Productive Efficiency of Energy-Aware Data Centers

Damián Fernández-Cerero 1,*, Alejandro Fernández-Montes 1 and Francisco Velasco 2

1 Department of Computer Languages and Systems, University of Seville, 41012 Sevilla, Spain; afdez@us.es
2 Department of Applied Economy I, University of Seville, 41018 Sevilla, Spain; velasco@us.es

* Correspondence: damiancerero@us.es; Tel.: +34-954-559-531

Received: 17 July 2018; Accepted: 1 August 2018; Published: 8 August 2018

Abstract: Information technologies must be made aware of the sustainability of cost reduction. Data centers may reach energy consumption levels comparable to many industrial facilities and small-sized towns. Therefore, innovative and transparent energy policies should be applied to improve energy consumption and deliver the best performance. This paper compares, analyzes and evaluates various energy efficiency policies, which shut down underutilized machines, on an extensive set of data-center environments. Data envelopment analysis (DEA) is then conducted for the detection of the best energy efficiency policy and data-center characterization for each case. This analysis evaluates energy consumption and performance indicators for natural DEA and constant returns to scale (CRS). We identify the best energy policies and scheduling strategies for high and low data-center demands and for medium-sized and large data-centers; moreover, this work enables data-center managers to detect inefficiencies and to implement further corrective actions.

Keywords: data envelopment analysis; return-to-scale; cloud computing; efficiency; energy policies

1. Introduction

Data centers, which constitute the computational muscle for cloud computing, can be compared in energy consumption to many industrial facilities and towns. The latest trends show that these infrastructures represent approximately 2% of global energy consumption [1], with a 5% annual growth rate [2].

The data envelopment analysis mathematical model enables the management organizational divisions to measure the performance of an organization by providing the relative efficiency of each organizational unit. This relative efficiency measurement can be applied to a set of decision-making units, also known as DMUs, or for productive efficiency. The productive efficiency, also called technical efficiency, involves a collection of inputs (the resources needed for the production) and outputs (the production achieved). To this end, DEA constructs an “efficiency frontier” which places the relative performance of all units so these can be contrasted. This method is notably well-suited for the examination of the behavior of complex relations, even unknown, between numerous inputs and outputs, where the decisions made are affected by a level of uncertainty [3]. Moreover, DEA has been used both in private [4,5] and in public contexts [6–9].

Many initiatives have emerged looking for the decrease of the consumption of energy and the CO₂ trace of data-centers, especially those of a medium and large size. These facilities are composed of thousands and even tens of thousands of machines.

A substantial part of these initiatives focuses on the improvement of the Power Usage Effectiveness (PUE), that is the amount of energy consumed in non-computational tasks, such as power supply, cooling and networking components. This accounts for more than half of the energy consumption of an Internet data-center (IDC).

Several strategies are proposed to significantly improve energy efficiency in large-scale clusters [10]: cooling and temperature management [11,12]; power proportionality for CPU and...
memory hardware components [13,14]; fewer energy-hungry and non-mechanical hard disks [15]; and new proposals for energy distribution [16].

On the other hand, almost 50% of energy is consumed by computational servers to satisfy the incoming workload. The job arrival is not stable over time, but usually presents correlative low and high periods, such as those present in day/night and weekday/weekend workload patterns.

Such scenarios present a huge opportunity for the improvement of energy efficiency through proper scheduling and through the application of low-energy consumption modes to servers, since keeping servers in an idle state is extremely energy-inefficient. Many energy-aware schedulers, which aim to raise server usage, have been proposed in order to free up the maximum amount of machines so that they may put into hibernation [17–19]. In addition to these schedulers, several energy-conservation strategies may be applied in virtualized environments, such as the consolidation and migration of virtual machines [20,21].

Other strategies focus on the reduction of energy consumption in specific scenarios, such as those of distributed file systems [22,23].

The most aggressive approach involves the shut-down of underutilized servers in order to minimize energy consumption. Several shut-down policies have been proposed for grid computing environments in [24]. This strategy is yet to be widely implemented in working data-centers since a natural reticence to worsening QoS is usually present in data-center operators [25].

The innovation of the research presented in this paper involves the utilization of data envelopment analysis (DEA) as a mathematical technique to compare the efficiency regarding the consumption of energy and the performance of various workload scenarios, scheduling models and energy efficiency policies. This efficiency analysis enables data-center operators to make appropriate decisions about the number of machines, the scheduling solution and the shut-down strategy that must be applied so that data-centers run optimally. The final goal is the maximization of the productive efficiency, which is computed as the amount of energy consumed to serve a workload with a determined performance.

The major contributions of this paper can be summarized as follows:

1. Extensive empirical experimentation and analysis of various cloud-computing scenarios with a trustworthy and detailed simulation tool.
2. Impact analysis in terms of the energy consumption and performance of several energy efficiency policies, which shut-down idle machines by means of data envelopment analysis.
3. DEA-conducted analysis of the performance impact and energy consumption of a set of scheduling models for large-scale data-centers.
4. Empirical determination and proposal of corrective actions to achieve optimal efficiency.

