

Improving the energy-efficiency of Virtual Data Centers in an IT service provider through proactive fuzzy rules-based multicriteria decision making

Alberto Cocaña-Fernández ·
Julio Rodríguez-Soares · Luciano Sánchez ·
José Ranilla

Received: date / Accepted: date

Abstract A proactive multicriteria mechanism for Virtual Datacenter optimization through server consolidation is proposed. In contrast with previous works where heuristic mechanisms were designed using expert knowledge, the new proactive approach uses multiobjective evolutionary algorithms to learn fuzzy rule-based systems that determine optimal reallocation decisions according to the preferences of the datacenter operator and a prediction of the load. Experimental evaluations based on an actual IT service provider show that the proactive mechanism is capable of improving energy savings compared to commercial hypervisors while complying with service provider's preferences and constraints.

Keywords Energy-efficiency · Virtualization · Multi-criteria decision making · Evolutionary algorithms · Distal learning

1 Introduction

Information Technology (IT) has played a key role in the transformation of modern society by consistently delivering innovative products and services, increasing productivity and supporting economic growth. Nevertheless, IT environmental impact is far from negligible as data centers alone are responsible for 23% of the global ICT industry CO₂ emissions, which in turn are estimated to be as high as the aviation industry [17, 18]. Moreover, data center-related power consumptions continue to grow, and are estimated to increase by 53% between 2013 and 2020 according to [12]. Given so, substantial efforts have been done by the leading IT service providers to maximize the energy efficiency of large-scale data centers in the pursuit of reducing power consumptions and operating costs. However, large-scale infrastructures only represent 5% of the data centers' energy use, with the

Alberto Cocaña-Fernández · Julio Rodríguez-Soares · Luciano Sánchez · José Ranilla
Departamento de Informática, Universidad de Oviedo, Gijón, Spain

Alberto Cocaña-Fernández. E-mail: cocanaalberto@gmail.com
Julio Rodríguez-Soares. E-mail: jrodriguezsoares@gmail.com
Luciano Sánchez. E-mail: luciano@uniovi.es
José Ranilla. E-mail: ranilla@uniovi.es

remaining 95% used by the far less efficient small and medium-scale ones [12]. As a result of this, efforts focused in improving the efficiency of multi-purpose small and medium-scale infrastructures with common workload patterns may yield greater overall power savings. It is noteworthy that certain types of services may only exhibit around 20% of resource usage rates, and underutilized systems can potentially expend a lot of energy needlessly, which can be saved by through virtualization and consolidation technologies [14]. Hence, in this paper we focus on the optimization of multi-purpose virtualized infrastructures by maximizing server consolidation through live virtual machines migrations, and thus reducing overall power consumptions.

The field of virtual machine allocation and server consolidation has been extensively studied by many authors over the last years [3, 27]. Though some authors propose algorithms based on mixed integer nonlinear programming or linear integer programming [5, 19], the majority of solutions are of heuristic nature [4, 16] because virtual machine consolidation can be formulated as a bin-packing problem, which is inherently NP-hard. These solutions, such as [6, 21] seek to reduce energy consumption while adhering to Service Level Agreement. In fact, commercial Virtual Machine Monitors (VMMs) such as Microsoft Hyper-V [29], Citrix XenServer [8] or VMware vSphere [20] already feature built-in heuristic algorithms to improve their energy efficiency through server consolidation. To further improve the results achieved with heuristic algorithms, load forecasting has also been explored [10, 15, 16, 25, 30, 37]. In addition to this, other authors have addressed this problem as a multiobjective one, where energy saving is one objective among others such as service quality, reallocations or the impact on hardware reliability, all of which shall also be optimized to build an effective solution [4, 7, 13, 24, 28, 31, 32, 38]. These last approaches are based on metaheuristics-based optimization, and present a robust solution that accounts explicitly for all factors involved, but lack the simplicity and determinism of simple heuristics. While heuristics have low runtimes, multi-objective allocation algorithms explore larger search spaces where the evolution must be often limited by a maximum number of evaluations. This hinders their application in commercial environments where the VMM must issue optimization decisions on a frequent basis with very low computational overhead, and preferably in a deterministic way.

Given so, and according to the best of our knowledge, it has not been yet presented an holistic and flexible solution that addresses the consolidation problem as a multiobjective learning task. Knowledge-based systems implementing low-complexity, deterministic heuristics should be elicited from workload records, while also leveraging load forecasting to maximize the results achieved. In particular, we propose a proactive multicriteria optimization mechanism based on the framework proposed in [1, 2], whose utility function is a fuzzy rule-based system learned by means of multiobjective genetics-based machine learning (GBML) in a distal supervised learning approach.

The remainder of the paper is as follows. Section 2 explains the architecture of the VDCs that are addresses in this paper. Section 3 explains the mechanism proposed. Section 4 shows the experimental results. Section 5 concludes the paper.

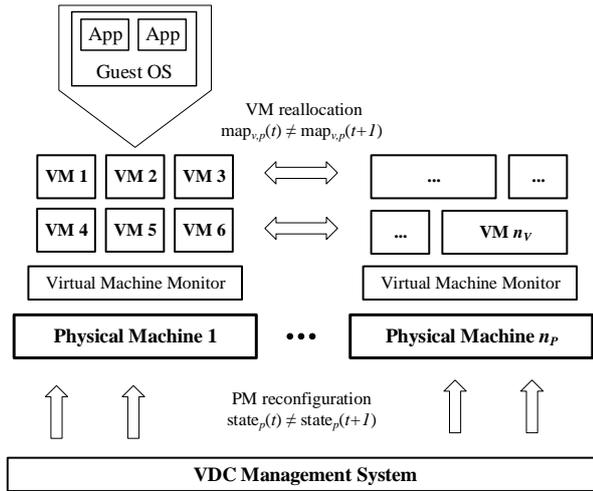


Fig. 1: Virtual Datacenter architecture.

2 System overview

The systems under study in this paper are the Virtualized Data Centers (VDC), as these have become the cornerstone infrastructure in any IT modern service provider, whether they supply applications (Software as a Service), software platforms (Platform as a Service) or infrastructure services (Infrastructure as a Service). The success of this model derives from the extensive use of virtualization technology, which is leveraged to achieve greater levels of hardware utilization and therefore reducing acquisition and operating costs, along with improved application isolation, security, fault tolerance, manageability and flexibility [3, 24, 26, 27]. The basic architecture of a Virtualized Data Center is depicted in Figure 1 and can be structured in three layers: 1) the physical layer, 2) the virtualization layer and 3) the application layer [35].

