave resources } Idle \\ \text{if } cdf(T_s) < \mu, & \text{then switch resources } Off \end{cases}$$

4. Simulation tool

In this paper, we extended the Google lightweight simulator presented in Schwarzkopf, Konwinski, Abd-El-Malek, and Wilkes (2013) in order to perform energy-efficiency analysis. This simulator lets the authors focus on the development of energy-efficiency policies and perform simulations of the different scheduling frameworks and various data-center environments, while abstracting the details of each of them. The following energy states are considered : (a) *On*: 150 W (b) *Off*: 10 W (c) *Idle*: 70 W (d) *Shutting Down*: 160 W (e) *Powering On*: 160 W. The energy consumption is linearly computed in terms of the usage of each core.

Moreover, this tool provides us with a trustworthy implementation of the monolithic scheduling processes, and results have been contrasted to Google's realistic simulator (Schwarzkopf et al., 2013). The simulator employed can be found at https://github.com/DamianUS/cluster-scheduler-simulator.

5. Experimentation

In order to test and measure the achieved power savings and the consequent impact on data-center performance, a set of experiments have been run. Each of these experiments simulates a period of seven days of operation, and applies various combinations of the energy policies developed and described in Section 5.2. These experiments are designed to simulate realistic and heterogeneous environments.

[1] T_S is defined *as the minimum time that ensures energy saving if a resource is switched off between two jobs*(Orgerie, Lefèvre, & Gelas, 2008).

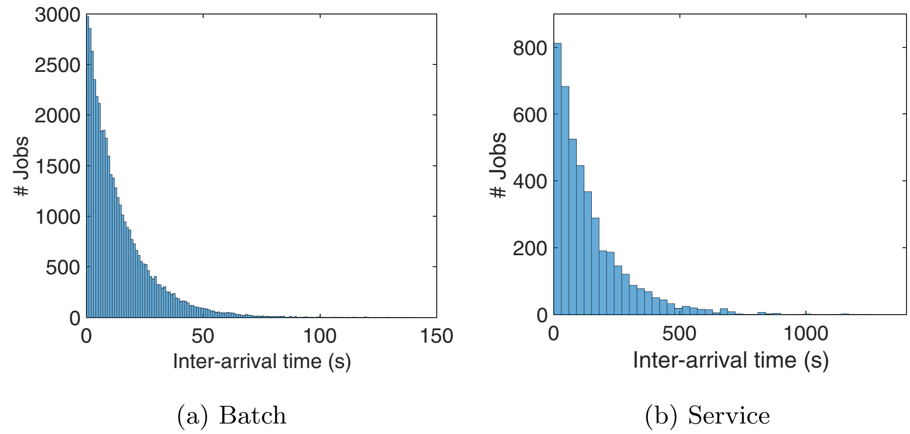


Fig. 1. Workload inter-arrival histogram.

In order to create a realistic and trustworthy testbed, realistic Google traces (Reiss, Wilkes, & Hellerstein, 2011, 2012b) were chosen and the interpretations carried out over these traces by the research community (Abdul-Rahman & Aida, 2014; Di, Kondo, & Franck, 2013; Liu & Cho, 2012; Reiss, Tumanov, Ganger, Katz, & Kozuch, 2012a) were studied.

In the following subsections, the test suite and environment designed and used are presented.

5.1. Workload

Jobs are composed of one or more tasks: sometimes thousands of tasks. In this work, two types of jobs are considered:

- *Batch* jobs: This workload is composed of jobs which perform a computation and then finish. These jobs have a determined start and end. MapReduce jobs are an example of a *Batch* job.
- *Service* jobs: This workload is composed of long-running jobs which provide end-user operations and infrastructure services. As opposed to *Batch* jobs, these jobs have no determined end. Web servers or services, such as BigTable (Chang et al., 2008), are good examples of a *Service* job.

Synthetic workloads are generated in each experiment run by replicating the behaviour of those workloads present in typical Google data centers. Therefore, although the workload generated in each simulation run is unique, they follow the same model design.

The subsequent job attributes have been covered and studied:

- *Inter-arrival time*: The inter-arrival time represents the time between two consecutive *Service* jobs or two consecutive *Batch* jobs. It also determines the amount of jobs executed in a specific time window. The inter-arrival time between two *Batch* jobs is usually shorter than that between two *Service* jobs, as illustrated in Fig. 1, leading to a higher number of *Batch* jobs, as illustrated in Fig. 4.
- *Number of tasks*: This parameter represents the number of tasks that comprise a job. As illustrated in Fig. 2, *Batch* jobs are composed of a higher number of tasks than *Service* jobs.
- *Job duration*: This parameter represents the time that a job consumes resources in the data center. As illustrated in Fig. 3, *Batch* jobs require less time to complete than *Service* jobs.
- *Resource usage*: Taking into account the parameters described above, although *Batch* jobs and tasks constitute the vast majority, the higher resource utilization and duration of *Service* jobs results in our synthetic workload as illustrated in Fig. 4. In this figure, it can be noticed that less than 10% of jobs in the workload are *Service* jobs, while less than 3% of tasks are *Service* tasks. It should be borne in mind, however, that almost 40% of CPU and 50% of RAM resources are used by *Service* jobs.