The work is organized as follows. In Section 2, the authors introduce the current literature for the utilization of DEA presented for various areas, as well as the DEA model employed in this work. In Section 3, we briefly explain the set of energy efficiency policies that shut down idle servers. The scheduling models considered are explained in Section 4. In Section 5, the tool used for the simulation, the experimental environment, the energy model and DEA inputs/outputs are presented. Natural constant returns to scale (CRS) DEA results are described and analyzed in Section 6. Finally, we summarize this paper and present conclusions in Section 7.

2. Data Envelopment Analysis Model

Data envelopment Analysis (DEA) is a method that analyzes the connections between the outputs and inputs required in a production process in order to establish the efficiency frontiers [26]. This non-parametric technique was first described for the determination of the efficiency of DMUs by [27] and was formally defined by [28]. DEA has been proposed to measure the efficiency in various areas of operations research and management science [29–32]. Moreover, it has been applied to measure the environmental performance by other authors [33–40], who describe the gains of this method in the field of environmental management, which is a matter of undoubted relevance.
where the unrestricted parameter $\xi$ which are unknown and are used for the connection of the input and output vectors by means of a (see [37] for a better understanding): vary the given rates, which implies that the system presents some inefficiencies.

Energies the inputs can be decreased when the positive slack variables composite unit, considering inputs such as: $\sum_{i=1}^{n} x_{ij} \lambda_j = 0$ in this work to minimize the influence of slack variables. If the restriction $\epsilon\sum_{r=1}^{n} \lambda_{rj} = 1$ is added to Model (1), then the obtained model is a VRS (Model (1')). Further, $d_i^r, i = 1, \ldots, m, d_j^s, r = 1, \ldots, s$ and $d_k^h, f = 1, \ldots, h$ are all slack variables which are related to inputs, desirable and undesirable outputs, respectively. $\lambda = (\lambda_1, \ldots, \lambda_n)^T$ are structural or intensity variables, which are unknown and are used for the connection of the input and output vectors by means of a convex combination. $R$ is the range resolute through the lower and upper limits of inputs, desirable outputs and undesirable outputs, denoted by:

$$R_i^f = (m + s + h)^{-1} \left( \max\{x_{ij}/j = 1, \ldots, n\} - \min\{x_{ij}/j = 1, \ldots, n\} \right)$$

$$R_j^g = (m + s + h)^{-1} \left( \max\{g_{ij}/j = 1, \ldots, n\} - \min\{g_{ij}/j = 1, \ldots, n\} \right)$$

$$R_k^b = (m + s + h)^{-1} \left( \max\{b_{ij}/j = 1, \ldots, n\} - \min\{b_{ij}/j = 1, \ldots, n\} \right)$$

The natural efficiency of the k-th policy is computed by the following CRS and radial VRS model (see [37] for a better understanding):

$$\begin{align*}
\max & \quad \xi + \epsilon \left( \sum_{i=1}^{m} R_i^r d_i^r + \sum_{r=1}^{s} R_j^g d_j^g + \sum_{f=1}^{h} R_k^b d_k^b \right) \\
\text{s.t.} & \quad \sum_{j=1}^{n} x_{ij} \lambda_j + d_i^r = x_{ik}, i = 1, \ldots, m, \\
& \quad \sum_{j=1}^{n} g_{ij} \lambda_j - d_j^g - \xi g_{rk} = g_{rk}, r = 1, \ldots, s, \\
& \quad \sum_{j=1}^{n} b_{ij} \lambda_j + d_j^s + \xi b_{fk} = b_{fk}, f = 1, \ldots, h, \\
& \quad d_i^r \geq 0, i = 1, \ldots, m, \\
& \quad d_j^g \geq 0, r = 1, \ldots, s, \\
& \quad d_k^h \geq 0, f = 1, \ldots, h, \\
& \quad \xi \text{ Unrestricted}
\end{align*}$$

where the unrestricted parameter $\xi$ denotes an unknown inefficiency rate expressing the gap between the efficiency frontier and an empirical group of undesirable and desirable outputs. The parameter $\epsilon$ takes the value of 0.0001 in this work to minimize the influence of slack variables. If the restriction $\sum_{j=1}^{n} \lambda_j = 1$ is added to Model (1), then the obtained model is a VRS (Model (1')).

The first restriction in equation systems ((1), (1')) explores the values of $\lambda_j$ to create a composite unit, considering inputs such as: $\sum_{j=1}^{n} x_{ij} \lambda_j = -d_i^r + x_{ik}, i = 1, \ldots, m$. The values of the inputs can be decreased when the positive slack variables $d_i^r$ are present. This may unuestionably vary the given rates, which implies that the system presents some inefficiencies.
In the same way, the second restriction, \( \sum_{i=1}^{m} s_{ij}^{*} \lambda_{j} = d_{i}^{b} + \xi g_{rk} + g_{rk}, r = 1, \ldots, s, \) indicates that the desirable outputs can be maintained or increased by making an increase of the slack variable \( d_{i}^{b} \) and a radial expansion \( \xi g_{rk} \).

