

A Pliant-based Virtual Machine Scheduling Solution to Improve the Energy Efficiency of IaaS Clouds

A. Kertesz · J. D. Dombi · A. Benyi

Received: date / Accepted: date

Abstract Recently cloud computing is facing increasing attention as it is applied in many business scenarios by advertising the illusion of infinite resources towards its customers. Nevertheless, it raises severe issues with energy consumption: the higher levels of quality and availability require irrational energy expenditures. This paper proposes a Pliant system-based virtual machine scheduling approaches for reducing the energy consumption of cloud datacenters. We have designed a CloudSim-based simulation environment for task-based cloud applications to evaluate our proposed solution, and applied industrial workload traces for our experiments. We show that significant savings can be achieved in energy consumption by our proposed Pliant-based algorithms, in this way a beneficial trade-off can be reached by IaaS providers between energy consumption and execution time.

Keywords Cloud Computing · Energy Awareness · VM scheduling · Pliant system · Simulation

1 Introduction

Cloud computing incorporates many aspects of sharing software and hardware solutions, including computing and storage resources, application runtimes or complex application functionalities. The cloud paradigm

changed the way people look at computing infrastructures. First, one does not need to be expert in infrastructure administration, operation and maintenance even if large scale systems are utilized. Second, the elasticity of Infrastructure as a Service clouds allows these systems to better follow the users' actual demands. However, there is also an adversary effect: the virtualized nature of these systems detaches users from several operational issues like energy efficient usage, that has been addressed previously in the context of parallel and distributed systems, and largely remains unnoticed [?,?].

The cloud computing technology made a qualitative breakthrough as it is present in many consumer appliances including various mobile devices. They advertise the illusion of infinite resources towards the consumers, meanwhile it also raises severe issues with energy consumption: the higher levels of quality and availability require irrational energy expenditures, according to some experts the consumed energy of resources spent for idling represent a considerable amount [?]. Current trends are claimed to be clearly unsustainable with respect to resource utilisation, CO₂ footprint and overall energy efficiency. It is anticipated that further growth is limited by energy consumption, furthermore competitiveness of companies are and will be strongly tied to these issues.

As cloud services become more and more popular, small- and medium-sized cloud service providers will soon face increasing user demands that cannot be met with their current infrastructures. These user demands range from occasional needs for extreme amount of resources (compared to the provider's current infrastructure) to the need for multi-site virtual machine (VM) deployment options that enable enhanced services such as disaster recovery. Thus these providers need to increase the size of their infrastructure by introducing

A. Kertesz
MTA SZTAKI,
H-1518 Budapest, P.O. Box 63, Hungary
E-mail: kertesz.attila@sztaki.mta.hu
J. D. Dombi, A. Benyi
University of Szeged,
6720 Szeged, Dugonics ter 13, Hungary
E-mail: dombijd@inf.u-szeged.hu, benyi.attila@stud.u-szeged.hu

multiple datacenters covering various locations, and offering unprecedented amount of resources. Current IaaS solutions provide the opportunity for service providers to satisfy these needs by focusing their attention to non-technical issues like the increased operating cost of their datacenters. Despite energy consumption is a major component of these operating costs, current IaaS solutions barely handle the infrastructure with energy aware solutions. Therefore providers were restricted to reduce their consumption on the hardware level so far, independently from the applied IaaS solution. Energy costs are also increasing, and datacenter equipment is stressing power and cooling infrastructures, thus the main issue is not the current amount of data center emissions but the fact that these emissions are raising faster than any other carbon emission [?]. Although these improvements in hardware are crucial, we believe that the energy consumption could also be significantly reduced with software in over-provisioned IaaS systems. Over-provisioning is a key behaviour at smaller sized providers, who offer services for users with occasional peaks in resource demands.

Reducing the carbon footprint of European countries is also a must and expected by the European Commission, as well as to increase the number and size of European cloud providers [?]. By federating these providers, more competitive initiatives can be founded, that can be sophisticatedly managed to meet these expectations [?]. Cloud federation refers to a mesh of cloud providers that are interconnected based on open standards to provide a universal decentralized computing environment where everything is driven by constraints and agreements in a ubiquitous, multi-provider infrastructure. The general goal of the management layer in a cloud federation is to distribute load among the participating cloud providers, to enhance user satisfaction by filtering out underperforming providers, and schedule and execute service calls with minimized energy consumption within the selected IaaS system. Concerning related solutions both hierarchical and horizontal federation types are used, and heterogeneity within the participating providers is mostly present in hierarchical solutions. We have already proposed an architecture called Federated Cloud Management (FCM [?]) belonging to the hierarchical category, where a holistic approach with a two-level brokering solution is used: a meta-brokering component is used to direct service calls to providers, and then a cloud-brokering component to map these calls onto an optimized number of virtual machines.

In this paper we target the later, cloud-brokering layer, and we focus on the energy-aware management of datacenters of single cloud providers specialized for pro-

visioning task-based cloud applications. In order to enable experimentation in this field, we have developed a CloudSim-based simulation environment. To cope with the high uncertainty and unpredictable load present in these heterogeneous, virtualized large-scale systems, we apply Pliant system-based approaches [?] to the management of these systems, which is similar to a fuzzy system [?]. The difference between the Pliant system and a fuzzy system lies in the choice of operators. The Pliant system can be applied to a wide variety of real world problems. It is possible to use it as a dynamic system, we can create a system like the Fuzzy Cognitive Map. We can apply the Pliant system to problems by introducing function approximation techniques, which have useful and practical aspects. We can also apply it in problems that need decision-making techniques.

The main contributions of this paper are: (i) the development of a cloud simulation environment for task-based cloud applications, (ii) the design of energy-aware Pliant-based virtual machine scheduling algorithms for IaaS cloud management, and (iii) the evaluation of the proposed algorithms in the extended simulation environment with real-world traces.

The remainder of this paper is as follows: Section ?? presents the related VM management approaches in datacenters; Section ?? introduces our extended simulation architecture; Section ?? introduces the advanced scheduling algorithms using the Pliant method for VM scheduling; and Section ?? describes the evaluation methodology and the simulation results. Finally, Section ?? summarizes the main contributions of the paper.