Taking into account the aforementioned environment and workload scenario, the generated workload is composed of 43,050 *Batch* jobs, 4238 *Service* jobs. This represents one week of operation time, and reaches 57, 81% computational power and 48.33% memory in use on average.

5.2. Experiments performed

After simulating a wide range of values for every parameter described in Section 3, for comprehension purposes, the most interesting and representative have been chosen:

In order to prevent resource contention, a power-on policy which turns on the necessary machines whenever the workload resource demands are higher than available machines, and a scheduling strategy which tries to fill every machine to the maximum (90%) while maintaining some randomness (Khaneja, 2015) is used. It is worth mentioning that in the experiments that simulate the *Never power off* policy, a scheduling strategy where resources are chosen randomly is used to represent the base scenario.

6. Results

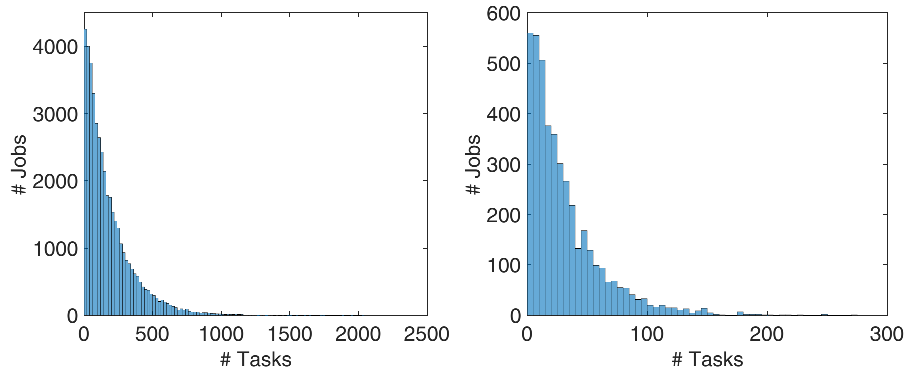
In this section, the obtained results are illustrated through key performance indicators concerning a) energy savings and b) impact over performance. In this way, energy savings and performance are analyzed and compared for each energy policy.

6.1. Energy savings indicators

The following indicators were selected in order to describe the energy savings and the behaviour of the powering on/off operations:

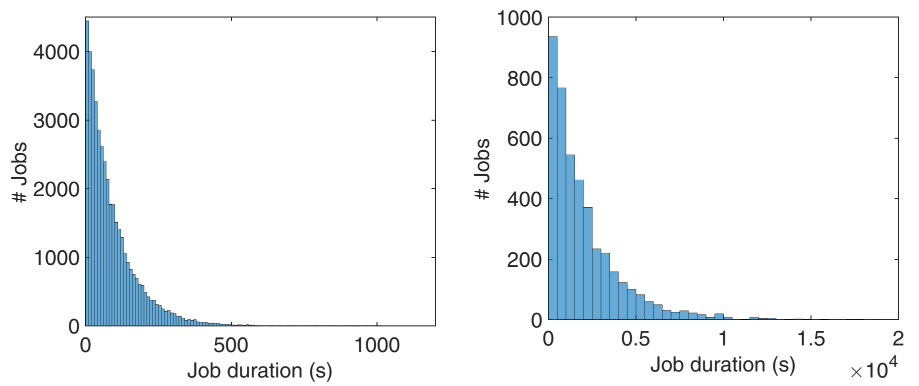
- *Energy consumed vs. current system*: The overall energy used in each experiment against the current[2] operation energy utilization.
- *Power-off operations*: The total number of shut-downs performed over all the resources during the overall simulated operation time.
- *KWh saved per shut-down*: This represents the energy saved against the shut-downs performed. It shows the *goodness* of the power-off actions performed.

[2] Current operation for the same data center and workload, but without applying energy-saving polices.



(a) Batch (b) Service

Fig. 2. Histogram of the number of tasks for workload.



(a) Batch (b) Service

Fig. 3. Workload job-duration histogram.

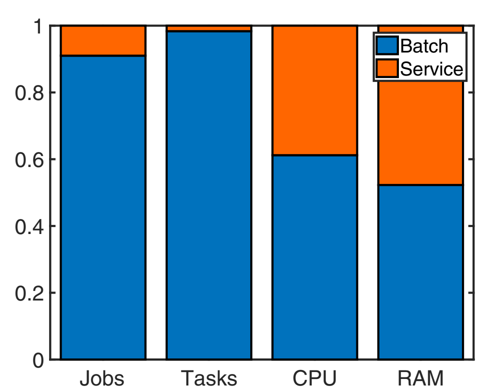


Fig. 4. Workload resource usage.

- *Idle resources*: Represents the percentage of resources in an idle state (turned on but not in use).

6.2. Performance indicators

The following indicators were selected as the most significant in the description of the impact of the various energy-efficiency policies on data-center performance.

- *Job queue time (first scheduled)*: Represents the time a job waits in the queue until its first task is scheduled.
- *Job queue time (fully scheduled)*: Represents the time a job waits in the queue until it is totally scheduled (not finished).