The third restriction, \( \sum_{j=1}^{n} b_{ji} \lambda_{j} = -d_{i}^{f} - \xi b_{jk} + b_{jk}, f = 1, \ldots, h, \) shows the decrease of the inputs, and then, we could reduce the undesirable outputs both in their slack variables and radially.

The objective function considers that two origins of inefficiency may be established. A k-policy can be considered efficient when the following two conditions are met: (a) \( \xi = 0 \); (b) \( d_{i}^{b} = 0, d_{i}^{f} = 0 \), \( d_{f}^{b} = 0 \). In this case, the k-policy belongs to the efficiency frontier, since it fulfills the constraints present in equation systems ((1), (1*)) and consequently, the objective function takes a value of zero. Otherwise, the value of the objective function for non-efficient policies is greater than zero, due to possible displacements in the slack variables and radial movements.

The natural efficiency is then computed by:

\[
\theta^{*} = 1 - \left[ \xi^{*} + \epsilon \left( \sum_{i=1}^{m} R_{i}^{b} d_{i}^{f} + \sum_{r=1}^{s} R_{r}^{b} d_{r}^{f} + \sum_{f=1}^{h} R_{f}^{b} d_{f}^{f} \right) \right]
\]

The value of this unified efficiency measure ranges between zero and one. If the k-policy is efficient, then the objective function of equation systems ((1), (1*)) is zero, and hence, the efficiency score equals \( \theta^{*} = 1 \). Slack variables resulting in the optimality of the models represented in equation systems ((1), (1*)) show the level of inefficiency.

2.2. Managerial Disposability

The managerial efficiency of the k-th policy is evaluated by the following CRS and VRS radial model [37]:

\[
\begin{align*}
\text{s.t.} & \quad \sum_{i=1}^{m} x_{ij} \lambda_{j} - d_{i}^{b} = x_{ik}, i = 1, \ldots, m, \\
& \quad \sum_{i=1}^{m} s_{ij}^{*} \lambda_{j} - d_{i}^{f} - \xi g_{rk} = g_{rk}, r = 1, \ldots, s, \\
& \quad \sum_{j=1}^{n} b_{ji} \lambda_{j} + d_{i}^{f} + \xi b_{jk} = b_{jk}, f = 1, \ldots, h, \\
& \quad d_{i}^{b} \geq 0, i = 1, \ldots, m, \\
& \quad d_{i}^{f} \geq 0, r = 1, \ldots, s, \\
& \quad d_{f}^{f} \geq 0, f = 1, \ldots, h, \\
& \quad \xi \quad \text{Unrestricted}
\end{align*}
\]

Similarly, if the restriction \( \sum_{j=1}^{n} \lambda_{j} = 1 \) is added to Model (2), then the obtained model is a VRS (Model (2')). In this model (2), increasing the inputs is allowed since new technologies that emit less CO₂ emissions to the atmosphere can be used.

By using the VRS models, we can obtain the returns to scale (RTS) and damage to scale (DTS) (see [37] for a better understanding). It is clear that for the natural efficiency, the returns to scale have to be increasing, and for managerial efficiency, the damages to scale have to be decreasing. Otherwise, the technical units are not working well and should correct the imbalances, using the information of the efficient units to which they have to be similar (peers).

3. Energy Policies for Data Centers at a Glance

The following set of energy efficiency policies for shutting down underutilized machines have been developed in this work as an evolution of those presented in [24], which have been adapted to the more complex reality of the cloud-computing paradigm:

- Never: prevents any shut-down process.
- Always: shuts down every server running in an idle state.
- Load: shuts down machines when data-center load pressure fails to reach a given threshold.
- Margin: assures that a determined number of machines are turned on and available before shutting down any machine.
• Random: shuts down machines randomly by means of a Bernoulli distribution with parameter 0.5.
• Exponential: shuts down machines when the probability of one incoming task negatively impacting on the data-center performance is lower than a given threshold. This probability is computed by means of the exponential distribution.
• Gamma: shuts down machines when the probability of incoming tasks oversubscribing to the available resources in a particular time period is lower than a given threshold; this probability is computed by means of the Gamma distribution.

4. Scheduling Models for Data Centers at a Glance

Cluster schedulers constitute a core part of cloud computing systems, since they are responsible for optimal task assignation to computing nodes. Several degrees of parallelism have been added to overcome the limitations present in central monolithic scheduling approaches when complex and heterogeneous systems with a high number of incoming jobs are considered. The following scheduling models are studied in this work:

• Monolithic: A centralized and single scheduler is responsible for scheduling all tasks in the workload in this model [44]. This scheduling approach may be the perfect choice when real-time responses are not required [45,46], since the omniscient algorithm performs high-quality task assignations by considering all restrictions and features of the data-center [47–50] at the cost of longer latency [46]. The scheduling process of a monolithic scheduler, such as that given by Google Borg [51], is illustrated in Figure 1.