The physical layer consists of a set of heterogeneous servers whose hardware is the computational foundation of the entire model. These servers are often denoted as *Physical Machines* (PM), and each PM is characterized by its physical resources and its ACPI state¹. The virtualization layer is responsible for the creation, allocation, running and management of Virtual Machines (VMs). The key element is the Virtual Machine Monitor (VMM) which is deployed onto the PMs as an agent to administrate their local resources in order to host the VMs. This layer is controlled from a centralized management server that orchestrates VM allocation and migration, and also PM reconfiguration by issuing commands to change their ACPI state. Finally, the application layer consists of the Operating System (OS) and applications (Apps) encapsulated in each VM.

¹ The Advanced Configuration and Power Interface (ACPI) is an open standard that Operating Systems use, among other things, to manage to perform power management operations over compatible devices. Further details can be found in <http://www.uefi.org/acpi/specs>.

Given a set P with a total of n_P PMs, let each $p \in P$ be characterized by the number of its physical cores ($\text{pcores}(p)$), its RAM memory capacity ($\text{memory}(p)$), and its network capacity ($\text{network}(p)$), having this last one computed as the aggregated bandwidth of its active network adapters. It is also assumed that centralized storage systems are used, so local disk access/write performance does not affect directly its hosting capacity and neither do its storage space. Similarly to PMs, given the set of VMs V , then each $v \in V$ is characterized by both the maximum virtual resources that is entitled to use and its current demand of those resources. In this case, let $\omega_v^{cpu}(t)$ denote the aggregated load of the virtual CPUs (vCPUs) at time t and let be $\omega_v^{mem}(t)$ and $\omega_v^{net}(t)$ its memory and network counterparts. Moreover, we define the allocation of the VM v onto the PM p using the function $\text{map}_{v,p}(t)$, which takes 1 if v is hosted on p at time t , and 0 otherwise.

We can now define four metrics to characterize the behavior of the VDC as a whole over a period of time delimited between t_{ini} and t_{end} , accounting for the Quality of Service (QoS) provided to the users, its power consumption, the amount of VM reallocations, and the impact on hardware reliability. The QoS is measured in terms of physical resources overload, as these situations produce service degradations that are noticeable to the end users. This is measured for the VDC as the average degree of PM overload as seen in Equation 1, having each individual PM overload assessed according to its highest overloaded resource, whether it is CPU, memory or network interfaces, as shown in Equation 2.

$$\text{QoS} = \frac{1}{n_P} \sum_{p \in P} \frac{1}{t_{\text{end}} - t_{\text{ini}}} \int_{t_{\text{ini}}}^{t_{\text{end}}} \text{overload}_p(t) dt. \quad (1)$$

$$\text{overload}_p(t) = \max \{ \text{overload}_p^{cpu}(t), \text{overload}_p^{mem}(t), \text{overload}_p^{net}(t) \}. \quad (2)$$

$$\text{overload}_p^{cpu}(t) = \begin{cases} \text{load}_p^{cpu}(t) - 1 & \text{if } \text{load}_p^{cpu}(t) > 1 \\ 0 & \text{otherwise} \end{cases}.$$

$$\text{load}_p^{cpu}(t) = \sum_{v \in V} \frac{\omega_v^{cpu}(t) \cdot \text{map}_{v,p}(t)}{\text{pcores}(p)}.$$

$$\text{overload}_p^{mem}(t) = \begin{cases} \text{load}_p^{mem}(t) - 1 & \text{if } \text{load}_p^{mem}(t) > 1 \\ 0 & \text{otherwise} \end{cases}.$$

$$\text{load}_p^{mem}(t) = \sum_{v \in V} \frac{\omega_v^{mem}(t) \cdot \text{map}_{v,p}(t)}{\text{memory}(p)}.$$

$$\text{overload}_p^{net}(t) = \begin{cases} \text{load}_p^{net}(t) - 1 & \text{if } \text{load}_p^{net}(t) > 1 \\ 0 & \text{otherwise} \end{cases}.$$

$$\text{load}_p^{net}(t) = \sum_{v \in V} \frac{\omega_v^{net}(t) \cdot \text{map}_{v,p}(t)}{\text{network}(p)}.$$

Let $\text{power}_p(t)$ denote the power consumption of the host p at time t , then the overall VDC power consumption is approximated by adding the power consumption of every powered-on PM:

$$\text{Power consumption} = \sum_{p \in P} \int_{t_{\text{ini}}}^{t_{\text{end}}} \text{power}_p(t) \cdot \text{state}_p(t) dt. \quad (3)$$

The amount of VM reallocations is the number of times that a VM has been migrated from a host to another. Let $\text{nd}_{\text{VM}}(v)$ be the number of discontinuities of the allocation function $\text{map}_{v,p}(t)$ in the time interval $t \in (t_{\text{ini}}, t_{\text{end}})$:

$$\text{VM reallocations} = \sum_{v \in V} \text{nd}_{\text{VM}}(v). \quad (4)$$

PM reconfigurations is the number of times that a PM has changed its power state. Let the function $\text{state}_p(t)$ be the state of host p at time t , taking the value 1 if p is powered-on, and 0 otherwise. Let $\text{nd}_{\text{PM}}(p)$ be the number of discontinuities of the function $\text{state}_p(t)$ in the time interval $t \in (t_{\text{ini}}, t_{\text{end}})$, then:

$$\text{PM reconfigurations} = \sum_{p \in P} \text{nd}_{\text{PM}}(p). \quad (5)$$

Given this, we can state that the VDC optimization problem consists of minimizing power consumptions (Eq. 3) within admissible figures of QoS penalty (Eq. 1), live VM migrations (Eq. 4) and impact in hardware reliability (Eq. 5) for the VDC operator.