Table 5
Summary of energy savings for the best energy policies. N – *Never power off.* A – *Always power off.* R – *Random.* L – *Maximum load.* M – *Minimum free-capacity margin.* E – *Exponential.* G – *Gamma.*

Energy policy	Energy % vs. Current	Power offs (10^3)	KWh saved Shutt.	Idle resources %	KWh saved (10^3)	Cost savings ($)
N	100	0.00	n/a	42.21	0.00	0
A	80.25	64.52	1.72	8.35	110.83	15,517
R	80.73	39.16	2.76	9.18	108.15	15,141
L	**80.21**	67.20	1.65	**8.27**	**111.09**	**15,553**
M	82.35	9.04	10.96	11.97	99.04	13,866
E	82.34	9.01	**11.00**	11.95	99.12	13,877
G	82.7	**8.98**	10.82	12.56	97.12	13,597

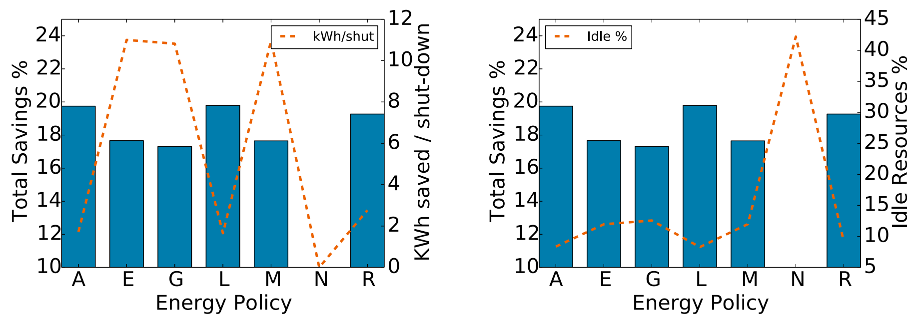
- *Job think time*: Represents the time needed for a schedule decision to be made.
- *Timed-out jobs*: A job is marked as timed out and left without scheduling when the scheduler completes 100 tries to schedule the same job, or 1000 tries of any task of the job. In all our experiments, the number of timed-out jobs is always 0.
- *Scheduler occupation fraction*: This represents the scheduler usage.

6.3. General results

In order to analyze and compare the performance of each family of policies, the best and exemplary energy policy from each family has been selected, in terms of the combination of energy-

Table 6
Summary of performance impact of best energy policies. B – *Batch* workload, S – *Service* workload.

Energy policy	Time first scheduled (s)				Time fully scheduled (s)				Sched. occu- pation(%)
	Mean		90p.		Mean		90p.		
	B	S	B	S	B	S	B	S	
N	0.19	0.19	0.63	0.67	0.19	0.19	0.63	0.67	14.30
A	0.22	0.22	0.78	0.84	0.30	0.32	0.92	0.95	15.18
R	0.20	0.21	0.72	0.80	0.25	0.27	0.80	0.84	14.79
L	0.22	0.22	0.78	0.83	0.31	0.33	0.94	0.95	15.20
M	0.19	0.19	0.63	0.67	0.19	0.19	0.63	0.67	14.30
E	0.19	0.19	0.63	0.67	0.19	0.19	0.63	0.67	14.30
G	0.19	0.19	0.63	0.67	0.19	0.19	0.63	0.67	14.30



(a) Savings vs. KWh/shut-down (b) Savings vs. Idle resources

Fig. 5. Energy-saving comparison. A – *Always power off.* E – *Exponential.* G – *Gamma.* L – *Maximum load.* M – *Minimum free-capacity margin.* N – *Never power off.* R – *Random.*

saving and performance results. Table 5 shows performance key indicators, while Table 6 shows energy related results. Fig. 5a and 5 b summarize and illustrate these numeric results.

From these results, several conclusions can be stated. In general, the more shut-downs there are, the more energy is saved, or from another point of view, the less idle the resources, the less energy wasted. Fig. 5b shows this behaviour, since the *Always power off* energy policy and other policies that tend to switch off resources are always the greatest energy savers, achieving savings of approximately 20%. This first conclusion provides evidence previously shown by Fernández-Montes, Gonzalez-Abril, Ortega, and Lefèvre (2012) in similar environments.

However, it should be borne in mind that the accuracy of the employed policies depends on the distribution of the data-center workload. The application of these policies without any previous knowledge of the workload and its distribution may be hard and might achieve sub-optimal results.

Fig. 5a shows that *Exponential, Gamma* and *Minimum free-capacity margin* policies perform fewer shut-down operations, but in a highly planned manner, and therefore the quantity of energy saved per shut-down operation is approximately 6 times better (from 2 to 12 kWh), and total savings are approximately 18%, which is only 2% less than the policies of the highest energy savings, while performing 85% less shut-down operations compared to those performed by *Always power off* and *Maximum load* policies.

In terms of costs, the saved energy adds up to a total of $15 K for 7 days, and hence, under similar conditions, this would indicate $60 K a month or $720 k a year.[3]

In terms of performance, Fig. 6a and 6 b show that the more shut-downs are performed, the more probability of causing a negative impact on the performance. This is noticeable for the *Always power off* and *Maximum load* policies. The negative impact in terms of queue time is shown on the queue-time parameters, such as *Job queue time (first scheduled)* and *Job queue time (fully scheduled)* parameters, which suffer a mean impact of 15% and 60%, respectively compared to those of the base/current scenario. The *Random* policy acts as an intermediate stage between the two previously stated sides. The queue-time parameters, such as *Job queue time (first scheduled)* and *Job queue time (fully scheduled)*, suffer a mean impact of 5% and 15%, respectively.