2 Related work

Regarding energy efficiency in a single cloud, Cioara et al. [?] introduced an energy aware scheduling policy to consolidate power management by using reinforcement learning techniques to restore a service center to an energy efficient state. Feller et al. proposed a dynamic cluster manager called Snooze [?], which is able to dynamically consolidate the workload of a heterogeneous large-scale cluster composed of resources using virtualization. In a later work [?], they use power meters to monitor energy usage of cloud resources, and estimate the resource usage of VMs. Their mechanisms address VM placement, relocation and migration by keeping VMs on as few nodes as possible. Also, IBM has proposed pMapper [?], which is a power-aware application placement controller in the context of an environment with heterogeneous virtualized server clusters. The placement component of the application management middleware takes into account the power and migration costs in addition to the performance benefit

while placing the application containers on the physical servers.

Cardosa et al. [?] presented a novel suite of techniques for placement and power consolidation of VMs in datacentres taking advantage of the min-max and shares features inherent in virtualization technologies, like VMware and Xen. These features allow to specify the minimum and maximum amount of resources that can be allocated to a VM, and provide a shares based mechanism for the hypervisor to distribute spare resources among contending VMs. Lee et al. [?] discuss service request scheduling in clouds based on achievable profits. They propose a pricing model using processor sharing for composite services in clouds. Berral et al. [?] present a framework to address energy efficiency using an intelligent consolidation methodology, which applies various techniques such as machine learning, power-aware consolidation algorithms, and turning on/off machines. Their work applies machine learning techniques on scheduling algorithms to improve server workload predictions thus achieving a better job consolidation in order to turn off spare servers and thereby saving energy in a datacenter.

J. L. Lucas-Simarro et al. [?] proposed different scheduling strategies for optimal deployment of services across multiple clouds based on various optimization criteria. The examined scheduling policies include budget, performance, load balancing and other dynamic conditions, but they neglected energy efficiency, which is the aim of our work.

Regarding fuzzy approaches, Salleh et al. [?] have shown how to set up and use fuzzy logic in a traditional way for dynamic task scheduling in multiprocessor systems. We have already published a paper [?] on applying the Pliant approach to job scheduling in Grids. In this current paper we would like to show that it is also possible to use Pliant system for scheduling, with only a few rules. The novelty of this contribution lies in the way we apply the Pliant system to clouds: the way we select cloud-specific properties as parameters of the Pliant system.

Concerning cloud simulations, Berge et al. [?] have designed a simulator called SVD within the CoolEmAll project for investigating energy consumption in datacenters. It is an extended version of the GSSIM simulator, and they are planning to support application modeling and profiling through benchmarks. Regarding federation-wide simulations, Sotiriadis et al. [?] investigated inter-cloud simulations by developing the SimIC simulation toolkit that is able to mimic the inter-cloud service formation to enable the investigation of service-oriented cloud utilization, but they also neglect energy efficiency.

3 Simulation of clouds

We have used the CloudSim simulator [?] to develop a simulation environment for our research, since it is a widely accepted, used and referred solution. Beloglazov and Buyya [?] have already started to examine how energy efficiency could be investigated within this simulator. Datacenters consume huge amounts of energy resulting in high operating costs and increased carbon dioxide emissions. The dynamic consolidation of VMs using live migration and switching off idle nodes can be used to optimize resource usage and reduce energy consumption, but they argue that aggressive consolidation may lead to performance degradation. They proposed adaptive heuristics for dynamic consolidation of VMs based on an analysis of historical data from the resource usage by VMs, while ensuring a high level of adherence to the Service Level Agreements (SLA). They used PlanetLab trace files [?] workload logs to simulate load changes of continuously running services in VMs. These traces contain records of each VM's periodic utilization, thus the simulation assumes each VM is going to process only one task (called as cloudlet in CloudSim) at a time as a service.

In this work our goal was to investigate task-based (HPC/HTC) cloud applications executed by a single cloud provider possibly having more than one datacenter. Since CloudSim is tailored to the evaluation of continuously running web-based applications [?], we decided to extend this simulation environment to suite our needs.

Our approach is slightly different to the one used by the original version of CloudSim, as we tried sending cloudlets with varying parameters, such as start time and length at random intervals. For that purpose we used the log files provided by Prezi Inc. [?] (discussed in detail in Section ??). These log files contain detailed data on each cloudlet received, such as its start time, length and queue type. To adapt CloudSim to the new features, several changes had to be made. One of the crucial changes was in the *CloudletScheduler* component, so each VM could handle multiple cloudlets at the same time. As long as the VM's utilization is below 100%, it can process new cloudlets, and once a VM reaches its full utilization, further cloudlets get queued. Once a VM has no cloudlets left to process, it is shut down, and if a host has no remaining VMs, it is shut down as well. Each host's power consumption is based on a power model, which is based on a benchmark result provided by SPEC [?]. We used 5 different power models to make the difference between varying algorithms more obvious. Each datacenter sums up the power consumed by its hosts for every timeframe a cloudlet is

Listing 1 Pseudo code of the default OptUtil algorithm

```

lowestVm = first VM with the same queue type
           as the cloudlet;
FOREACH (vmlist as vm)
  IF (vm.utilization() < lowestVm.
      utilization())
    AND vm.queueType == lowestVm.
      queueType)
      lowestVm=vm;
IF (lowestVm.utilization > 100)
  IF (try to create a new vm)
    lowestVm = new vm;
cloudlet.setVm = lowestVm;

```

being processed, giving us a close approximation of the amount of power and time needed to complete all the requested cloudlets. For each cloudlet a VM is chosen by our default VM scheduling algorithm called 'OptUtil' shown in Listing ???. The hosts (physical machines) created during the simulations differ in their characteristics, altogether 5 types of hosts were used. However, while there are different hosts, only one type of VM was used in all simulations.

In case the utilization of all VMs is over 100%, the algorithm will try to create a new one, thus ensuring the lowest process time. For each new VM the host is chosen based on its power model, and we are assuming that every host will be fully utilized, so the host with the lowest power consumption on 100% utilization will be submitted, ensuring the lowest power consumption. In the following section we discuss the Pliant-based VM scheduling solution.

4 Pliant scheduling approach

Fuzzy sets were introduced by Lofti Zadeh in 1965 with the aim of reconciling mathematical modeling and human knowledge in the engineering sciences. Most of the building blocks of the theory of fuzzy sets were proposed by him, especially fuzzy extensions of classical basic mathematical notions like logical connectives, rules, relations and quantifiers.

Over the last century, fuzzy sets and fuzzy logic have become more popular areas for research, and they are being applied in fields such as computer science, mathematics and engineering. This has led to a truly enormous literature, where there are presently over thirty thousand published papers dealing with fuzzy logic, and several hundreds books have appeared on the various facets of the theory and the methodology. However, there is not a single, superior fuzzy logic or fuzzy rea-

soning method available, although there are numerous competing theories.