3 Proactive multicriteria optimization

Reducing the power consumption of a VDC for any service provider is not a trivial task, since these infrastructures play a key role at their core business. Unsuitable optimization policies may lead to severe impacts on the service quality perceived by the end customers, or produce early hardware degradation that increases maintenance costs and reduces service reliability. According to our experience, any VMM optimization mechanism intended to have a practical application in real world VDCs should be based upon four key principles. First, it should address the optimization problem as a multiobjective one where the goal is not to find the best solution but rather a set of optimal tradeoffs between all the conflicting objectives, thus allowing the VDC operator to select the most suitable option given his or her particular set of preferences and/or constraints. Second, optimization mechanisms that are used in many VDCs are often required to be deterministic and also to require low runtimes and computational costs. Though sophisticated metaheuristics or complex search algorithms could deliver better scheduling solutions, those included in commercial VMMs such as the VMware vSphere, Microsoft Hyper-V or the Citrix XenServer are based on deterministic knowledge-based systems with expert-defined rule bases that are capable of exhibiting conservative behaviors aimed at protecting service quality, while featuring very low runtimes that allow for a more frequent invocation. Third, tuning the optimization mechanism should not be done by hand but rather machine-learned to assure achieving the expected results. Lastly, the mechanism must be robust so that whenever is presented with a new demand scenario, the system will continue to respond as expected according to the preferences that were implicitly modeled. This is better addressed by avoiding reactive approaches that make reallocation or reconfiguration decisions based on the past demand of the VDC, instead of forecasting the expected one and acting proactively in response.

Given so, we propose a proactive VDC optimization mechanism based on the application of predictive techniques for computing systems introduced in [1, 2]. In essence, the VDC is optimized over a future temporal horizon, by issuing the reallocation and reconfiguration decisions that achieve the highest alignment with the preferences/constraints of the VDC operator. This alignment is enforced using a utility function that assess the value of each optimization, and that is elicited from data in a machine learning approach. Temporal horizons are forecast according to a predictive model of the incoming demand built from past records of the VMs load and requests volume in each time slot.

Figure 2 represents the architecture of the proposed mechanism, and its control algorithm is showed in Algorithm 1. Let t denote the beginning of the current control interval and $t + 1$ be its end, let V be the set of VMs and P the set of PMs in the datacenter, let $\underline{d}_v(t-1, k)$ be the k last workload records for the VM v , let ϕ_v be the demand forecasting model built at time t for the VM v , let f be a function that computes the load of VM resources given a certain volume of requests per unit of time, let $\hat{\omega}_v(t+1)$ the forecast resource utilization of the v VM during the current control interval, let J be the utility function, let $\text{map}_{v,p}(t)$ and $\text{state}_p(t)$ be as defined in the preceding section and let $\text{map}'_{v,p}(t)$ be a temporal variable that stores an auxiliary replica of the VM-to-PM mappings to assess potential allocations using the utility function J . Then, the control cycle is as follows:

1. The expected resource load of each VM is estimated for the current control interval using the demand forecasting model ϕ_v . This model is built from the k last workload records of the VM v , and is used to estimate the average volume of concurrent user demands. This can be done using a variety of techniques as proposed by other authors (see [10, 15, 16, 25, 30, 37]). In the experiments done in Section 4, the demand was forecast through the Monte Carlo simulation method using adjusted probability distributions with their parameters values elicited from past demand records. CPU, memory and network loads are then estimated by multiplying the expected demand by a series of coefficients representing the average resource load per concurrent request. Nevertheless, more sophisticated load models based on queuing networks as proposed in [5] can also be used for more accurate results.
2. For each VM v , all candidate PMs $p \in P$ are found where v is allocatable without overloading the host resources.
3. Every candidate mapping (v, p) is scored using the utility function J to determine its alignment with the VDC preferences and constraints. The highest-valued mapping is chosen and the required reallocations are implemented by the VMM.
4. Those PMs without any VM allocated on them are shutdown or put in sleep mode.

3.1 Fuzzy Rule-Based Utility Function

VM reallocations and PM reconfigurations issued by the controller are selected according to the value scored in a utility function that establishes how well these actions align with the predefined preferences and constraints for the VDC operation. The degree of alignment is denoted as “utility”, and ranges between 0 and 1

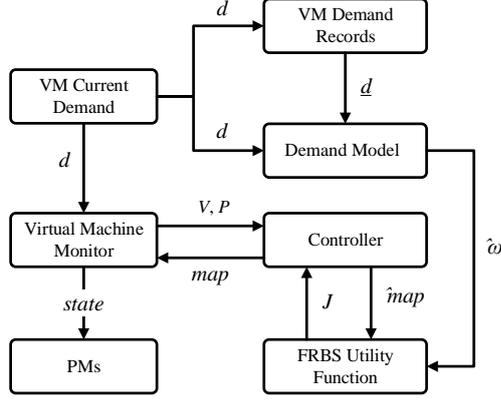


Fig. 2: Predictive controller components overview.

Algorithm 1 Predictive control for an interval beginning at time t

input: $V, P, \text{map}(t), \underline{d}(t-1, k)$ ▷ Forecast future workload for each VM v

- 1: **for each** $v \in V$ **do**
- 2: $\hat{d}_v(t+1) \leftarrow \phi_v(\underline{d}_v(t-1, k))$
- 3: $\hat{\omega}_v^{cpu}(t+1) \leftarrow f_v^{cpu}(\hat{d}_v(t+1))$
- 4: $\hat{\omega}_v^{mem}(t+1) \leftarrow f_v^{mem}(\hat{d}_v(t+1))$
- 5: $\hat{\omega}_v^{net}(t+1) \leftarrow f_v^{net}(\hat{d}_v(t+1))$
- 6: $\hat{\omega}_v(t+1) \leftarrow \{\hat{\omega}_v^{cpu}(t+1), \hat{\omega}_v^{mem}(t+1), \hat{\omega}_v^{net}(t+1)\}$
- 7: **end for each**
- 8: $\text{map}_{v,p}(t+1) \leftarrow 0 \quad \forall v \in V, p \in P$ ▷ Reset all VM mappings to PMs
- 9: **for each** $v \in V$ **do** ▷ Compute new VM allocations
- 10: **for each** $p \in P$ **do**
- 11: **if** v is allocatable in p at time t **then**
- 12: $\text{map}'(t+1) \leftarrow \text{map}(t+1)$
- 13: $\text{map}'_{v,p}(t+1) \leftarrow 1$
- 14: Compute utility $U_{v,p}$ with $J(\text{map}'(t+1), \text{map}(t), \hat{\omega}_v(t+1))$
- 15: **end if**
- 16: **end for each**
- 17: $host \leftarrow \arg \max_p \{U_{v,p}\}$
- 18: $\text{map}_{v,host}(t+1) \leftarrow 1$ ▷ Implement the highest J -valued allocation for v
- 19: **end for each**
- 20: **for each** $p \in P$ **do** ▷ Power off idle PMs
- 21: **if** $\text{map}_{v,p}(t+1) = 0 \quad \forall v \in V$ **then**
- 22: $\text{state}_p(t+1) \leftarrow 0$
- 23: **end if**
- 24: **end for each**