On the other hand, once again, *Exponential, Gamma* and *Margin* energy policies do not affect negatively to the performance, but achieve major energy savings (~ 18%).

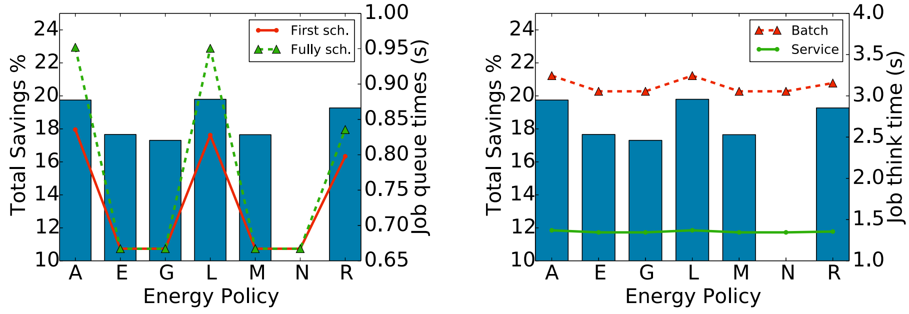
In order to better understand the behaviour of these energy policies, Fig. 7 shows the evolution of the resource state for each policy.

It should be borne in mind that there is a short-time period at the beginning until each policy reaches its normal pattern. This adjusting period occurs due to the *On* state of all the resources of the data center at the beginning of the simulation. Two groups of policies can be determined according to their behaviour. On one hand, the *Always power off, Maximum load* and *Random* policies suffer from the same problem: they try to adjust available resources to fit, as much as possible, the current workload demand, which leads to a high number of power on/off operations. Moreover, it can be observed that the time needed by the *Random* policy to adjust to workload changes is double that of the *Always power off* policy, since the *Random* policy performs half the number of shut-down operations compared to the *Always power off* policy.

On the other hand, prediction-based policies perform much smoother adjustments to the workload, therefore leading to a lower number of power on/off operations.

Finally, at the end of day #1, there is a peak of machines that are switched on for *Always power off*-like policies. Hence, it should be pointed out that maintaining a set of machines as a security margin can lead to the ability to satisfy the workload needs in a much more gradual way. Moreover, these workload peaks do not affect these prediction-based energy policies. Aggressive policies

[3] $0.14 per KwH was considered to compute economic costs.



(a) Savings vs. Job queue time (b) Savings vs. Job think time

Fig. 6. Comparison of energy savings vs. performance.

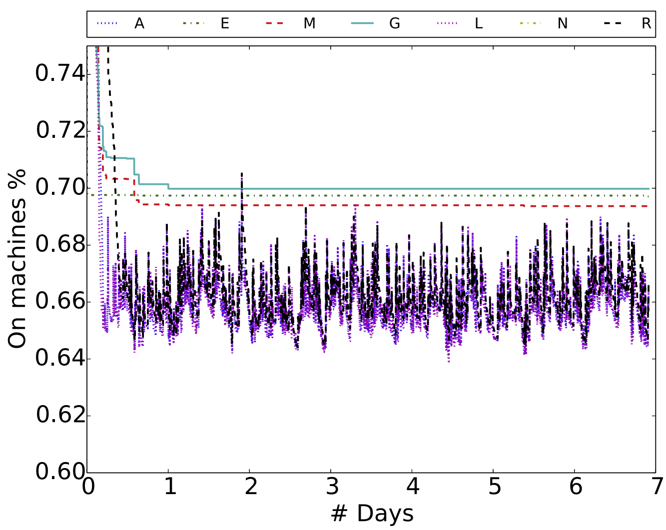


Fig. 7. Behaviour of energy policies.

solve these load bursts by switching on a large set of machines, even larger than actually needed for that moment.

The presented evidences lead us to recognize that controlled and prediction-based polices are preferable to deterministic policies.

6.4. Exponential policy: detailed results

Exponential power-off policies described in Section 3 show a high dependency on the *Lost factor* parameter. This parameter represents the percentage of resources that can not be used even if they are available because these resources are insufficient to hold the task. For example, lets consider a workload where all tasks will consume 1 GB of memory and 2 CPU cores. In this scenario, even if a machine has 900 MB of memory and 1.8 CPU cores available, these resources will be completely useless and should not be computed as available resources. The *Lost factor* allows the authors to take into account the useful available resources instead the total not-used resources.

In order to fully understand the results presented in this section, it should be borne in mind the nature of the workload employed: a vast majority are low-resource consuming jobs comprising very few tasks which are easily to serve. Due to this, the risk of not satisfying the requirements of these tasks is very low, tending to 0. In the other hand, very few jobs are composed of an enormous number of tasks, where it is almost impossible to serve their requirements. This means that the risk of not satisfying the requirements of these tasks tends towards 1. Due to this, the decision threshold μ has a lower impact in terms of performance and energy savings, unless a value extremely close to 0 or 1 is taken, whereby it behaves as the *Never power off* or *Always power off* policies, respectively.

In addition, the number of these high-demanding jobs is very low. This leads to a poor prediction when only a low number of the last jobs are taken into account. Thus, the *Window size* values evaluated are of less impact in terms of performance and energy savings than the *Lost factor* parameter.

