Energy-aware VM Scheduling in IaaS Clouds using Pliant Logic

Attila Benyi[1], Jozsef Daniel Dombi[1] and Attila Kertesz[2,1]
[1]*Software Engineering Department, University of Szeged, 6720 Szeged, Dugonics ter 13, Hungary*
[2]*Institute for Computer Science and Control, MTA SZTAKI, H-1111 Budapest, Kende u. 13-17, Hungary*

Keywords: Cloud Computing, VM Scheduling, Pliant System, Simulation.

Abstract: Cloud Computing is facing an increasing attention nowadays as it is present in many consumer appliances 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 approach for reducing energy consumption of IaaS datacenters. In order to evaluate our proposed solution, we have designed a CloudSim-based simulation environment, and applied real-world traces for the experiments. We show 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 energy consumption and execution time.

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 (Buyya et al., 2009; Schubert and Jeffery, 2012).

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 (Lefèvre and Orgerie, 2009). Current trends are claimed to be clearly unsustainable with respect to resource utilisation, CO_2 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 deployment options that enable enhanced services such as disaster recovery. Thus these providers need to increase the size of their infrastructure by introducing 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

Benyi A., Daniel Dombi J. and Kertesz A..
Energy-aware VM Scheduling in IaaS Clouds using Pliant Logic.
DOI: 10.5220/0004842905190526
In *Proceedings of the 4th International Conference on Cloud Computing and Services Science* (CLOSER-2014), pages 519-526
ISBN: 978-989-758-019-2

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 (Berral et al., 2010). 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 (Schubert and Jeffery, 2012). By federating these providers, more competitive initiatives can be founded, that can be sophistically managed to meet these expectations. 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. To achieve this, we have already proposed an architecture called Federated Cloud Management (FCM – as introduced in (Kecskeméti et al., 2012)). In this holistic approach 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 provisioning 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 a Pliant system based approach (Dombi, 2012) to the management of these systems, which is similar to a fuzzy system (Dombi, 1982).

Therefore the main contributions of this paper are: (i) the development of a cloud simulation environment for task-based cloud applications, (ii) the design of an energy-aware and Pliant-based VM scheduling algorithm for VM management Clouds, 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 2 presents the related VM management approaches in datacenters; Section 3 introduces our extended simulation architecture; Section 4 introduces the advanced scheduling algorithms using the Pliant method for VM scheduling; and Section 5 describes the evaluation methodology and the simulation results. Finally, Section 6 summarizes the main contributions of the paper.

2 RELATED WORK

Regarding energy efficiency in a single cloud, Cioara et al. (Cioara et al., 2011) 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 (Feller et al., 2010), which is able to dynamically consolidate the workload of a heterogeneous large-scale cluster composed of resources using virtualization. In a later work (Feller et al., 2012), 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.

Cardosa et al. (Cardosa et al., 2009) 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. (Lee et al., 2010) discuss service request scheduling in Clouds based on achievable profits. They propose a pricing model using processor sharing for composite services in Clouds.

Lucas-Simarro et al. (Lucas-Simarro et al., 2013) 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. (Salleh et al., 1999) 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 (Dombi and Kertész, 2011) 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. (vor dem Berge et al., 2012) 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. (Sotiriadis et al., 2013) 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 neglect energy efficiency.