with 0 being the least useful one and 1 the most useful one. The utility function takes as inputs three metrics that numerically summarize the effect of any given allocation of a VM v onto a PM p at time $t+1$: the total load of the destination PM, the accumulated power consumption and the need to migrate the VM v to a different PM:

$$\text{load} = \max\{\text{load}_p^{cpu}(t+1), \text{load}_p^{mem}(t+1), \text{load}_p^{net}(t+1)\}.$$

$$\begin{aligned}
\text{load}_p^{cpu}(t+1) &= \sum_{v \in V} \frac{\omega_v^{cpu}(t+1) \cdot \text{map}'_{v,p}(t+1)}{\text{pcores}(p)}. \\
\text{load}_p^{mem}(t+1) &= \sum_{v \in V} \frac{\omega_v^{mem}(t+1) \cdot \text{map}'_{v,p}(t+1)}{\text{memory}(p)}. \\
\text{load}_p^{net}(t+1) &= \sum_{v \in V} \frac{\omega_v^{net}(t+1) \cdot \text{map}'_{v,p}(t+1)}{\text{network}(p)}. \\
\text{power} &= \sum \text{power}_p(t+1) \quad \forall p \in P \mid \exists v \in V, \text{map}'_{v,p}(t+1) = 1. \\
\text{migration} &= \begin{cases} 1 & \text{if } \text{map}'_{v,p}(t+1) \neq \text{map}_{v,p}(t) \\ 0 & \text{otherwise} \end{cases}.
\end{aligned}$$

The utility function is a fuzzy rule-based system taking the form of a Takagi-Sugeno-Kang (TSK) fuzzy model [22, 34] with Q rules. The structure of this fuzzy model can be expressed as follows:

if load is \tilde{L}_1 and power is \tilde{P}_1 and migration is \tilde{M}_1 **then** value = w_1
if load is \tilde{L}_1 and power is \tilde{P}_1 and migration is \tilde{M}_2 **then** value = w_2
if ... **then** ...
if load is \tilde{L}_1 and power is \tilde{P}_1 and migration is \tilde{M}_{N_3} **then** value = w_{N_3}
if load is \tilde{L}_1 and power is \tilde{P}_2 and migration is \tilde{M}_1 **then** value = w_{N_3+1}
if ... **then** ...
if load is \tilde{L}_1 and power is \tilde{P}_{N_2} and migration is \tilde{M}_{N_3} **then** value = $w_{N_2 \times N_3}$
if load is \tilde{L}_2 and power is \tilde{P}_1 and migration is \tilde{M}_1 **then** value = $w_{N_2 \times N_3 + 1}$
if ... **then** ...
if load is \tilde{L}_{N_1} and power is \tilde{P}_{N_2} and migration is \tilde{M}_{N_3} **then** value = w_Q

where $\tilde{L}_1, \dots, \tilde{L}_{N_1}$, $\tilde{P}_1, \dots, \tilde{P}_{N_2}$ and $\tilde{M}_1, \dots, \tilde{M}_{N_3}$ are fuzzy sets with triangular memberships forming a fuzzy partition [22] of the domain of the variables “load”, “power” and “migration”, respectively, and where w_1, \dots, w_Q are the weights of the Q rules, taking values between 0.0 and 1.0, and representing the utility of the reallocation decision. Note that the partition \tilde{L} has N_1 linguistic terms, \tilde{P} has N_2 and \tilde{M} has N_3 . Then, let the defuzzification function defuzz that computes the output of the TSK model be defined as:

$$\text{defuzz}(\text{load}, \text{power}, \text{migration}) = \frac{\sum_{r=1}^Q \tilde{L}_r(\text{load}) \cdot \tilde{P}_r(\text{power}) \cdot \tilde{M}_r(\text{migration}) \cdot w_r}{\sum_{r=1}^Q \tilde{L}_r(\text{load}) \cdot \tilde{P}_r(\text{power}) \cdot \tilde{M}_r(\text{migration})}. \quad (6)$$

Finally, the utility of each potential VM reallocation can be expressed as:

$$J(\text{map}'(t+1), \text{map}(t), \hat{\omega}_v(t+1)) = \text{defuzz}(\text{load}, \text{power}, \text{migration}) \quad (7)$$

As explained earlier, the overall behaviour of the proactive optimization model relies on the decisions made by the TSK fuzzy system, having this defined by a combination of: $(\tilde{L}_1, \dots, \tilde{L}_{N_1}, \tilde{P}_1, \dots, \tilde{P}_{N_2}, \tilde{M}_1, \dots, \tilde{M}_{N_3}, w_1, \dots, w_Q)$. Since an exhaustive search is unfeasible due to the large search space, and that the objectives that are sought to be minimized are in conflict with each other, multiobjective

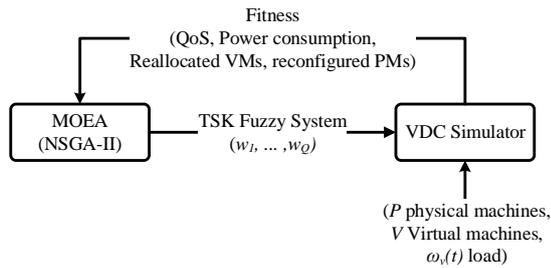


Fig. 3: Distal supervised learning of the proactive model.

evolutionary algorithms (MOEAs) are leveraged to learn the fuzzy model. From the set of non-dominated solutions (Pareto Efficient Frontiers) learned through MOEAs, then the VDC operator can pick the most suitable one according to its tradeoff between the fitness components and to his or her particular set of preferences and/or constraints. Specifically the chosen MOEA is the Non-dominated Sorting Genetic Algorithm II (NSGA-II) [11] and the fitness is composed of the metrics defined in Section 2 for QoS, power consumption, VM migrations and PM reconfiguration (see Eq. 1, 3, 4 and 5, respectively). It is also remarked that uniform partitions are used to avoid tuning the membership functions $\tilde{L}_1, \dots, \tilde{L}_{N_1}$, $\tilde{P}_1, \dots, \tilde{P}_{N_2}$, $\tilde{M}_1, \dots, \tilde{M}_{N_3}$, as this eases the learning process and does not present a fundamental limitation since this can be compensated by altering the rule weights w_1, \dots, w_Q . In other words, the genotype of each individual in the NSGA-II is composed of the weights w_1, \dots, w_Q of the TSK Fuzzy System rule base. Learning is done in a distal supervised learning approach [23], as depicted in Figure 3. In this setup, the *learner* (the NSGA-II algorithm) evolves a population of individuals in order to find the best Pareto Efficient Frontier. Nevertheless, the learner can only control directly the weights of the rule base of each individual (the *proximal* variables), while its fitness (the *distal* variables) is only controlled indirectly by altering these proximal variables. The learner does this by creating new offspring individuals that are tested in a VDC simulation (the *environment*). The outcome of this simulation is used to compute the fitness of the individual which is then returned as feedback to the NSGA-II algorithm. Note that the fitness components are *distal* variables as their value depend on a given VDC scenario characterized by a particular set of VMs, PMs, and workloads.