![Monolithic scheduler architecture. M, worker node; S, service task; B, batch task [52].](image)

• Two-level: This model achieves a higher level of parallelism by splitting the resource allocation and the task placement: a central manager blocks the whole cluster every time a scheduler makes a decision to offer computing resources to schedulers; and a set of parallel application-level schedulers performs the scheduling logic against the resources offered. This strategy enables the
development of sub-optimal scheduling logic for each application, since the state of the data-center is not shared with the central manager, nor with the application schedulers. The workflow of the Two-level schedulers [53,54] is represented in Figure 2.

**Figure 2.** Two-level scheduler architecture. C, commit; O, resource offer; SA-, scheduler agent [52].

- Shared-state schedulers: On the other hand, in shared-state schedulers, such as Omega [55], the state of the data-center is available to all the schedulers. The central manager coordinates all the simultaneous parallel schedulers, which perform the scheduling logic against an out-of-date copy of the state of the data-center. The scheduling decisions are then committed to the central manager, which strives to apply these decisions. The utilization of stale views of the cluster by the schedulers can result in conflicts, since the chosen resources may not longer be available. In such a scenario, the local view of the state of the data-center stored in the scheduler is refreshed before the repetition of the scheduling process. The workflow of the shared-state scheduling model is represented in Figure 3.
5. Methodology

In these next sections, the experimental environment designed for the implementation of the natural CRS DEA analysis is presented. The workflow followed in this work is shown in Figure 4.

5.1. Simulation Tool

The SCORE simulator [52] is employed in this work, since simulation is the best alternative in scenarios where the implementation of the proposed strategies on real large-scale data-centers remains unfeasible. This simulator provides us with the tools for the development and application of the energy policies described in Section 3 and the scheduling models presented in Section 4 on realistic large-scale cloud computing systems.
5.2. Environment and DMU Definition

Following the trends presented in [56,57], two utilization environments have been simulated in this paper for seven days of operation:

- the low-utilization scenario, which represents highly over-provisioned infrastructures and achieves an average utilization of approximately 30%.
- the high-utilization scenario, which represents facilities of a more efficient nature that use approximately 65% of available resources on average.

These scenarios are applied to three data-center sizes: (a) Small: composed of 1000 computing servers; (b) Medium: composed of 5000 computing servers; and (c) Large: composed of 10,000 computing servers. Each server is equipped with four CPU cores and 8 GB of RAM.

Decision-making units (DMUs) are defined by the following elements: (a) an energy efficiency policy; (b) a scheduling model; and (c) a workload scenario.

5.3. Energy Model

The following states are presented for each resource in the energy model applied in this work: (a) Idle: when the machine is not executing tasks; and (b) Busy: otherwise.

Let $t_{i \text{ idle}}$ represent the time the $i$-th resource is idle, and let $t_{i \text{ busy}}$ denote the time during which the machine is computing tasks. In the same way, $P_{i \text{ idle}}$ and $P_{i \text{ busy}}$ represent the power required for the machines to run in these states, respectively.

The time a machine spends on executing a job may be defined as follows:

$$t_{ij \text{ busy}} = \max_{t \in \text{Tasks}^i} C_t$$

where $\text{Tasks}^i$ represents the tasks of the $j$-th job assigned to $M_i$ and $C_t$ denotes the completion time of the $t$-th task of the $j$-th job.

In the same way, the total time a machine is executing tasks and the total time it is in an idle state may be defined as follows:

$$t_i^{\text{busy}} = \sum_{j=1}^{j} t_{ij \text{ busy}}$$

$$t_i^{\text{idle}} = t_i^{\text{total}} - t_i^{\text{busy}}$$

where $t_i^{\text{total}}$ represents the total operation time. Therefore, we can express the energy consumption as follows:

$$\sum_{i=1}^{n} (P_{i \text{ busy}} \ast t_{i \text{ busy}} + P_{i \text{ idle}} \ast t_{i \text{ idle}})$$

The considered power states, transitions and values for the energetic model are shown in Figure 5.
5.4. DEA Inputs and Outputs

The inputs and outputs considered in DEA analysis and representative experimentation values are shown in Tables 1 and 2, respectively. One hundred and eight DMUs were analyzed, which were the result of the combination of all energy policies, scheduling models, data-center sizes and workload types described in Sections 3, 4 and 5.2, respectively. However, for clarity, a subset of the most interesting eighteen DMUs is shown in this paper. Each environment presents the following inputs and outputs:

- **Inputs**: Two inputs are considered in this work: (a) the number of machines in the data-center (D.C.), as shown in Section 5.2; and (b) the number of shut-down operations performed. These inputs may be reduced or kept equal.