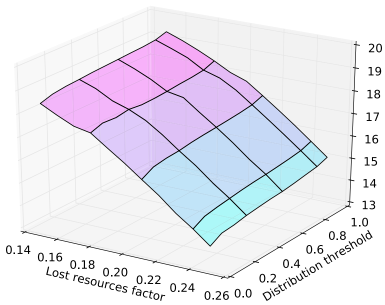
Fig. 8 presents the dependency on the *Lost factor* clearly. In terms of kWh saved per shut-down, as shown in Fig. 8b, the best results are reached when a *Lost factor* of 20% is considered. This value makes sense, because as stated in Section 5, our environment is designed to simulate the one presented in Lo, Cheng, Govindaraju, Ranganathan, and Kozyrakis (2015), which attains a level of utilization of 90% of resources without causing any noticeable negative impact.

- *Energy savings*: In terms of energy savings, as presented in Table 7 and in Fig. 8a, for low *Lost factor* values, the *Exponential* policy behaves similar to the *Always power off* policy, and achieves the highest rates of energy savings at the expense of a negative performance impact, as presented in Table 8. The higher this parameter increases, the lower the number of power-off cycles, and approaches the *Never power off* policy.
- *Performance*: In terms of performance, as presented in Table 8, the *Exponential* policy follows the same trend present in the energy savings. However, it can be observed that if 20% of resources are taken as unusable (*lost factor*), as suggested by the *kWh saved per shut-down* parameter, then a virtually non-negative impact in terms of performance is imposed. Moreover only $\sim$2% more of energy is consumed compared to *Always power off* policy, but only $\sim$15% of the number of shut-downs is performed. In addition this is consistent with the *Minimum free-capacity margin* policy. Finally, if the *Lost factor* value continues to rise above $\sim$20%, it does not impact positively in terms of performance, but negatively in terms of energy savings.

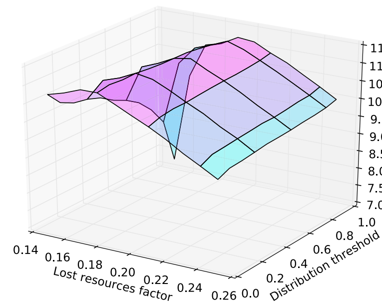
6.5. Gamma policy: detailed results

As described for the *Exponential* policy, the exponential nature of the generated workload links the *Gamma* policy performance impact and energy savings to the *Lost factor* parameter, whereby the rest of the parameters, *Window size* and decision threshold μ, hold a minor influence.

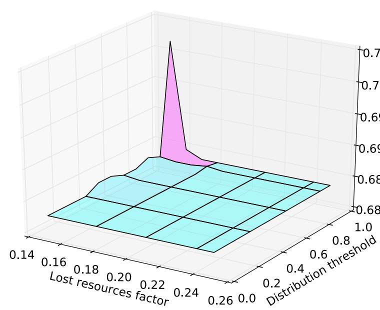
In terms of behaviour, the *Gamma* policy follows the same trends present in the *Exponential* policy. However, due to the difference in the predictive model construction, the *Gamma* policy be-



(a) Energy savings vs. Exponential parametrization



(b) kWh saved per shut-down vs. Exponential parametrization



(c) Queue time vs. Exponential parametrization

Fig. 8. Energy savings and performance indicators in Exponential parametrization.

Table 7
Energy savings for *Exponential* policies. Exponential parameterization: [*Decision threshold* μ, *Window size, Lost factor*].

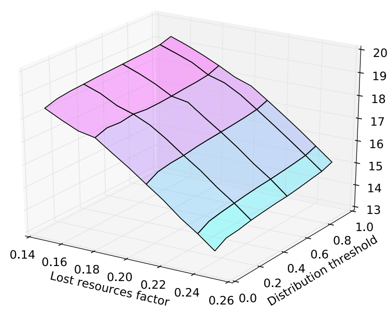
Energy policy		Energy	Power	KWh	Idle	KWh	Cost
Acr.	Params	% vs. Current	offs (10^3)	saved Shutt.	resources %	saved (10^3)	savings ($)
N	n/a	100	n/a	n/a	42.21	n/a	0
A	n/a	80.25	64.52	1.72	8.35	110.83	15,517
R	[0.50]	80.73	39.16	2.76	9.18	108.15	15,141
E	[0.30, 25, 0.10]	80.01	64.94	1.73	7.93	112.21	15,710
E	[0.30, 25, 0.15]	80.68	19.00	5.71	9.11	108.42	15,179
E	**[0.30, 25, 0.20]**	82.34	9.01	11.00	11.95	99.12	13,877
E	[0.30, 25, 0.25]	85.06	8.54	9.82	16.62	83.83	11,736
E	[0.30, 25, 0.30]	88.34	8.03	8.16	22.22	65.47	9166

Table 8
Performance results for the *Exponential* energy-efficiency policy.