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;
```

3 SIMULATION OF CLOUDS

We have used the CloudSim simulator (Calheiros et al., 2011) to develop a simulation environment for our research. Beloglazov and Buyya (Beloglazov and Buyya, 2012) 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 (Park and Pai, 2006) 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 (Beloglazov and Buyya, 2012), 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. (Prezi, 2013) (discussed in detail in Section 5). 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 (SPEC, 2013). 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 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 1. 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 every VMs utilization 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. The Pliant system (Dombi, 2012) is a kind of fuzzy theory that is similar to a traditional fuzzy system (Dombi, 1982). The difference between the two systems lies in the choice of operators. The Pliant system has a strict, monotonously increasing t-norm and t-conorm, and the following expression is valid for the generator function:

$$f_c(x) f_d(x) = 1, \tag{1}$$

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. In our previous paper (Dombi and Kertész, 2011), we developed a scheduling component that uses the Pliant system to select a good performing Grid broker for a user's job even under conditions of high uncertainty. The algorithm we developed calculates 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. In our previous work (Dombi and Kertész, 2011) we found that if we use the aggregation operator to calculate the score number, we can achieve better results.

Here, we created two scheduling algorithms in order to handle the energy aware management case with a similar approach. One 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.

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

- The power consumption counter gives the energy usage of the given host at a given time. The value is generally between 40 and 120 MIPS, but it depends on the actual physical processor.
- The number of processors gives the available number of processors of a host.

We have developed two 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 the 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 define the value of the Sigmoid's parameter. Table 1 shows the predefined values of the parameters of the normalization functions.

In this environment 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 1, the minimum energy in this environment is around 40 and the maximum is around 120. Here 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.

After the normalization step we modify the normalized value to emphasize the importance of the result. To achieve this we will modify the normalized value by using the Kappa function with $\nu = 0.4$ and $\lambda = 3.0$ parameters:

$$\kappa_\nu^\lambda(x) = \frac{1}{1+\left(\frac{\nu}{1-\nu}\frac{1-x}{x}\right)^\lambda} \tag{2}$$

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

$$a_{\nu,\nu_0}(x_1,\cdots,x_n) = \frac{1}{1+\frac{1-\nu_0}{\nu_0}\frac{\nu}{1-\nu}\prod_{i=1}^{n}\frac{1-x_i}{x_i}}, \tag{3}$$

where ν is the neutral value and ν_0 is the threshold value of the corresponding negation. Here we don't

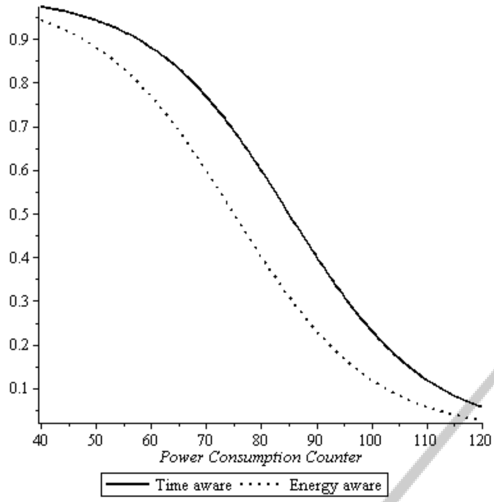

Figure 1: Utilized normalized function for the power consumption (PCC).

want to threshold the result so both parameters have the 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 the uploaded media files to other formats before they can display them on all devices. In April 2013, they launched a competition titled ”Scale Contest” (Prezi, 2013) 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 system for the jobs participating in the conversion processes:

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

Table 2: Evaluation results for RoundRobin.

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

Table 3: Evaluation results for OptUtil.

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

For a preliminary evaluation phase we used the trace file of the first week. We have performed experiments with datacenters having 100 to 500 hosts, and submitted 10000 to 100000 jobs 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 2. 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 3.