- **Outputs**: One desirable output and two undesirable outputs are considered in this paper: (a) the time used to perform tasks’ operations. The longer the time, the less idle the data-center. This good input can be maximized or kept equal; (b) the energy consumption of the data-center. The lower the energy consumption, the more efficient the data-center. This bad input may be reduced or kept equal; and (c) the average time jobs spend in a queue until they are scheduled. The shorter the time, the more performant the system is. This bad input may be reduced or kept equal.

![Figure 5. Machine power states [52].](image)

Table 1. DEA inputs and outputs. Action column arrows mean whether the input/output value may be decreased (down arrow), increased (up arrow) or kept equal.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Action</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Inputs</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data-center size</td>
<td>Number of machines in the data-center</td>
<td>↓ **</td>
</tr>
<tr>
<td>#shut-downs</td>
<td>Number of shut-down operations</td>
<td>↓ **</td>
</tr>
<tr>
<td><strong>Outputs</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Computation time</td>
<td>Total amount of useful task computation</td>
<td>↑ **</td>
</tr>
<tr>
<td>Energy consumption</td>
<td>Total data-center energy consumption</td>
<td>↓ **</td>
</tr>
<tr>
<td>Queue time</td>
<td>Average time until jobs are fully scheduled</td>
<td>↓ **</td>
</tr>
</tbody>
</table>
Table 2. Sample from the dataset for DEA analysis. The full dataset showing the results for the 108 DMUs analyzed can be found as the Supplementary Material. Energy policies, scheduling models, data-center sizes and workload types can be found in Sections 3, 4 and 5.2, respectively. D.C., data-center.

<table>
<thead>
<tr>
<th>DMU</th>
<th>Inputs</th>
<th>Outputs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Energy Policy</td>
<td>Scheduling Model</td>
</tr>
<tr>
<td>Always</td>
<td>Monolithic</td>
<td>High</td>
</tr>
<tr>
<td>Margin</td>
<td>Mesos</td>
<td>High</td>
</tr>
<tr>
<td>Gamma</td>
<td>Omega</td>
<td>High</td>
</tr>
<tr>
<td>Always</td>
<td>Mono.</td>
<td>Low</td>
</tr>
<tr>
<td>Exponential</td>
<td>Mesos</td>
<td>Low</td>
</tr>
<tr>
<td>Load</td>
<td>Omega</td>
<td>High</td>
</tr>
<tr>
<td>Margin</td>
<td>Mono.</td>
<td>High</td>
</tr>
<tr>
<td>Gamma</td>
<td>Mono.</td>
<td>High</td>
</tr>
<tr>
<td>Random</td>
<td>Mesos</td>
<td>High</td>
</tr>
<tr>
<td>Margin</td>
<td>Omega</td>
<td>High</td>
</tr>
<tr>
<td>Exponential</td>
<td>Omega</td>
<td>High</td>
</tr>
<tr>
<td>Margin</td>
<td>Omega</td>
<td>Low</td>
</tr>
<tr>
<td>Margin</td>
<td>Mono.</td>
<td>High</td>
</tr>
<tr>
<td>Gamma</td>
<td>Mono.</td>
<td>High</td>
</tr>
<tr>
<td>Margin</td>
<td>Omega</td>
<td>High</td>
</tr>
<tr>
<td>Gamma</td>
<td>Omega</td>
<td>High</td>
</tr>
<tr>
<td>Gamma</td>
<td>Mesos</td>
<td>Low</td>
</tr>
<tr>
<td>Margin</td>
<td>Omega</td>
<td>Low</td>
</tr>
</tbody>
</table>

6. Natural CRS DEA Results

The whole dataset included as an Appendix is analyzed by means of natural CRS and VRS DEA. However, only the most relevant natural CRS DEA results for the most representative DMUs, which are presented in Table 2, are described in this section.

An efficiency analysis depending on the data-center size and on the energy policy is shown in Tables 3 and 4. The following conclusions can be drawn:

- The best efficiency levels are achieved for small data-centers. The data-center size input is predominant in this group of DMUs, since no major differences between energy policies, scheduling frameworks and workload scenarios are present ($\sigma = 0.01$, $\bar{\tau} = 0.99$).
- Mid-size data-centers should use the margin energy policy and monolithic or Omega schedulers and should avoid all other energy policies and the Mesos scheduler. Moreover, high workload scenarios are also more efficient than low workload scenarios. In addition, the following DMUs achieve a good level of efficiency, but they do not belong to the efficiency frontier: (a) the DMU combining the Gamma energy policy and the monolithic or Omega schedulers; (b) the DMU combining the exponential energy policy and the Omega scheduler.
- No DMU is efficient in large-scale data-centers. However, the following DMUs present good levels of efficiency: (a) the DMUs combining the Gamma, exponential or margin energy policy with the high workload scenario and the monolithic scheduler; and (b) the DMUs combining the Gamma or margin energy policy with the high workload scenario and the Omega scheduler.
- In high-loaded scenarios, the monolithic scheduler presents the lowest deviation regardless of the data-center size ($\sigma = 0.32$).