4 Experimental results

In order to evaluate the proposed solution, experiments were carried out taking as reference ASAC Comunicaciones², an actual service provider of a variety of cloud services including SaaS, DaaS (Desktop as a Service), BaaS (Backup as a Service) and IaaS. This provider was chosen as a reference infrastructure given

² <http://www.asac.as>

that its mid-size TIER-III-certified³ datacenters make an excellent representation of many medium-scale service providers running commercial VMMs and that have room to improve their operating efficiency. The architecture of ASAC datacenters is as explained in Section 2, with PMs grouped in clusters forming a VDC where all VMs and services are isolated to ensure data protection and QoS. An example of the VDCs operated by ASAC is one with 492 VMs running different type of services including mail, web and application servers, domain controllers, DNS servers and databases, among others. This VDC is built over 40 heterogeneous PMs with Intel Xeon E5 and AMD Opteron 6000 CPUs ranging from 12 to 64 total cores and from 196 to 393 GB of memory.

The experimental setup is based on a total of 250 VMs of four different types and 30 PMs. Given that VDCs may exhibit different patterns of activity, five workload scenarios were defined. Briefly explained, Scenario 1 features a static demand where all VMs maintain the same arrival pattern. Scenario 2 shows progressive hourly changes in the arrival patterns so that the load represents that of most corporate services. Scenario 3 sharpens the changes in the arrival patterns. Scenarios 4 and 5 are similar to Scenarios 2 and 3, respectively, but shifting the arrival coefficients a few hours for each VM type so that the demand for each type differ from the others. A detailed specification on how these scenarios are defined and generated is accessible in [9]. In addition to the five scenarios, four arbitrary sets of preferences were defined to resemble the potential inclinations of the VDC operator, and therefore assess the ability of every algorithm to comply with them:

- A. Minimum power use regardless of QoS, VM reallocations and/or PM reconfigurations.
- B. Minimum power use without any noticeable QoS penalty (QoS = 0.0).
- C. Minimum power use with QoS = 0.0 and less or equal to 100 VM reallocations.
- D. Minimum power use with QoS = 0.0 and less or equal to 100 VM reallocations and 10 PM reconfigurations.

A simulator was developed for both learning and testing so that every algorithm can be evaluated in the four criteria of the fitness defined in Section 2. The holdout method was used for validation, randomly sampling three different datasets in each scenario for training, validation and testing. The proactive model described in Section 3 along with the optimization algorithms of three leading VMM commercial solutions [33] were tested under each of the five scenarios using the simulator. In particular the algorithms evaluated are:

1. The Microsoft Hyper-V [29], labeled as “HyperV(*high*, *low*)” where *high* and *low* are the load percentages for the “high” and “low” thresholds.
2. The Citrix XenServer [8], labeled as “XenServer(*mode*, *critical*, *high*, *low*)” where *mode* is the configured mode in XenServer (either “Performance” or “Density”) and *critical*, *high*, *low* are the load percentage in each of the three defined “critical”, “high” and “low” thresholds.
3. The VMware vSphere [20, 36], labeled as “VMware(*target*, *tolerance*)” where *target* is the target resource utilization and *tolerance* is the utilization rate tolerance above and below the target.

³ These certifications are granted by the Uptime institute according to the availability and redundancy of its physical infrastructure. Further information can be found on its website <https://uptimeinstitute.com>

	Scenario 2 test set			
	QoS	Power use (kWh)	VM realloc.	PM reconf.
HyperV(70, 20)	0.00×10^{00}	208.32	58	18
HyperV(80, 30)	0.00×10^{00}	187.59	14	19
HyperV(90, 40)	4.08×10^{-05}	112.94	141	42
HyperV(95, 45)	9.34×10^{-05}	104.13	144	50
XenServer(P, 90, 80, 20)	0.00×10^{00}	192.17	8	7
XenServer(D, 90, 80, 20)	9.90×10^{-05}	126.14	6430	35
XenServer(D, 95, 80, 25)	9.90×10^{-05}	126.14	6430	35
XenServer(D, 99, 80, 30)	9.90×10^{-05}	126.14	6430	35
VMware(50, 20)	0.00×10^{00}	155.18	5853	82
VMware(63, 0)	0.00×10^{00}	155.82	7347	152
VMware(63, 7)	0.00×10^{00}	144.03	7227	157
VMware(63, 18)	0.00×10^{00}	131.80	6781	115
VMware(63, 27)	6.55×10^{-06}	127.46	5403	66
Proactive TSK(A)	2.65×10^{-02}	99.97	3092	69
Proactive TSK(B)	0.00×10^{00}	109.77	9811	64
Proactive TSK(C)	0.00×10^{00}	168.31	44	17
Proactive TSK(D)	0.00×10^{00}	174.25	46	10

Table 1: Experiment results for the test set of the Scenario 2.

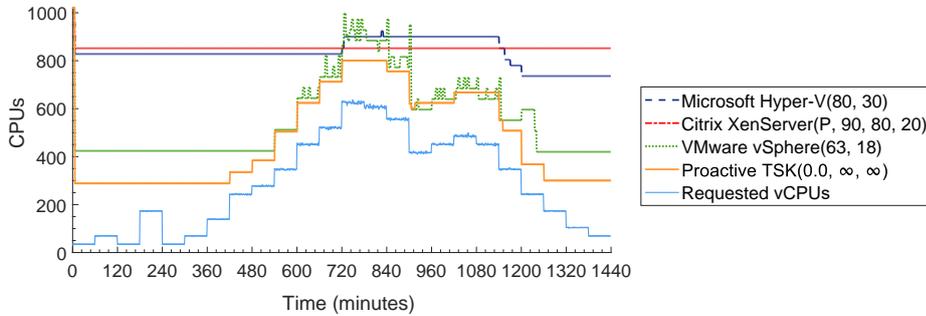


Fig. 4: VDC simulation trace for the test set of the Scenario 2.