The Pliant system is a kind of fuzzy theory that is similar to a fuzzy system [?]. The difference between the two systems lies in the choice of operators. In fuzzy theory the membership function plays an important role, but the exact definition of this function is often unclear. In Pliant systems we use a so-called distending function, which represents a soft inequality. In the Pliant system the various operators, which are called the conjunction, disjunction and aggregative operators, are closely related to each other. We usually have a generator function and using this function we can create aggregation operator, conjunctive operator or disjunctive operator. In the Pliant Systems the corresponding aggregative operators of the strict t-norm and strict t-conorm are equivalent, and DeMorgans law is obeyed with the corresponding strong negation of the strict t-norm or t-conorm.

The Pliant system has a strict, monotonously increasing t-norm and t-conorm, and the following expression is valid for the generator function:

$$f_c(x)f_d(x) = 1, \quad (1)$$

where $f_c(x)$ and $f_d(x)$ are the generator functions for the conjunctive and disjunctive logical operators, respectively. This system is defined in the $[0,1]$ interval. The operators of the Pliant system are

$$c(\mathbf{x}) = \frac{1}{1 + \left(\sum_{i=1}^n w_i \left(\frac{1-x_i}{x_i} \right)^\alpha \right)^{1/\alpha}} \quad (2)$$

$$d(\mathbf{x}) = \frac{1}{1 + \left(\sum_{i=1}^n w_i \left(\frac{1-x_i}{x_i} \right)^{-\alpha} \right)^{-1/\alpha}} \quad (3)$$

$$a_{\nu_*}(\mathbf{x}) = \frac{1}{1 + \left(\frac{1-\nu_*}{\nu_*} \right) \prod_{i=1}^n \left(\frac{1-x_i}{x_i} \frac{1-\nu_*}{\nu_*} \right)^{w_i}} \quad (4)$$

$$n(x) = \frac{1}{1 + \left(\frac{1-\nu_*}{\nu_*} \right)^2 \frac{x}{1-x}}, \quad (5)$$

$$\kappa_\nu^{(\lambda)}(x) = \frac{1}{1 + \frac{1-\nu_0}{\nu_0} \left(\frac{\nu}{1-\nu} \frac{1-x}{x} \right)^\lambda}$$

where $\nu_* \in]0, 1[$, with generator functions

$$f_c(x) = \left(\frac{1-x}{x} \right)^\alpha \quad f_d(x) = \left(\frac{1-x}{x} \right)^{-\alpha}, \quad (6)$$

where $\alpha > 0$.

The operators c , d and n fulfill the DeMorgan identity for all ν , a and n fulfill the self-DeMorgan identity for all ν , and the aggregative operator is distributive with the strict t-norm or t-conorm. The ν value express the expected value of the given context. This means that if the given x value is greater than ν , then the operators increase the value of x . The opposite is true when x is smaller than ν . Later we will define the value of ν in the virtual machine environment.

In fuzzy concepts the most powerful term is the membership function. In the Pliant concept this function is connected to the operator system. We can also introduce the distending function and the notation of the distending function is

$$\delta(x) = \text{truth}(0 < x) \quad x \in R$$

We can generalize this in the following way

$$\delta(g(\mathbf{x})) = \text{truth}(0 < g(\mathbf{x})) \quad \mathbf{x} \in R^n$$

Instead of a strict relation, we can define a function which provides information on the validity of the relation.

In fuzzy logic theory, the membership function has a different interpretation. In the Pliant concept, the membership function is replaced by a soft interval. Its mathematical description is

$$\delta_{a,b}^{\lambda_1, \lambda_2}(x) = \text{truth}(a <_{\lambda_1} x <_{\lambda_2} b)$$

Definition 1 In a pliant system if the initial conditions are

$$\delta_{a,b}^{\lambda_1, \lambda_2}(a) = \nu_0 \quad \delta_{a,b}^{\lambda_1, \lambda_2}(b) = \nu_0, \quad (7)$$

then the distending interval is

$$\delta_{a,b}^{\lambda_1, \lambda_2}(x) = f^{-1} \left(\frac{1}{A} \left(A_1 e^{-\lambda_1(x-a)} + A_2 e^{-\lambda_2(b-x)} \right) \right), \quad (8)$$

where

$$\begin{aligned} A &= \frac{1}{f(\nu_0)} \left(1 - e^{-(\lambda_1 + \lambda_2)(b-a)} \right) \\ A_1 &= 1 - e^{-\lambda_2(b-a)} \\ A_2 &= 1 - e^{-\lambda_1(b-a)} \end{aligned} \quad (9)$$

In our earlier work [?], we developed a meta-brokering component that uses the Pliant system to select a good performing Grid broker for a user's job even under conditions of high uncertainty. In this paper we address energy consumption of VMs in clouds, therefore we created scheduling algorithms in order to handle the energy aware management case with a similar approach.

These algorithms calculate a score for each cloudlet using the cloud's properties. The calculation step includes a normalization step, where we apply a special Sigmoid function. In the normalization step it should be mentioned that if the normalized value is close to one, it means it is a more valuable property, and if the normalized value is close to zero, it means it is a less prioritized property. For example, if the counter of power consumption is high, the normalization algorithm should give a value close to zero.

One of these algorithms considers time and the other considers energy for optimization. There are hosts in the simulated datacenters, and each host can run several VMs. This environment can be described with the same three properties, namely a power usage counter (PUC), the power consumption counter (PCC) and the number of processors (PROC):

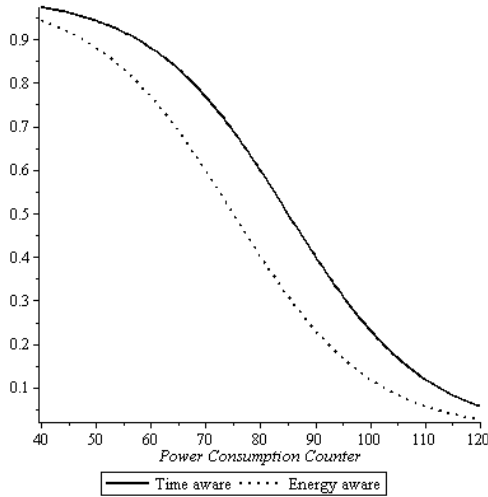
- The power usage counter gives performance of the CPU usages of the given simulation time. The value can be larger than 100, which means that there are more cloudlets in the VM's queue.
- The power consumption counter gives the energy usage of the given host at a given time. The value is generally between 40 and 120 W, but it depends on the actual physical processor.
- The number of processors gives the available number of processors of a host.