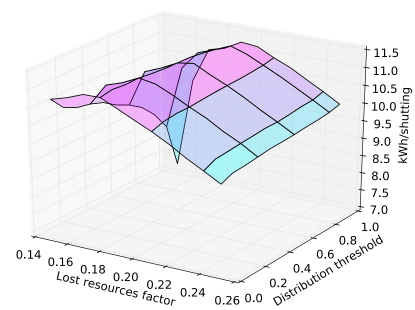
Energy policy	Time first scheduled (s)				Time fully scheduled (s)				Sched.
	Mean		90p.		Mean		90p.		occu-
	B	S	B	S	B	S	B	S	pation(%)
N	0.19	0.19	0.63	0.67	0.19	0.19	0.63	0.67	14.30
A	0.22	0.22	0.78	0.84	0.30	0.32	0.92	0.95	15.18
R [0.50]	0.20	0.21	0.72	0.80	0.25	0.27	0.80	0.84	14.79
E [0.30, 25, 0.10]	0.22	0.22	0.77	0.84	0.30	0.32	0.92	0.95	15.15
E [0.30, 25, 0.15]	0.19	0.20	0.65	0.69	0.20	0.21	0.68	0.69	14.45
E [0.30, 25, 0.20]	0.19	0.19	0.63	0.67	0.19	0.19	0.63	0.67	14.30
E [0.30, 25, 0.25]	0.19	0.19	0.63	0.67	0.19	0.19	0.63	0.67	14.30
E [0.30, 25, 0.30]	0.19	0.19	0.63	0.67	0.19	0.19	0.63	0.67	14.30

Table 9
Energy-saving results for Gamma energy policy. Gamma parameterization: [*Decision threshold μ, Window size, Lost factor*].

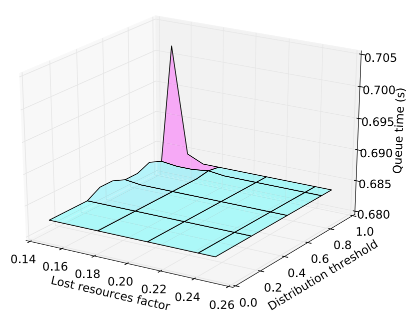
Energy policy		Energy	Power	KWh	Idle	KWh	Cost
Acr.	Params	%vs. Current	offs (10^3)	saved Shutt.	resources %	saved (10^3)	savings ($)
N	n/a	100	n/a	n/a	42.21	0.00	0
A	n/a	80.25	64.52	1.72	8.35	110.83	15,517
R	[0.50]	80.73	39.16	2.76	9.18	108.15	15,141
G	[0.90, 25, 0.10]	80.28	60.21	1.84	8.40	110.67	15,494
G	[0.90, 25, 0.15]	81.03	15.08	7.06	9.71	106.45	14,902
G	**[0.90, 25, 0.20]**	82.7	8.98	10.82	12.56	97.12	13,597
G	[0.90, 25, 0.25]	85.36	8.49	9.67	17.13	82.14	11,500
G	[0.90, 25, 0.30]	88.44	7.99	8.12	22.40	64.89	9084



(a) Energy savings vs Gamma parametrization



(b) kWh saved per shutting vs Gamma parametrization



(c) Queue time vs Gamma parametrization

Fig. 9. Energy savings and performance indicators in Gamma parametrization.

haves slightly less aggressively in terms of number of shut-downs applied.

- *Energy savings*: In terms of energy savings, as presented in Table 9 and in Fig. 9a and stated in the *Exponential* policy, if the *Lost factor* is too low, then the *Gamma* policy behaves like the *Always power off* policy, in that it achieves the highest rates of energy savings at the expense of a negative performance impact, as presented in Table 10. The higher this parameter increases, the lower the number of power-off cycles, and approaches the *Never power off* policy.
- *Performance*: In terms of performance, as presented in Table 10, the *Gamma* policy follows the same trend present in the energy savings. However, it can be observed that if 20% of resources are taken as unusable (*Lost factor*), as suggested by the *kWh saved per shut-down* parameter, then a virtually non-negative impact in terms of performance is imposed. Moreover only $\sim$2.5% more of energy is consumed compared to *Always power off* policy, but only $\sim$13% of the number of shut-downs is performed. In addition this is consistent with the *Minimum free-capacity margin* and *Exponential* policies. Finally, if the *Lost factor* value continues to rise above $\sim$20%, then it does not impact positively in terms of performance, but negatively in terms of energy savings.

Table 10
Performance results for the Gamma energy policy.

Energy policy	Time first scheduled (s)				Time fully scheduled (s)				Sched. occu-pation(%)
	Mean		90p.		Mean		90p.		
	B	S	B	S	B	S	B	S	
N	0.19	0.19	0.63	0.67	0.19	0.19	0.63	0.67	14.30
A	0.22	0.22	0.78	0.84	0.30	0.32	0.92	0.95	15.18
R [0.50]	0.20	0.21	0.72	0.80	0.25	0.27	0.80	0.84	14.79
G [0.90, 25, 0.10]	0.21	0.22	0.77	0.83	0.29	0.31	0.89	0.94	15.09
G [0.90, 25, 0.15]	0.19	0.20	0.65	0.68	0.20	0.20	0.66	0.69	14.40
G [0.90, 25, 0.20]	0.19	0.19	0.63	0.67	0.19	0.19	0.63	0.67	14.30
G [0.90, 25, 0.25]	0.19	0.19	0.63	0.67	0.19	0.19	0.63	0.67	14.30
G [0.90, 25, 0.30]	0.19	0.19	0.63	0.67	0.19	0.19	0.63	0.67	14.30

7. Conclusions

We have empirically proven that a suitable policy in data centers can save a considerable amount of energy and reduce the pollution of CO_2 in the atmosphere. Industrial partners willing to deploy this kind of energy-saving policies would have a direct positive impact on their competitiveness: in addition to become greener by minimizing the environmental impact, these policies may notably reduce their operation costs.