4. The proactive model proposed in this paper, labeled as “Proactive TSK(*SoP*)”, where *SoP* is the previously defined set of preferences (A-D) that the chosen TSK individual is tailored to fit, as according to the results obtained during training.

Due to space limitations, only the results for two of the scenarios from the experimental setup are included in this article. The remainder of the results are accessible in [9]. In particular, experimental results for Scenario 2 are shown in Table 1 and Figure 4, and for Scenario 5 in Table 2 and Figure 5. Lastly, Figure 6 shows the set of non-dominated solutions obtained for these two scenarios using for the proactive model proposed in this paper.

These results show that XenServer changes its behavior according to its binary working mode and with little regard to the thresholds configured, achieving scant energy savings and no QoS penalty in *Performance* mode (P) and greater savings with greater QoS penalty in *Density* mode (D). This limited coverage of

	Scenario 5 test set			
	QoS	Power use (kWh)	VM realloc.	PM reconf.
HyperV(70, 20)	0.00×10^{00}	199.79	67	15
HyperV(80, 30)	3.37×10^{-05}	170.27	84	31
HyperV(90, 40)	3.63×10^{-04}	119.51	331	62
HyperV(95, 45)	6.15×10^{-04}	99.81	396	105
XenServer(P, 90, 80, 20)	0.00×10^{00}	192.17	34	7
XenServer(D, 90, 80, 20)	3.14×10^{-04}	127.95	625	48
XenServer(D, 95, 80, 25)	3.14×10^{-04}	127.95	625	48
XenServer(D, 99, 80, 30)	3.14×10^{-04}	127.95	625	48
VMware(50, 20)	9.83×10^{-05}	163.46	11219	281
VMware(63, 0)	6.48×10^{-05}	157.05	12458	377
VMware(63, 7)	2.12×10^{-04}	141.21	11662	373
VMware(63, 18)	4.49×10^{-04}	135.67	11922	290
VMware(63, 27)	8.10×10^{-04}	134.20	9720	219
Proactive TSK(A)	8.52×10^{-02}	96.91	60919	473
Proactive TSK(B)	0.00×10^{00}	157.29	64241	220
Proactive TSK(C)	0.00×10^{00}	166.45	85	22
Proactive TSK(D)	0.00×10^{00}	179.71	10	10

Table 2: Experiment results for the test set of the Scenario 5.

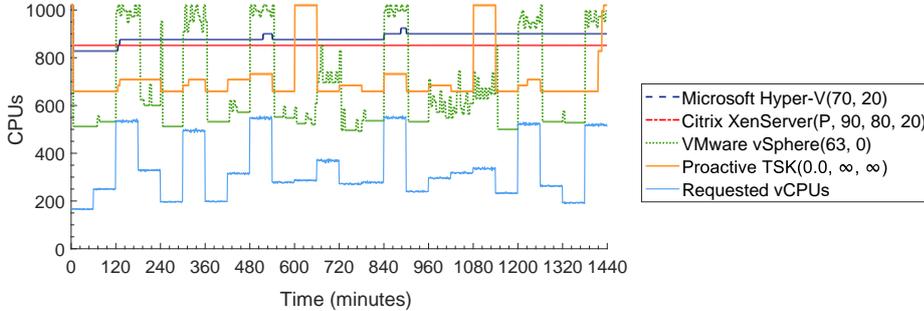


Fig. 5: VDC simulation trace for the test set of the Scenario 5.

the objective space diminishes its **flexibility** to match the preferences of the VDC operator while being competitive compared **to the other VMMs**. For instance, the *Performance* mode does comply with all the evaluated sets of preferences, but its power use is generally higher than with the other commercial VMMs, and *Density* mode often leads to a certain degree of QoS degradation. vSphere features better flexibility as any modification of its configuration parameters alters its behavior. Nevertheless, this **flexibility** translates to a greater density of the objective space in certain areas, but not of a greater overall coverage. This can be noticed by looking at the predominately high number of VM relocations. **This reduces its applicability** to real-world cases when the VDC operator may prefer a conservative approach towards relocations and/or reconfigurations, as can be seen by consistently failing to comply with the sets of preferences C and D in all scenarios. Moreover, the same configurations of vSphere lead to different QoS penalties in different scenarios, thus increasing the risk of an unsuitable behavior on production environments.

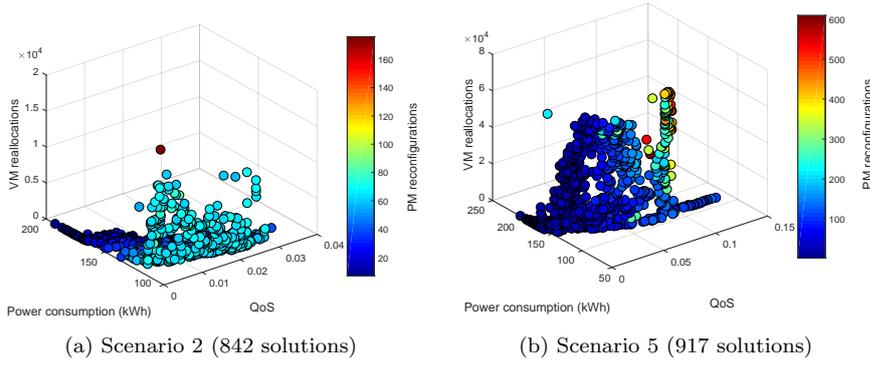


Fig. 6: Pareto Efficient Frontiers of the proactive model obtained in the experiments for the test set of each scenario.

Hyper-V exhibits greater coverage of the objective space in terms of QoS and power use. Finally, the proactive model proposed achieves an overall greater coverage and density of the objective space, while its machine-learned approach diminishes the risk of not complying with the operator preferences when used in live VDCs, as opposed to the aforementioned hand-tuned commercial VMMs. Moreover, given the defined sets of preferences, it overperformed the commercial VMMs in terms of energy savings while strictly **complying to their corresponding** fitness constraints in QoS, reallocations and reconfigurations.