We can determine that it is always inefficient to operate in a low utilization scenario in medium-sized and large data-centers. Moreover, both the margin and the probabilistic energy policies (Gamma and exponential) perform more efficiently than the rest of the energy policies, as shown in Figure 6. The monolithic scheduler seems to achieve good results even for large-scale data-centers,
while the two-level scheduling approach has a negative impact on data-center performance. However, the trends show that the performance of the monolithic scheduling approach suffers from degradation on larger data-centers and higher workload pressure, and hence, lower efficiency levels are to be expected if larger sizes and higher utilization scenarios are to be considered.

![Figure 6](image_url). Summary of DEA natural constant returns to scale (CRS) efficiency results for energy efficiency policies.

The actions proposed for the improvement of efficiency of the most relevant DMUs are shown in Table 5.

Table 3. Efficiency analysis for data-center sizes.

<table>
<thead>
<tr>
<th>Scheduling Model</th>
<th>Workload Scenario</th>
<th>Data-Center Size</th>
<th>Efficiency</th>
<th>Efficiency</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>1000</td>
<td>5000</td>
<td>10,000</td>
<td>σ</td>
</tr>
<tr>
<td>Monolithic High</td>
<td>1.00</td>
<td>0.60</td>
<td>0.37</td>
<td>0.32</td>
<td>0.66</td>
</tr>
<tr>
<td>Monolithic Low</td>
<td>0.98</td>
<td>0.33</td>
<td>0.18</td>
<td>0.43</td>
<td>0.49</td>
</tr>
<tr>
<td>Mesos High</td>
<td>1.00</td>
<td>0.47</td>
<td>0.18</td>
<td>0.41</td>
<td>0.55</td>
</tr>
<tr>
<td>Mesos Low</td>
<td>0.97</td>
<td>0.32</td>
<td>0.17</td>
<td>0.43</td>
<td>0.49</td>
</tr>
<tr>
<td>Omega High</td>
<td>1.00</td>
<td>0.62</td>
<td>0.27</td>
<td>0.36</td>
<td>0.63</td>
</tr>
<tr>
<td>Omega Low</td>
<td>0.97</td>
<td>0.32</td>
<td>0.17</td>
<td>0.43</td>
<td>0.49</td>
</tr>
<tr>
<td>σ</td>
<td>0.01</td>
<td>0.14</td>
<td>0.08</td>
<td></td>
<td></td>
</tr>
<tr>
<td>τ</td>
<td>0.99</td>
<td>0.44</td>
<td>0.23</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

0.40 0.55
Table 4. Efficiency analysis of energy policies.

<table>
<thead>
<tr>
<th>Scheduling Model</th>
<th>Energy Policy</th>
<th>Monolithic</th>
<th>Mesos</th>
<th>Omega</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1000 5000 10,000</td>
<td>1000 5000 10,000</td>
<td>1000 5000 10,000</td>
<td>1000 5000 10,000</td>
<td>σ</td>
</tr>
<tr>
<td>Always</td>
<td>0.99 0.33 0.18</td>
<td>0.99 0.33 0.18</td>
<td>0.99 0.33 0.18</td>
<td>0.99 0.33 0.18</td>
<td>0.37</td>
</tr>
<tr>
<td>Random</td>
<td>0.99 0.33 0.18</td>
<td>0.98 0.32 0.18</td>
<td>0.98 0.33 0.18</td>
<td>0.99 0.33 0.18</td>
<td>0.37</td>
</tr>
<tr>
<td>Load</td>
<td>0.99 0.33 0.18</td>
<td>0.99 0.33 0.18</td>
<td>0.99 0.33 0.18</td>
<td>0.99 0.33 0.18</td>
<td>0.37</td>
</tr>
<tr>
<td>Margin</td>
<td>0.99 0.66 0.42</td>
<td>0.99 0.53 0.18</td>
<td>0.99 0.66 0.30</td>
<td>0.31 0.63</td>
<td></td>
</tr>
<tr>
<td>Exp.</td>
<td>0.99 0.54 0.31</td>
<td>0.99 0.40 0.18</td>
<td>0.99 0.58 0.22</td>
<td>0.33 0.58</td>
<td></td>
</tr>
<tr>
<td>Gamma</td>
<td>0.99 0.58 0.38</td>
<td>0.98 0.47 0.18</td>
<td>0.99 0.61 0.29</td>
<td>0.31 0.61</td>
<td></td>
</tr>
</tbody>
</table>

6.1. Proposed Corrections for a Sample DMU

DMU #104 is selected to illustrate how corrective actions are proposed by DEA in order to achieve efficiency. This DMU is defined by the combination of the random energy efficiency policy, the Omega scheduling model and a low utilization workload scenario.