We have developed Pliant decision making algorithms that take into account the above-mentioned properties and decide to which VM a cloudlet should be submitted: one optimizes cloudlet executions for time, and the other one for energy. We use different normalization for these two strategies. First we start with a normalization step and we apply different kinds of Sigmoid functions to normalize the environment's property value. We examine the environment's variable and check the interval of the given properties. The interval determine the α parameter of the Sigmoid function. The λ parameter expresses our knowledge of the system environment. Using this information we define the value of the Sigmoid's parameter. In Figure ??, ?? and ?? we show three different kinds of normalization functions we use.

Table ?? shows the predefined values of the parameters of the normalization functions.

Table 1 Parameters of the Sigmoid function

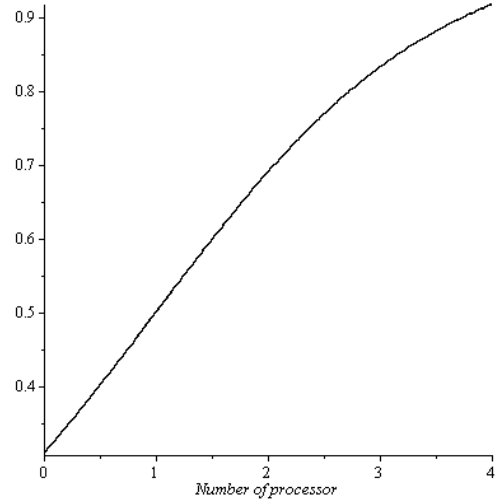
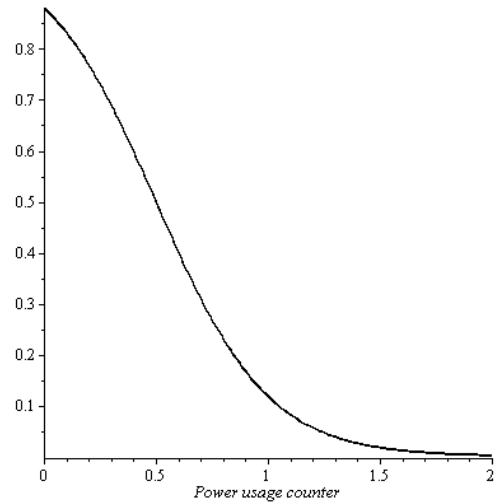
Property	Time		Energy	
Property	Alpha	Lambda	Alpha	Lambda
PUC	0.5	-4.0	0.5	-4.0
PCC	85.0	-0.08	75.0	-0.08
PROC	1.0	0.8	1.0	0.8

**Fig. 1** Utilized normalized function for the power consumption (PCC)

In the simulation environment (to be discussed later in Section ??) every host has 4 processors, so after the normalization the normalized property value is the same for all instances. We would like to emphasize that it is better if we use less power, therefore we created two different parameter sets: one for time-aware and one for energy-aware scheduling. As we can see in Figure ??, the minimum energy usage in this environment is around 40 and the maximum is around 120 W. In Figure ?? we can see that if the number of power consumption is increasing then the value of the normalized function is decreasing. The opposite is true for the number of processors (in Figure ??).

We should also emphasize that the closer the value to one, the better the property is, and if the value is close to zero, it means that the property is not so good. For example if the power usage is high, the normalization algorithm should give a value close to zero, because it is not a good thing if the cloudlet uses a lot of power. The opposite is true for the processor number.

After the normalization step we modify the normalized value to emphasize the importance of the result. This means that if the given x value is greater than our expectation (ν) than we will increase the value of x . the opposite is true when the given x is smaller than ν . To achieve this we will modify the normalized value

**Fig. 2** Utilized normalized function for the processor number (PROC)**Fig. 3** Utilized normalized function for the power usage (PUC)

by using the Kappa function shown in Figure ?? with $\nu = 0.4$ and $\lambda = 3.0$ parameters:

$$\kappa_{\nu}^{\lambda}(x) = \frac{1}{1 + \left(\frac{\nu}{1-\nu} \frac{1-x}{x}\right)^{\lambda}} \quad (10)$$

Finally to calculate a VM's score number for the given cloudlet, we use the aggregation operator:

$$a_{\nu, \nu_0}(x_1, \dots, x_n) = \frac{1}{1 + \frac{1-\nu_0}{\nu_0} \frac{\nu}{1-\nu} \prod_{i=1}^n \frac{1-x_i}{x_i}}, \quad (11)$$

where ν is the neutral value and ν_0 is the threshold value of the corresponding negation. Here we don't want to threshold the result so both parameters have the

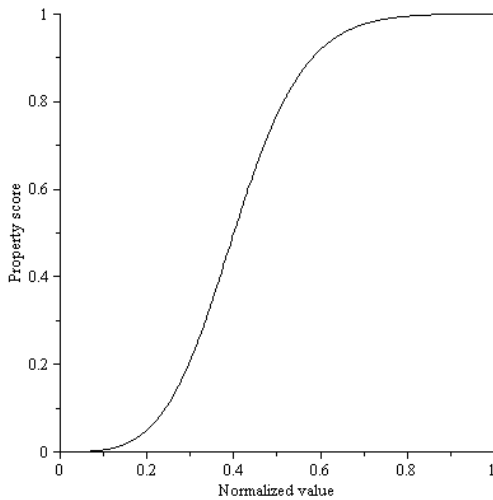


Fig. 4 The kappa function

same value 0.5. The result of the calculation is always a real number that lies in the $[0,1]$ interval. So we calculate the score for all VM to find which VM is the most suitable for our strategy. If the best score value is very low (the value depends on the strategy), then we try to create a new VM.

5 Evaluation

In order to investigate the energy consumption of cloud providers in our extended simulation environment, we have used real-world trace files of an international company called Prezi Inc, who offers a presentation editing service, which is available on multiple platforms, therefore they have to convert some of their created media files to other formats before they can display them on all devices. In April 2013, they launched a competition titled "Scale Contest" [?] for university students to test their knowledge of control and queueing theories on real-life problems. Their conversion processes are carried out on virtual machines: at peak times, they need to launch more instances of these VMs, but over the weekend they can stop most of them. This campaign was initiated in order to find a suitable algorithm that launches the exact number of VMs for a given workload. They published log files on their website containing workload traces for two weeks of utilization, which serves as a basis for algorithmic experimentations.