5 Concluding remarks

Reducing the power consumption of VDCs through VM consolidation in any commercial IT service provider is a challenging task given how critical these infrastructures are for its core business. Therefore, feasible optimization mechanisms must be at the same time robust, deterministic and fast, while capable of complying with the operating constraints dictated by the VDC operator. In order to achieve this, a new proactive multicriteria optimization model based on fuzzy rule-based system was proposed. This approach makes reallocation and reconfiguration decisions according to a prediction of the workload and a utility function that assesses the alignment of each decision with the desired behavior for the mechanism. This function is machine-learned through multiobjective optimization under a distal supervised learning setup. Experimental results obtained using as a reference infrastructure that of an actual IT service provider, support the adoption of the proposed mechanism to improve the energy savings of commercial VMMs while strictly complying with the service provider’s preferences and/or constraints.

Acknowledgements

This work has been partially supported by the Ministry of Economy and Competitiveness (“Ministerio de Economía y Competitividad”) from Spain/FEDER un-

der grants TIN2016-81840-REDT, TEC2015-67387-C4-3-R and TIN2014-56967-R, and by the Regional Ministry of the Principality of Asturias under grant FC-15-GRUPIN14-073.

References

1. Abdelwahed, S., Bai, J., Su, R., Kandasamy, N.: On the application of predictive control techniques for adaptive performance management of computing systems. *IEEE Transactions on Network and Service Management* **6**(4), 212–225 (2009). DOI 10.1109/TNSM.2009.04.090402
2. Abdelwahed, S., Kandasamy, N., Neema, S.: A control-based framework for self-managing distributed computing systems. In: *Proceedings of the 1st ACM SIGSOFT workshop on Self-managed systems - WOSS '04*, pp. 3–7. ACM Press, New York, New York, USA (2004). DOI 10.1145/1075405.1075406. URL <http://dl.acm.org/citation.cfm?id=1075405.1075406>
3. Ahmad, R.W., Gani, A., Hamid, S.H.A., Shiraz, M., Yousafzai, A., Xia, F.: A survey on virtual machine migration and server consolidation frameworks for cloud data centers. *Journal of Network and Computer Applications* **52**(C), 11–25 (2015). DOI 10.1016/j.jnca.2015.02.002. URL <http://www.sciencedirect.com/science/article/pii/S1084804515000284>
4. Antonescu, A.F., Robinson, P., Braun, T.: Dynamic SLA management with forecasting using multi-objective optimization. In: *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*, pp. 457–463 (2013)
5. Ardagna, D., Panicucci, B., Trubian, M., Zhang, L.: Energy-Aware Autonomic Resource Allocation in Multitier Virtualized Environments. *IEEE Transactions on Services Computing* **5**(1), 2–19 (2012). DOI 10.1109/TSC.2010.42. URL <http://ieeexplore.ieee.org/document/5560637/>
6. Beloglazov, A., Buyya, R.: 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 (2012). DOI 10.1002/cpe.1867. URL <http://doi.wiley.com/10.1002/cpe.1867>
7. Boxiong Tan, Hui Ma, Yi Mei: A NSGA-II-based approach for service resource allocation in Cloud. In: *2017 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2574–2581. IEEE, San Sebastian, Spain (2017). DOI 10.1109/CEC.2017.7969618. URL <http://ieeexplore.ieee.org/document/7969618/>
8. Citrix Systems: Citrix XenServer Workload Balancing 7.0 Administrator's Guide. Tech. rep. (2016). URL <https://docs.citrix.com/content/dam/docs/en-us/xenserver/xenserver-7-0/downloads/xenserver-7-0-wlb-admin-guide.pdf>
9. Cocaña-Fernández, A., Rodríguez-Soares, J., Sánchez, L., Ranilla, J.: Improving the energy-efficiency of Virtual Data Centers in an IT service provider through proactive fuzzy rules-based multicriteria decision making. Supplementary material. URL <http://pirweb.edv.uniovi.es/JoSEEVDC>
10. Dabbagh, M., Hamdaoui, B., Guizani, M., Rayes, A.: Energy-Efficient Resource Allocation and Provisioning Framework for Cloud Data Centers. *IEEE Transactions on Network and Service Management* **12**(3), 377–391 (2015). DOI 10.1109/TNSM.2015.2436408. URL <http://ieeexplore.ieee.org/document/7111351/>
11. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* **6**(2), 182–197 (2002). DOI 10.1109/4235.996017. URL <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=996017>
12. Delforge, P., Whitney, J.: Issue Paper: Data Center Efficiency Assessment scaling up energy efficiency across the Data Center Industry: evaluating Key Drivers and Barriers. Tech. rep., Natural Resources Defense Council (NRDC) (2014). URL <http://www.nrdc.org/energy/files/data-center-efficiency-assessment-IP.pdf>
13. Deng, W., Liu, F., Jin, H., Liao, X., Liu, H., Chen, L.: Lifetime or energy: Consolidating servers with reliability control in virtualized cloud datacenters. In: *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*, pp. 18–25. IEEE, Taipei, Taiwan (2012). DOI 10.1109/CloudCom.2012.6427550. URL <http://ieeexplore.ieee.org/document/6427550/>