Several energy-saving policies have been explained, and their advantages and disadvantages have been presented, which outline which policy is more suitable for each data-center operational environment and administrator criteria. The behaviours of these energy policies are also consistent for various scheduler strategies.

This work characterizes the impact of these power-off policies. Unlike the presented related work, it is focused on the use of a Best-fit-like VM allocation heuristic. In addition, these power-off policies are applied at the data center operating system/resource manager level, not to a framework or to a subsystem. This approach makes it possible to apply the proposed power-off policies to any framework that can run as a VM/Linux container on the data center.

In this work, we go beyond the presented state of the art by focusing on the development of realistic, empirically-driven and production-ready energy policies that have a minor impact on data-center performance. These policies are simulated on a realistic environment that has been contrasted with real-life production systems, such as those of Google data centers. We can point out the following strengths in our research method: (a) A clear description of data-center utilization and workload distribution, which follow the industry trends; (b) A detailed explanation on the workload parameters, classification, generation and heterogeneity; (c) A complete description of the scheduling model and algorithms employed; and (d) A detailed explanation on the impact on both the main goals of our system: energy-efficiency and performance. On the other hand, the greatest weaknesses of this work include: (a) The lack of means to contrast the provided results with a real-life system; and (b) The lack of some real-life system aspects in simulation, such as task inter-dependency, networking and data-related considerations. However, we plan to overcome these limitations in future steps of this research.

The authors consider that prediction-based policies present much better behaviour for the data center, since they perform a much lower number of power-off cycles and save considerable amounts of energy. Moreover, it is also shown that it is possible to save energy by switching off machines and maintaining QoS and SLA levels, even for data centers in great demand.

For future work, the authors aim to focus on the following research directions:

1. Development of energy-efficiency policies based on machine learning, especially deep learning techniques.
2. Utilization of no-monolithic scheduling frameworks, such as two-level, shared state, distributed and hybrid schedulers.
3. Development of an intelligent system that may dynamically change the scheduling framework depending on environmental and workload-related parameters, as well as the study of the impact of such a system in terms of energy efficiency and performance.
4. Development of new simulation features, such as new workload patterns, task inter-dependency, networking and data-related considerations.

Acknowledgments

This research is supported by the VPPI - University of Sevilla.

References

Abdul-Rahman, O. A., & Aida, K. (2014). Towards understanding the usage behavior of Google cloud users: the mice and elephants phenomenon. In *Proceedings of the IEEE international conference on cloud computing technology and science (Cloudcom), Singapore* (pp. 272–277).
Amur, H., Cipar, J., Gupta, V., Ganger, G. R., Kozuch, M. A., & Schwan, K. (2010). Robust and flexible power-proportional storage. In *Proceedings of the first ACM symposium on cloud computing* (pp. 217–228). ACM.
Andersen, D. G., & Swanson, S. (2010). Rethinking flash in the data center. *IEEE Micro, 30*(4), 52–54.
Beloglazov, A., Abawajy, J., & Buyya, R. (2012). Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems, 28*(5), 755–768.
Beloglazov, A., & Buyya, R. (2010). Energy efficient resource management in virtualized cloud data centers. In *Proceedings of the tenth IEEE/ACM international conference on cluster, cloud and grid computing* (pp. 826–831). IEEE Computer Society.
Beloglazov, A., & Buyya, R. (2012). Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurrency and Computation: Practice and Experience, 24*(13), 1397–1420.
Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., et al. (2008). Bigtable: A distributed storage system for structured data. *ACM Transactions on Computer Systems (TOCS), 26*(2), 4.
De Assuncao, M. D., Gelas, J.-P., Lefevre, L., & Orgerie, A.-C. (2012). The green gridâ;;5000: Instrumenting and using a grid with energy sensors. In *Remote instrumentation for Escience and related aspects* (pp. 25–42). Springer.
Di, S., Kondo, D., & Franck, C. (2013). Characterizing cloud applications on a Google data center. In *Proceedings of the forty-second international conference on parallel processing (ICPP). Lyon, France*.
Duy, T. V. T., Sato, Y., & Inoguchi, Y. (2010). Performance evaluation of a green scheduling algorithm for energy savings in cloud computing. In *Proceedings of the IEEE international symposium on parallel and distributed processing, workshops and Ph.D. forum (IPDPSW)* (pp. 1–8). IEEE.
El-Sayed, N., Stefanovici, I. A., Amvrosiadis, G., Hwang, A. A., & Schroeder, B. (2012). Temperature management in data centers: Why some (might) like it hot. *ACM SIGMETRICS Performance Evaluation Review, 40*(1), 163–174.
Fan, X., Weber, W.-D., & Barroso, L. A. (2007). Power provisioning for a warehouse-sized computer. In *ACM sigarch computer architecture news: 35* (pp. 13–23). ACM.
Femal, M. E., & Freeh, V. W. (2005). Boosting data center performance through non-uniform power allocation. In *Proceedings of the second international conference on autonomic computing (ICAC'05)* (pp. 250–261). IEEE.
Fernández-Montes, A., Fernández-Cerero, D., González-Abril, L., Álvarez-García, J. A., & Ortega, J. A. (2015). Energy wasting at internet data centers due to fear. *Pattern Recognition Letters, 67*, 59–65.