They operate three queues in their cloud system for the jobs participating in the conversion processes:

- export: contains jobs which result in downloadable zipped prezi files.
- url: these jobs download an image from a URL and insert them into a prezi file.

- general: all other conversion jobs (audio, video, pdf, ppt, etc).

The lines of the published workload traces have the following format:

```
"2012-12-14 21:35:12 237 general 9.134963"
```

This means that at the given time, a job enters the *general* queue with the id 237, and the job will take 9.134963 seconds to run. These logs had to be used as input by the competitors. They contain three weeks of actual data accumulated by Prezis conversion system, and the first two weeks of logs are publicly available. They planned to use unpublished logs from the third and fourth week to evaluate your submissions to the competition. The available trace files having two weeks of utilization contain more than 2000000 lines, and their submitted (and processed) jobs highly varies over the 14 days. Table ?? shows the exact number of jobs per day.

In the next subsections we detail our measurements. In the first one we perform a preliminary evaluation with a greedy strategy without using fuzzy methods. In the second subsection we compare two extreme algorithms (using minimum and maximum number of VMs) with a smartly randomized VM selection to determine the possible ranges of execution time and energy consumption values. Finally in the third one, we present the evaluation of our proposed Pliant-based algorithms.

Table 2 Jobs in the Perzi trace files

Days	1st week	2nd week
1	269344	253354
2	122438	116356
3	170136	158945
4	317481	302715
5	332769	331816
6	339371	328555
7	330854	318323

Table 3 Evaluation results for RoundRobin

Hosts	Cloud-lets	VMs	Energy (kWh)	Time (sec)
100	10000	1<	63.20	25200
	50000		104.66	39000
500	100000		143.62	48600
			381.37	70200

5.1 Preliminary evaluation

For a preliminary evaluation phase we used the trace file of the first week. We have performed experiments

Table 4 Evaluation results for OPTUTIL

Hosts	Cloud-lets	VMs	Energy (kWh)	Time (sec)
100	10000	1<	18.90	7500
	50000		87.12	32400
500	100000		90.41	7200
	100000		197.26	15000

with datacenters having 100 to 500 hosts, and submitted 10000 to 100000 jobs (i.e. cloudlets) from the log. By default we used a round robin strategy to schedule the logs to the available VMs (1 at the beginning), and if no more available VM was present in the system (that could execute the job without any delay) at a given time, we have deployed another one continuously. The results of this evaluation can be seen in Table ???. We have also executed similar simulations by applying our proposed optimized utilization strategy called "OPTUTIL", that deploys another VM, if the available ones are at least 80% loaded. The results of this second evaluation can be seen in Table ??.

From this preliminary evaluation we can see that our proposed algorithm performed better than the round robin, both in energy consumption and execution time.

5.2 First evaluation round

To develop Pliant-based algorithms, in the first round of experiments we created three initial strategies: the first one uses only one VM to execute all submitted jobs (referred to as MINIMUM), the second deploys a new VM for all jobs (MAXIMUM), and the third uses randomized VM selection from the available VMs (smartly prioritizing the less loaded ones), and deploys a new one, if no free VM is found (SMARTRANDOM). Tables ??, ?? and ?? summarize the results of evaluating these algorithms. From these results we can see that utilizing the lowest number of VMs results in the lowest energy consumption, but of course on the expense of the execution time, which is the highest in this case.

Table 5 Evaluation results for MAXIMUM

Hosts	Cloud-lets	VMs	Energy (kWh)	Time (sec)
100	1000	241	7.64	759
	10000		76.35	4088
	50000		365.35	14220
	100000		934.22	39224

Based on the results of these artificial strategies we have created a Pliant-based strategy (referred to as PLIANTDEFAULT), first focusing on execution time

Table 6 Evaluation results for MINIMUM

Hosts	Cloud-lets	VMs	Energy (kWh)	Time (sec)
100	1000	3	0.19	8179
	10000		1.91	81008
	50000		6.54	240940
	100000		13.87	461724

Table 7 Evaluation results for SMARTRANDOM

Hosts	Cloud-lets	VMs	Energy (kWh)	Time (sec)
100	1000	3	0.20	8619
	10000		1.53	60298
	50000		5.77	198060
	100000		12.50	386074

reduction with some energy savings. Concerning this default algorithm Table ?? shows the results of the simulation. This table shows that this strategy could achieved significant performance gains in terms of execution time as expected, but it also had much higher energy consumption than the MINIMUM and SMARTRANDOM initial strategy.

Table 8 Evaluation results for PLIANTDEFAULT

Hosts	Cloud-lets	VMs	Energy (kWh)	Time (sec)
100	1000	14	0.26	749
	10000	16	2.87	3768
	50000	24	17.26	14240
	100000	25	53.21	39304

Table 9 Evaluation results for PLIANTTIME

Hosts	Cloud-lets	VMs	Energy (kWh)	Time (sec)
100	1000	13	0.21	629
	10000	16	2.77	4128
	50000	21	15.20	14380
	100000	21	43.55	39274

Table 10 Evaluation results for PLIANTENERGY

Hosts	Cloud-lets	VMs	Energy (kWh)	Time (sec)
100	1000	12	0.18	669
	10000	16	2.34	3788
	50000	18	12.99	14380
	100000	18	34.55	39274

After examining these results, we have modified the normalization parameters of the applied Pliant system and created more focused algorithms. We changed the