14. Ebbers, M., Archibald, M., França, C.F., Fonseca, D., Griffel, M., Para, V., Searcy, M.: Smarter Data Centers Achieving Greater Efficiency Improve energy efficiency and reduce costs Minimize stranded space, power, and cooling Monitor, manage, and report across both facilities and IT. Tech. rep., IBM (2011). URL <http://www.redbooks.ibm.com/redpapers/pdfs/redp4413.pdf>
15. Farahnakian, F., Liljeberg, P., Plosila, J.: LiRCUP: Linear Regression Based CPU Usage Prediction Algorithm for Live Migration of Virtual Machines in Data Centers. In: 2013 39th Euromicro Conference on Software Engineering and Advanced Applications, pp. 357–364. IEEE, Santander, Spain (2013). DOI 10.1109/SEAA.2013.23. URL <http://ieeexplore.ieee.org/document/6619533/>
16. Farahnakian, F., Pahikkala, T., Liljeberg, P., Plosila, J., Tenhunen, H.: Utilization Prediction Aware VM Consolidation Approach for Green Cloud Computing. In: 2015 IEEE 8th International Conference on Cloud Computing, pp. 381–388. IEEE, New York, NY, USA (2015). DOI 10.1109/CLOUD.2015.58. URL <http://ieeexplore.ieee.org/document/7214068/>
17. Gartner: Gartner Estimates ICT Industry Accounts for 2 Percent of Global CO2 Emissions (2007). URL <http://www.gartner.com/newsroom/id/503867>
18. Gartner: Gartner Says Data Centres Account for 23 Per Cent of Global ICT CO2 Emissions (2007). URL <http://www.gartner.com/newsroom/id/530912>
19. Ghribi, C., Hadji, M., Zeglache, D.: Energy Efficient VM Scheduling for Cloud Data Centers: Exact Allocation and Migration Algorithms. In: 2013 13th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing, pp. 671–678. IEEE (2013). DOI 10.1109/CCGrid.2013.89. URL <http://ieeexplore.ieee.org/document/6546155/>
20. Gulati, A., Shanmuganathan, G., Holler, A., Waldspurger, C., Ji, M., Zhu, X.: VMware Distributed Resource Management: Design, Implementation, and Lessons Learned - VMware Technical Journal (2012). URL <https://labs.vmware.com/vmtj/vmware-distributed-resource-management-design-implementation-and-lessons-learned>
21. Horri, A., Mozafari, M.S., Dastghaibifard, G.: Novel resource allocation algorithms to performance and energy efficiency in cloud computing. *The Journal of Supercomputing* **69**(3), 1445–1461 (2014). DOI 10.1007/s11227-014-1224-8. URL <http://link.springer.com/10.1007/s11227-014-1224-8>
22. Ishibuchi, H., Nakashima, T., Nii, M.: Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining (Advanced Information Processing) (2004). URL <http://dl.acm.org/citation.cfm?id=1044904>
23. Jordan, M., Rumelhart, D.E.: Forward models: Supervised learning with a distal teacher. *Cognitive Science* **16**(3), 307–354 (1992). DOI 10.1016/0364-0213(92)90036-T. URL <http://www.sciencedirect.com/science/article/pii/036402139290036T>
24. Li Xu, Zhibin Zeng, Xiucui Ye: Multi-objective Optimization Based Virtual Resource Allocation Strategy for Cloud Computing. In: 2012 IEEE/ACIS 11th International Conference on Computer and Information Science, pp. 56–61. IEEE, Shanghai, China (2012). DOI 10.1109/ICIS.2012.74. URL <http://ieeexplore.ieee.org/document/6211077/>
25. Liang, Q., Zhang, J., Zhang, Y.h., Liang, J.m.: The placement method of resources and applications based on request prediction in cloud data center. *Information Sciences* **279**, 735–745 (2014). DOI 10.1016/J.INS.2014.04.026. URL <http://www.sciencedirect.com/science/article/pii/S0020025514004733>
26. Mann, Z.Á.: Modeling the virtual machine allocation problem. In: International Conference on Mathematical Methods, Mathematical Models and Simulation in Science and Engineering, pp. 102–106 (2015). URL http://www.cs.bme.hu/~mann/publications/MMSSE-2015/Mannf_}MMSSE{_}2015.pdf
27. Mann, Z.Á., Ádám, Z.: Allocation of Virtual Machines in Cloud Data Centers—A Survey of Problem Models and Optimization Algorithms. *ACM Computing Surveys* **48**(1), 1–34 (2015). DOI 10.1145/2797211. URL <http://dl.acm.org/citation.cfm?doid=2808687.2797211>
28. Mezmaz, M., Melab, N., Kessaci, Y., Lee, Y., Talbi, E.G., Zomaya, A., Tuytens, D.: A parallel bi-objective hybrid metaheuristic for energy-aware scheduling for cloud computing systems. *Journal of Parallel and Distributed Computing* **71**(11), 1497–1508 (2011). DOI 10.1016/j.jpdc.2011.04.007. URL <http://linkinghub.elsevier.com/retrieve/pii/S0743731511000827>
29. Microsoft: Configuring Dynamic Optimization and Power Optimization in VMM (2016). URL [https://technet.microsoft.com/en-us/library/gg675109\(v=sc.12\).aspx](https://technet.microsoft.com/en-us/library/gg675109(v=sc.12).aspx)

30. Nagpure, M.B., Dahiwalé, P., Marbate, P.: An efficient dynamic resource allocation strategy for VM environment in cloud. In: 2015 International Conference on Pervasive Computing (ICPC), pp. 1–5. IEEE, Pune, India (2015). DOI 10.1109/PERVASIVE.2015.7087186. URL <http://ieeexplore.ieee.org/document/7087186/>
31. Pires, F.L., Baran, B.: Multi-objective Virtual Machine Placement with Service Level Agreement: A Memetic Algorithm Approach. In: 2013 IEEE/ACM 6th International Conference on Utility and Cloud Computing, pp. 203–210. IEEE (2013). DOI 10.1109/UCC.2013.44. URL <http://ieeexplore.ieee.org/document/6809358/>
32. Ramezani, F., Naderpour, M., Lu, J.: A multi-objective optimization model for virtual machine mapping in cloud data centres. In: 2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), pp. 1259–1265. IEEE, Vancouver, BC, Canada (2016). DOI 10.1109/FUZZ-IEEE.2016.7737833. URL <http://ieeexplore.ieee.org/document/7737833/>
33. T. J. Bittman, Dawson, P., Warrilow, M.: Magic Quadrant for x86 Server Virtualization Infrastructure. URL <https://www.gartner.com/doc/3400418/magic-quadrant-x-server-virtualization>
34. Takagi, T., Sugeno, M.: Fuzzy identification of systems and its applications to modeling and control. *IEEE Transactions on Systems, Man, and Cybernetics* **SMC-15**(1), 116–132 (1985). DOI 10.1109/TSMC.1985.6313399. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6313399>
35. Vasudevan, M., Tian, Y.C., Tang, M., Kozan, E.: Profile-based application assignment for greener and more energy-efficient data centers. *Future Generation Computer Systems* **67**, 94–108 (2017). DOI 10.1016/J.FUTURE.2016.06.037. URL <http://www.sciencedirect.com/science/article/pii/S0167739X16302606>
36. VMware: VMware Distributed Power Management Concepts and Use. Tech. rep., VMware (2010). URL <https://www.vmware.com/content/dam/digitalmarketing/vmware/en/pdf/techpaper/Distributed-Power-Management-vSphere.pdf>
37. Xiao, Z., Song, W., Chen, Q.: Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment. *IEEE Transactions on Parallel and Distributed Systems* **24**(6), 1107–1117 (2013). DOI 10.1109/TPDS.2012.283. URL <http://ieeexplore.ieee.org/document/6311403/>
38. Yu, J., Kirley, M., Buyya, R.: Multi-objective planning for workflow execution on Grids. In: 2007 8th IEEE/ACM International Conference on Grid Computing, pp. 10–17. IEEE (2007). DOI 10.1109/GRID.2007.4354110. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4354110>