Fernández-Montes, A., Gonzalez-Abril, L., Ortega, J. A., & Lefèvre, L. (2012). Smart scheduling for saving energy in grid computing. *Expert Systems with Applications, 39*(10), 9443–9450.

Fernández-Cerero, D., Jakóbik, A., Grzonka, D., Kołodziej, J., & Fernández-Montes, A. (2018). Security supportive energy-aware scheduling and energy policies for cloud environments. *Journal of Parallel and Distributed Computing, 119*, 191–202. doi:10.1016/j.jpdc.2018.04.015.

Jakóbik, A., Grzonka, D., Kolodziej, J., Chis, A. E., & González-Vélez, H. (2017). Energy efficient scheduling methods for computational grids and clouds. *Journal of Telecommunications and Information Technology, 1*, 56.

Juarez, F., Ejarque, J., & Badia, R. M. (2018). Dynamic energy-aware scheduling for parallel task-based application in cloud computing. *Future Generation Computer Systems, 78*, 257–271. https://doi.org/10.1016/j.future.2016.06.029.

Kaushik, R. T., & Bhandarkar, M. (2010). Greenhdfs: Towards an energy-conserving, storage-efficient, hybrid hadoop compute cluster. In *Proceedings of the usenix annual technical conference* (p. 109).

Khaneja, G. (2015). *An experimental study of monolithic scheduler architecture in cloud computing systems*. Ph.D. thesis. University of Illinois at Urbana-Champaign.

Koomey, J. (2011). *Growth in data center electricity use 2005 to 2010: A report by Analytical Press, completed at the request of The New York Times*: 9. Analytic Press.

Lee, Y. C., & Zomaya, A. Y. (2012). Energy efficient utilization of resources in cloud computing systems. *The Journal of Supercomputing, 60*(2), 268–280.

Liu, Z., & Cho, S. (2012). Characterizing machines and workloads on a Google cluster. In *Proceedings of the eight international workshop on scheduling and resource management for parallel and distributed systems (SRMPDS). Pittsburgh, PA, USA*.

Lo, D., Cheng, L., Govindaraju, R., Ranganathan, P., & Kozyrakis, C. (2015). Heracles: improving resource efficiency at scale. In *ACM sigarch computer architecture news: 43* (pp. 450–462). ACM.

Luo, X., Wang, Y., Zhang, Z., & Wang, H. (2013). Superset: A non-uniform replica placement strategy towards high-performance and cost-effective distributed storage service. In *Proceedings of the international conference on advanced cloud and big data (CBD)* (pp. 139–146). IEEE.

Miyoshi, A., Lefurgy, C., Van Hensbergen, E., Rajamony, R., & Rajkumar, R. (2002). Critical power slope: Understanding the runtime effects of frequency scaling. In *Proceedings of the sixteenth international conference on supercomputing* (pp. 35–44). ACM.

Orgerie, A.-C., Lefèvre, L., & Gelas, J.-P. (2008). Save watts in your grid: Green strategies for energy-aware framework in large scale distributed systems. In *Proceedings of the fourteenth IEEE international conference on parallel and distributed systems* (pp. 171–178). IEEE.

Reiss, C., Tumanov, A., Ganger, G. R., Katz, R. H., & Kozuch, M. A. (2012a). Heterogeneity and dynamicity of clouds at scale: Google trace analysis. *ACM symposium on cloud computing (SOCC). San Jose, CA, USA*.

Reiss, C., Wilkes, J., & Hellerstein, J. L. (2011). Google cluster-usage traces: format + schema. *Technical Report*. Mountain View, CA, USA: Google Inc.

Reiss, C., Wilkes, J., & Hellerstein, J. L. (2012b). Obfuscatory obscanturism: Making workload traces of commercially-sensitive systems safe to release. In *Proceedings of the third international workshop on cloud management (Cloudman)* (pp. 1279–1286). Maui, HI, USA: IEEE.

Ricciardi, S., Careglio, D., Sole-Pareta, J., Fiore, U., Palmieri, F., et al. (2011). Saving energy in data center infrastructures. In *Proceedings of the first international conference on data compression, communications and processing (CCP)* (pp. 265–270). IEEE.

Schwarzkopf, M., Konwinski, A., Abd-El-Malek, M., & Wilkes, J. (2013). Omega: Flexible, scalable schedulers for large compute clusters. In *Proceedings of the eight ACM European conference on computer systems* (pp. 351–364). ACM.

Sharma, R. K., Bash, C. E., Patel, C. D., Friedrich, R. J., & Chase, J. S. (2005). Balance of power: Dynamic thermal management for internet data centers. *IEEE Internet Computing, 9*(1), 42–49.

Sohrabi, S., Tang, A., Moser, I., & Aleti, A. (2016). Adaptive virtual machine migration mechanism for energy efficiency. In *Proceedings of the fifth international workshop on green and sustainable software* (pp. 8–14). ACM.

Thereska, E., Donnelly, A., & Narayanan, D. (2011). Sierra: Practical power-proportionality for data center storage. In *Proceedings of the sixth conference on computer systems* (pp. 169–182). ACM.