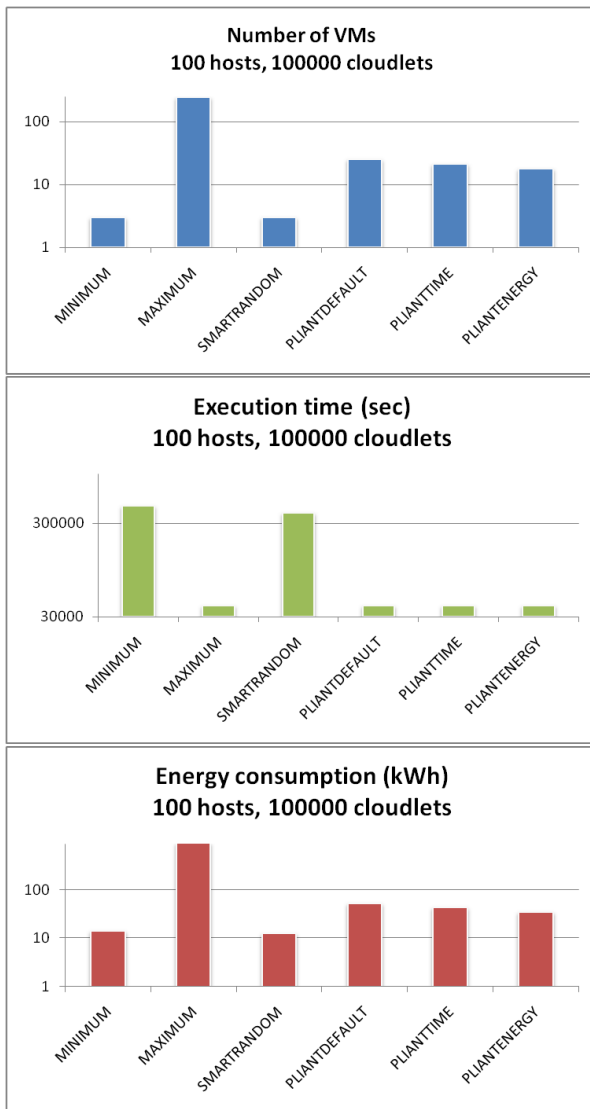


Fig. 5 Evaluation diagrams for 100000 cloudlets

sharpness of the Sigmoid function in order to emphasize the importance of execution time. We tried several normalization parameter combination to achieve our goal. In Table ?? we used a Pliant version that is more focused on execution time savings (PLIANTTIME), while in Table ?? we modified a Pliant parameter to focus on energy savings (PLIANTENERGY). Figure ?? shows comparison diagrams concerning the last rows of the tables.

5.3 Second evaluation round

As a second round of experiments, we used the whole Perzi log containing 14 days of resource utilization containing 3692457 jobs. In this round we used the same strategies as in the previous round, except for the SMAR-

TRANDOM, which we changed for the OPTUTIL, our original algorithm defined in the preliminary evaluations.

Figure ?? shows the number of utilized VMs, the measured execution time and energy usage for each day of the whole Perzi log. From these results we can see that the energy consumption is much lower in the 2nd, 3rd, 9th and 10th days, which is in correlation with the number of jobs submitted per day (shown in Table ??). Since the MINIMUM strategy operates with the lowest number of VMs, it has the lowest energy consumption, and on the contrary, the MAXIMUM has the highest. As we experienced in the first evaluation round, the Pliant strategies perform much better here as well than the OPTUTIL. Finally as we expected, the PLIANTENERGY has the lowest energy consumption among the Pliant algorithms.

Concerning the execution time, the MINIMUM strategy has a much worse performance than the others, which performed around the same at a global scale. If we take a look at the exact numbers shown in Table ??, it is also true that the MAXIMUM strategy takes the least time to execute all the jobs in the trace, and the PLIANTTIME is the closest to it.

Concerning the number of VMs used by the different strategies, the MAXIMUM and the OPTUTIL have the highest numbers, and the MINIMUM the lowest, obviously. PLIANTTIME has a bit more than the PLIANTDEFAULT, and as expected by us, the PLIANTENERGY has the lowest among the Pliant algorithms. We can also notice that the PLIANTTIME and PLIANTENERGY curves cross each other several times during the days, which means they really use different parameters to govern the number of available VMs. The sum of the measured values for the whole trace file are given in Table ?? and depicted in Figure ??.

As a result of these evaluations we can state that for minimal energy consumption the least amount of VMs should be used with smartly randomized VM selection (experienced in the first evaluation round). Nevertheless, when there is a need for execution time optimizations (as usual in real world systems), we have to find a trade-off between energy consumption and execution time. With our proposed Pliant-based VM scheduling algorithms we have shown that significant savings can be achieved in energy consumption with moderate execution time reductions.

6 Conclusion

Cloud computing is facing an increasing attention nowadays, but it raises severe issues with energy consumption: the higher levels of quality and availability require