DMU #104 presents a natural efficiency of 0.1697. This means it is far from being efficient. The following corrective actions are suggested for it to belong to the efficiency frontier, as shown in Table 6:

- The time the data-center spends on task computation must be increased by 38.28 h (+83%).
- Energy consumption must be reduced by 193.88 MWh (−83%).
- The average time jobs wait in a queue must be reduced by 3.23 s (−83%).
- The number of servers must be reduced by 9190 (−92%).
- Shut-down operations must be reduced by 9680 (−24%).

In addition to these corrective actions, the peers this DMU should emulate are #13, #34 and #18. This means that the workload must be increased, and better energy efficiency policies, such as margin and always, must be used. The full dataset containing all the DMUs and DEA analysis and corrections can be found as Supplementary Material in the Appendix.

Some of the proposed changes involve the switching of the scheduling framework, which is hardly achievable with the current resource manager systems. To implement these corrections, a resource managing system able to dynamically change the scheduling framework during runtime would be necessary. Such a system is an interesting improvement to the current state of the art that the DEA analysis leads us to develop.
Table 5. Resulting proposed corrections following DEA analysis. Peer projections for a DMU indicate which DMU it should emulate. The following actions may be taken for each input and output: ↑ when the parameter must be increased; ↓ if the parameter must be reduced; and ± if no further actions are needed to achieve efficiency.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Always</td>
<td>Mono.</td>
<td>High</td>
<td>↑↑</td>
<td>↑↑</td>
<td>↑↑</td>
<td>↑↑</td>
<td>↑↑</td>
</tr>
<tr>
<td>10</td>
<td>Margin</td>
<td>Mesos</td>
<td>High</td>
<td>4 (88%)</td>
<td>↓</td>
<td>↑↓</td>
<td>↑↓</td>
<td>↓↓</td>
</tr>
<tr>
<td>18</td>
<td>Gamma</td>
<td>Omega</td>
<td>High</td>
<td>↑↑</td>
<td>↑↑</td>
<td>↑↑</td>
<td>↑↑</td>
<td>↑↑</td>
</tr>
<tr>
<td>19</td>
<td>Always</td>
<td>Mono.</td>
<td>Low</td>
<td>↑↑</td>
<td>↓</td>
<td>↑↓</td>
<td>↑↓</td>
<td>↓↓</td>
</tr>
<tr>
<td>29</td>
<td>Exp.</td>
<td>Mesos</td>
<td>Low</td>
<td>23 (56%)</td>
<td>↓</td>
<td>↑↑</td>
<td>↑↓</td>
<td>↓↓</td>
</tr>
<tr>
<td>33</td>
<td>Load</td>
<td>Omega</td>
<td>Low</td>
<td>31 (100%)</td>
<td>↓</td>
<td>↑↑</td>
<td>↑↓</td>
<td>↓↓</td>
</tr>
<tr>
<td>40</td>
<td>Margin</td>
<td>Mono.</td>
<td>High</td>
<td>↑↑</td>
<td>↑↑</td>
<td>↑↑</td>
<td>↑↑</td>
<td>↑↑</td>
</tr>
<tr>
<td>42</td>
<td>Gamma</td>
<td>Mono.</td>
<td>High</td>
<td>6 (59%)</td>
<td>↓</td>
<td>↑</td>
<td>↓↓</td>
<td>↓↓</td>
</tr>
<tr>
<td>44</td>
<td>Random</td>
<td>Mesos</td>
<td>High</td>
<td>7 (100%)</td>
<td>↓</td>
<td>↑</td>
<td>↓↓</td>
<td>↓↓</td>
</tr>
<tr>
<td>52</td>
<td>Margin</td>
<td>Omega</td>
<td>High</td>
<td>↑↑</td>
<td>↑↑</td>
<td>↑</td>
<td>↓↓</td>
<td>↑↓</td>
</tr>
<tr>
<td>53</td>
<td>Exp.</td>
<td>Omega</td>
<td>High</td>
<td>16 (63%)</td>
<td>↓</td>
<td>↑</td>
<td>↓↓</td>
<td>↓↓</td>
</tr>
<tr>
<td>70</td>
<td>Margin</td>
<td>Omega</td>
<td>Low</td>
<td>18 (72%)</td>
<td>↓</td>
<td>↑</td>
<td>↓↓</td>
<td>↓↓</td>
</tr>
<tr>
<td>76</td>
<td>Margin</td>
<td>Mono.</td>
<td>High</td>
<td>6 (55%)</td>
<td>↓</td>
<td>↑</td>
<td>↓↓</td>
<td>↓↓</td>
</tr>
<tr>
<td>78</td>
<td>Gamma</td>
<td>Mono.</td>
<td>High</td>
<td>6 (90%)</td>
<td>↓</td>
<td>↑</td>
<td>↓↓</td>
<td>↓↓</td>
</tr>
<tr>
<td>88</td>
<td>Margin</td>
<td>Omega</td>
<td>High</td>
<td>18 (95%)</td>
<td>↓</td>
<td>↑</td>
<td>↓↓</td>
<td>↓↓</td>
</tr>
<tr>
<td>90</td>
<td>Gamma</td>
<td>Omega</td>
<td>High</td>
<td>18 (96%)</td>
<td>↓</td>
<td>↑</td>
<td>↓↓</td>
<td>↓↓</td>
</tr>
<tr>
<td>102</td>
<td>Gamma</td>
<td>Mesos</td>
<td>Low</td>
<td>1 (49%)</td>
<td>↓</td>
<td>↑</td>
<td>↓↓</td>
<td>↓↓</td>
</tr>
<tr>
<td>104</td>
<td>Random</td>
<td>Omega</td>
<td>Low</td>
<td>13 (53%)</td>
<td>↓</td>
<td>↑</td>
<td>↓↓</td>
<td>↓↓</td>
</tr>
</tbody>
</table>