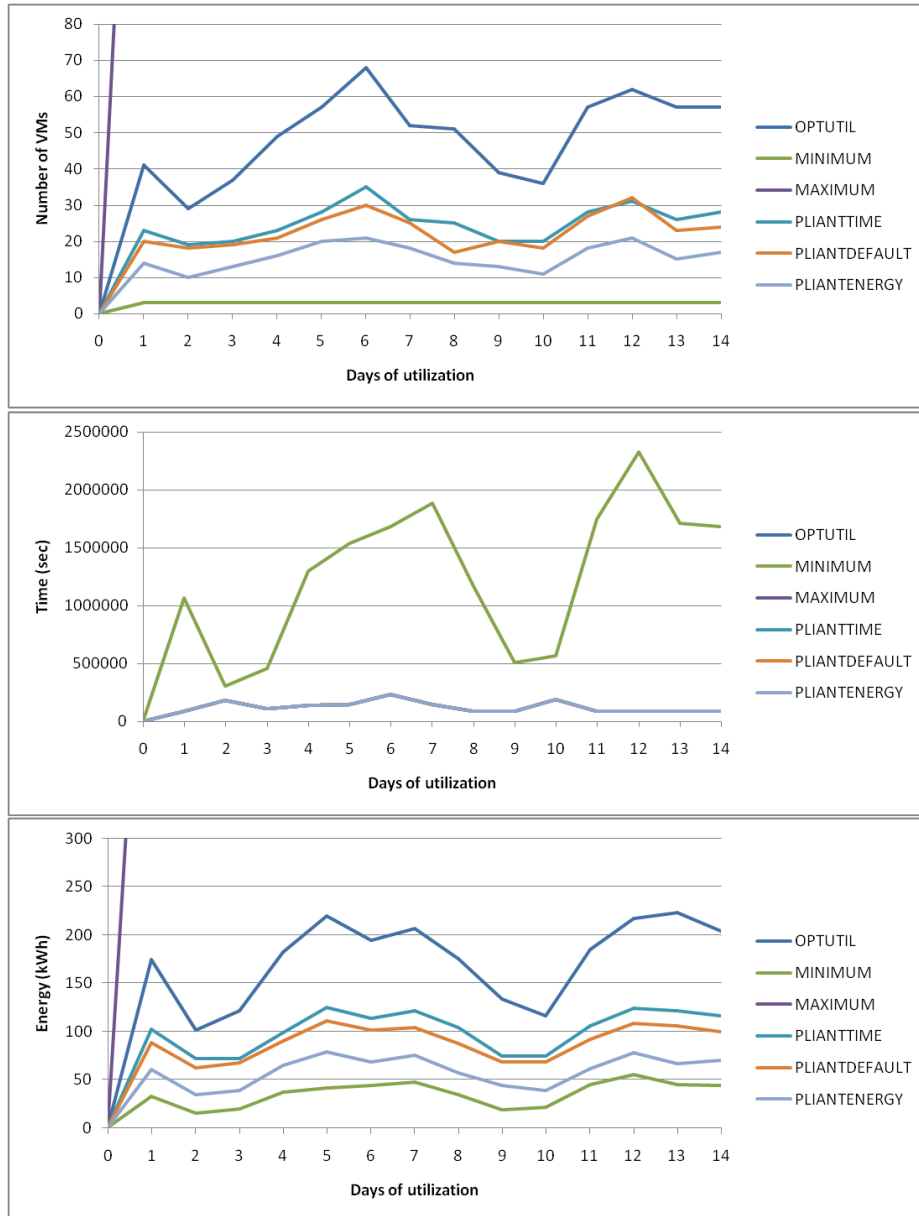


Fig. 6 Detailed evaluation results for the Prezi logs of 14 days

Table 11 Evaluation results for 14 days of Prezi logs

Metric	MINI- NUM	MAXI- MUM	OPT- UTIL	PLIANT- DEFAULT	PLIANT- TIME	PLIANT- ENERGY
VMs	3	241	68	32	35	21
Time	17921526	1751554	1751254	1752524	1751344	1753285
Energy	497.62	27793.51	2451.26	1250.26	1420.92	833.61

irrational energy expenditures. Reducing the carbon footprint of European countries is also a must, as well as to increase the number and size of European cloud providers.

In this paper we have proposed a Pliant system-based virtual machine scheduling approach for reducing energy consumption of IaaS cloud datacenters. We have

designed a CloudSim-based simulation environment for task-based cloud applications, and applied real-world traces for the performed experiments. We have shown that significant savings can be achieved in energy consumption with our proposed Pliant-based algorithms, and by fine-tuning the parameters of the proposed Pliant strategy, a beneficial trade-off can be set between

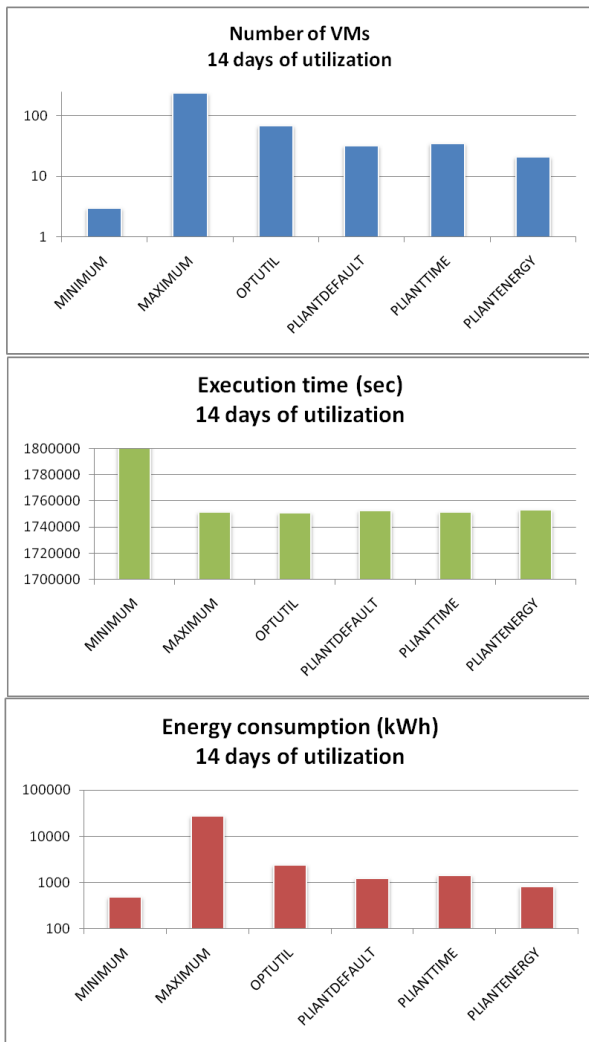


Fig. 7 Evaluation diagrams for the Prezi logs of 14 days

energy consumption and execution time. By incrementing the value of the alpha parameter of the Sigmoid function in our experiments resulted in less energy consumption.

Our future work aims at automating the parameter selection in different IaaS systems, and adapting the proposed approach to production-level academic clouds.

7 Acknowledgment

The research leading to these results has received funding from the European COST programme under Action identifier IC1304 (ACROSS), and it was supported by the European Union and the State of Hungary, co-financed by the European Social Fund in the framework of TAMOP 4.2.4. A/2-11-1-2012-0001 'National Excellence Program'. The paper is an extended and revised version of a conference short paper [?].

References

1. R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 2009, vol. 25, issue. 6, pp. 599–616.
2. M. D. Assuncao, R. Buyya and S. Venugopal. Inter-Grid: A Case for Internetworking Islands of Grids. *Concurrency and Computation: Practice and Experience (CCPE)*, Jul. 16 2007.
3. M. vor dem Berge, G. Da Costa, A. Kopecki, A. Oleksiak, J-M. Pierson, T. Piontek, E. Volk, S. Wesner. Modeling and simulation of data center energy-efficiency in CoolEmAll Energy Efficient Data Centers. *Lecture Notes in Computer Science*, 73 (2012), pp. 2536.
4. J. L. Berral, I. Goiri, R. Nou, F. Julia, J. Guitart, R. Gavaldà and J. Torres. Towards energy-aware scheduling in data centers using machine learning. In *proceedings of the 1st International Conference on Energy-Efficient Computing and Networking*, pp. 215-224, 2010.
5. R. Buyya, R. Ranjan, and R. N. Calheiros. InterCloud: Utility-Oriented Federation of Cloud Computing Environments for Scaling of Application Services. *Lecture Notes in Computer Science: Algorithms and Architectures for Parallel Processing*. Volume 6081, 2010.
6. R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, and R. Buyya. CloudSim: A Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms. *Software: Practice and Experience (SPE)*, Volume 41, Number 1, pp. 23–50, 2011.
7. A. Beloglazov and R. Buyya. Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers. *Concurr. Comput. : Pract. Exper.* 24, 13, pp. 1397-1420, September 2012.
8. Paul Marshall, Kate Keahey and Tim Freeman: Elastic Site: Using Clouds to Elastically Extend Site Resources. *T. IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGrid 2010)*, Melbourne, Australia. May 2010.
9. Celesti, A., Tusa, F., Villari, M., and Puliafito, A. (2010). How to Enhance Cloud Architectures to Enable Cross-Federation. In *IEEE 3rd Conference on Cloud Computing (CLOUD)*. 2010.
10. L. Schubert, and K. Jeffery. *Advances in Clouds Research in Future Cloud Computing*, Report from the Cloud Computing Expert Working Group Meeting. Cordis (Online), BE: European Commission, 2012. Online: <http://cordis.europa.eu/fp7/ict/ssai/docs/future-cc-2may-finalreport-experts.pdf>
11. J. Dombi. A general class of fuzzy operators, the demorgan class of fuzzy operators and fuzziness measures induced by fuzzy operators. *Fuzzy Sets and Systems* 8, 1982.
12. J. Dombi. Pliant system. *IEEE International Conference on Intelligent Engineering System Proceedings*, Budapest, Hungary, 1997.
13. J. D. Dombi, A. Kertesz. *Advanced Scheduling Techniques with the Pliant System for High-Level Grid Brokering*. *Communications in Computer and Information Science (CCIS)*, Vol. 129, Springer-Verlag Berlin Heidelberg, pp. 173–185, 2011.
14. A. Kertesz. Characterizing Cloud Federation Approaches. In book: *Cloud Computing - Challenges, Limitations and R&D Solutions*, Zaigham Mahmood (Ed.),

- Springer Series on Computer Communications and Networks, pp. 277-296, 2014.
15. L. Lefvre and A. Orgerie. Towards Energy Aware Reservation Infrastructure for Large-Scale Experimental Distributed Systems. *Parallel Processing Letters*, 19(3):419-433, September 2009.
 16. A. Kertesz, P. Kacsuk. GMBS: A New Middleware Service for Making Grids Interoperable. *Future Generation Computer Systems*, vol. 16, pp. 542-553, 2010.
 17. D. E. Knuth. *The art of computer programming. Seminumerical Algorithms*, 3rd edn., vol. 2. Addison-Wesley Longman Publishing Co., Inc., Boston, 1997.
 18. L. Liu, H. Wang, X. Liu, X. Jin, W. B. He, Q. B. Wang, and Y. Chen. Greencloud: a new architecture for green data center. In *Proceedings of the 6th international conference industry session on Autonomic computing and communications industry session*, ICAC-INDST '09, pages 29-38, New York, NY, USA, 2009. ACM.
 19. X. Meng, C. Isci, J. Kephart, L. Zhang, E. Bouillet, and D. Pendarakis. Efficient resource provisioning in compute clouds via vm multiplexing. In *Proceeding of the 7th international conference on Autonomic computing*, ICAC '10, pages 11-20, New York, NY, USA, 2010. ACM.
 20. KS. Park, VS. Pai. CoMon: a mostly-scalable monitoring system for PlanetLab. *ACM SIGOPS Operating Systems Review* 2006; 40(1):74.
 21. B. Khargharia, S. Hariri, and M. S. Yousif. Autonomic power and performance management for computing systems. *Cluster Computing*, 11(2):167-181, 2008.
 22. Y. Yu and S. Bhatti. Energy measurement for the cloud. *Parallel and Distributed Processing with Applications, International Symposium on*, 0:619-624, 2010.
 23. J. L. Lucas-Simarro, R. Moreno-Vozmediano, R. S. Montero, I. M. Llorente. Scheduling strategies for optimal service deployment across multiple clouds. *Future Generation Computer Systems*, doi: 10.1016/j.future.2012.01.007, 2012.
 24. T. Cioara, I. Anghel, I. Salomie, G. Copil, D. Moldovan, A. Kipp. Energy Aware Dynamic Resource Consolidation Algorithm for Virtualized Service Centers based on Reinforcement Learning. In *proc. of the 10th International Symposium on Parallel and Distributed Computing*, pp. 163-169, 2011.
 25. Eugen Feller, Louis Rilling, Christine Morin, Renaud Lottiaux, Daniel Leprince. Snooze: A Scalable, Fault-Tolerant and Distributed Consolidation Manager for Large-Scale Clusters. *IEEE/ACM Int'l Conference on Green Computing and Communications (GreenCom)*, pp. 125-132, 2010.
 26. Cardosa et al. Shares and Utilities based Power Consolidation in Virtualized Server Environments. In *Proc. IEEE/IFIP Conf. Integrated Management*, 2009.
 27. Y. C. Lee, C. Wang, A. Y. Zomaya, and B. B. Zhou. Profit-driven service request scheduling in clouds. In *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid)*, 2010, pages 15-24, May 2010.
 28. J. Berral, I. Goiri, R. Nou, F. Juli, J. Guitart, R. Gavald, J. Torres. Towards energy-aware scheduling in data centers using machine learning. In: *1st International Conference on Energy-Efficiency Computing and Networking*, Passau, Germany, 2010.
 29. E. Feller, C. Rohr, D. Margery, C. Morin. Energy Management in IaaS Clouds: A Holistic Approach. *IEEE International Conference on Cloud Computing (CLOUD)*, Honolulu, Hawaii, USA, June 2012.
 30. S. Salleh, B. Sanugi, H. Jamaluddin. Fuzzy logic model for dynamic multiprocessor scheduling. *Matematika*, 15(2), pp. 95-109, 1999.
 31. S. Sotiriadis, N. Bessis, N. Antonopoulos. Towards Inter-cloud Simulation Performance Analysis: Exploring Service-Oriented Benchmarks of Clouds in SimIC. In *proc. of the 27th International Conference on Advanced Information Networking and Applications Workshops (WAINA'13)*, Barcelona, Spain, pp. 765-771, 2013.
 32. A. Verma, P. Ahuja, A. Neogi. pMapper: power and migration cost aware application placement in virtualized systems. In *book: Middleware*, pp. 243-264, 2008.
 33. A. Benyi, J. D. Dombi and A. Kertesz. Energy-aware VM Scheduling in IaaS Clouds using Pliant logic. In *proc. of the 4th International Conference on Cloud Computing and Services Science (CLOSER'14)*, Barcelona, Spain, April 2014.
 34. Prezi Inc. "Scale Contest" website. <http://prezi.com/scale/>, Accessed on 23 April, 2013.
 35. SPEC website. www.spec.org, Accessed on 12 April, 2014.