7. Conclusions and Policy Implications

In this work, we have confirmed the hypothesis that DEA constitutes a powerful tool for the analysis of technical efficiency in cloud-computing scenarios where large-scale data-centers provide the computational core.

Data envelopment analysis provides cloud-computing operators with the means for the identification of which data-center configuration better suits their requirements, both in terms of performance and energy efficiency.

This methodology allows us to analyze several energy efficiency policies that shut down idle servers, so that their behavior and differences can be compared in various data-center environments. It has been proven that policies based on a security margin and those that use statistical tools to predict the future workload, such as exponential and Gamma, deliver better results than policies based on data-center workload pressure and random strategies.

In addition, it has been empirically shown that even under medium and high workload pressure, in data-centers composed of up to 10,000 machines, monolithic schedulers perform better than other scheduling models, such as the two-level and shared-state approaches.
Table 6. Corrections proposed for DMU #104.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Original Value</th>
<th>Radial Movement</th>
<th>Slack Movement</th>
<th>Projected Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output Computation (h)</td>
<td>46.09</td>
<td>+83%</td>
<td>0</td>
<td>84.37</td>
</tr>
<tr>
<td>Output MWh consumed</td>
<td>233.50</td>
<td>−83%</td>
<td>0</td>
<td>39.62</td>
</tr>
<tr>
<td>Output Queue time (ms)</td>
<td>3.80</td>
<td>−83%</td>
<td>0</td>
<td>0.6</td>
</tr>
<tr>
<td>Input #Servers</td>
<td>10,000</td>
<td>0</td>
<td>−9190</td>
<td>810</td>
</tr>
<tr>
<td>Input #Shut-downs</td>
<td>40,772</td>
<td>0</td>
<td>−9680</td>
<td>31,092</td>
</tr>
</tbody>
</table>

Listing of Peers:

<table>
<thead>
<tr>
<th>Peer</th>
<th>Lambda Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>#13</td>
<td>53%</td>
</tr>
<tr>
<td>#34</td>
<td>36%</td>
</tr>
<tr>
<td>#18</td>
<td>11%</td>
</tr>
</tbody>
</table>

Finally, cloud-computing infrastructure managers are provided with empirical knowledge of which data-centers are not being used optimally, and hence, they can make decisions regarding the shut-down of machines in order to achieve higher utilization levels of the cloud-computing system as a whole.

As future work related to the limitations of the presented work, we may include:

- The addition of different kind of workload patterns, as well as real workload traces.
- The analysis of other scheduling models, such as distributed and hybrid models.
- The development of a new-generation resource-managing system that could dynamically apply the optimal scheduling framework depending on the environment and workload.
- The analysis of simulation data with other DEA approaches, such as Bayesian and probabilistic models, which could minimize the impact of the noise in current DEA models.


Author Contributions: D.F.-C. and A.F.-M. conceived of and designed the experiments. D.F.-C. performed the experiments. D.F.-C., A.F.-M. and V.F. analyzed the data. D.F.-C. and F.V. contributed reagents/materials/analysis tools. D.F.-C., A.F.-M. and F.V. wrote the paper.

Funding: This research was funded by VI Plan Propio de Investigación y Transferencia—University of Seville 2018 grant number 2018/0000520.

Acknowledgments: The research is supported by the VI Plan Propio de Investigación y Transferencia (VIPPI), University of Seville.

Conflicts of Interest: The authors declare no conflict of interest.

References
1. Koomey, J. Growth in Data Center Electricity Use 2005 to 2010; Analytical Press: Piedmont, CA, USA, 1 August 2011.


15. Andersen, D.G.; Swanson, S. Rethinking flash in the data-center. *IEEE Micro* 2010, 30, 52–54. [CrossRef]


43. Cooper, W.W.; Park, K.S.; Pastor, J.T. RAM: A range adjusted measure of inefficiency for use with additive models, and relations to other models and measures in DEA. *J. Prod. Anal.* 1999, 11, 5–42. [CrossRef]