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

To develop Pliant-based algorithms, we created three initial strategies: the first one uses only one VM to execute all submitted jobs (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 5, 4 and 6 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 4: Evaluation results for MAXIMUM.

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

Table 5: Evaluation results for MINIMUM.

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

Table 6: Evaluation results for SMARTRANDOM.

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

Based on these preliminary evaluations we have created a Pliant-based strategy (PLIANTDEFAULT), first focusing on execution time reduction with some energy savings. For its default algorithm Table 7 shows the results of the simulation. This table shows that it 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.

Therefore we have modified the parameters of the applied Pliant system, and created more focused algorithms. In Table 8 we used a Pliant version that is more focused on execution time savings (PLIANTTIME), while in Table 9 we modified a Pliant parameter to focus on energy savings (PLIANTENERGY).

Table 7: Evaluation results for PLIANTDEFAULT.

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

Table 8: Evaluation results for PLIANTTIME.

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

Figure 2 shows comparison diagrams concerning the last rows of the tables.

As a result of the evaluations we can state that for minimal energy consumption the least amount of VMs should be used with smartly randomized VM selection. Nevertheless, when there is a need for execution time optimizations, 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 and it raises severe issues with energy consumption: the higher levels of quality and availability require irrational energy expenditures. 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.

In this paper we have proposed a Pliant system based virtual machine scheduling approach for reducing energy consumption of IaaS 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 energy consumption and execution time.

Our future work aims at automating the parameter selection in different IaaS systems, and adapting

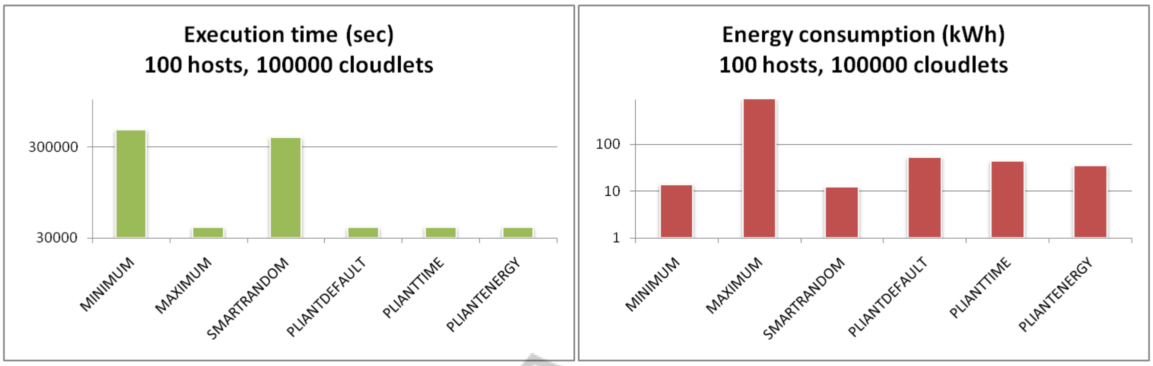

Figure 2: Comparison diagrams for 100000 cloudlets.

Table 9: Evaluation results for PLIANTENERGY.

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

the proposed approach in production-level academic Clouds.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the EU FP7 IDGF-SP project under grant agreement 312297, 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'.

REFERENCES

Beloglazov, A. and Buyya, R. (2012). Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurrency and Computation: Practice and Experience*, 24(13):1397–1420.

Berral, J. L., Goiri, I., Nou, R., Julià, F., Guitart, J., Gavaldà, R., and Torres, J. (2010). Towards energy-aware scheduling in data centers using machine learning. In de Meer, H., Singh, S., and Braun, T., editors, *e-Engery*, pages 215–224. ACM.

Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., and Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Comp. Syst*, 25(6):599–616.

Calheiros, R. N., Ranjan, R., Beloglazov, A., Rose, C. A. F. D., and Buyya, R. (2011). Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Softw, Pract. Exper*, 41(1):23–50.

Cardosa, M., Korupolu, M. R., and Singh, A. (2009). Shares and utilities based power consolidation in virtualized server environments. In *Integrated Network Management*, pages 327–334. IEEE.

Cioara, T., Anghel, I., Salomie, I., Copil, G., Moldovan, D., and Kipp, A. (2011). Energy aware dynamic resource consolidation algorithm for virtualized service centers based on reinforcement learning. In *ISPDC*, pages 163–169. IEEE.

Dombi, J. (1982). A general class of fuzzy operators, the De Morgan class of fuzzy operators and fuzziness measures induced by fuzzy operators. *Fuzzy Sets and Systems*, 8:149–163.

Dombi, J. (2012). Pliant operator system. In Fodor, J. C., Klempous, R., and Araujo, C. P. S., editors, *Recent Advances in Intelligent Engineering Systems*, volume 378 of *Studies in Computational Intelligence*, pages 31–58. Springer.

Dombi, J. D. and Kertész, A. (2011). Advanced scheduling techniques with the pliant system for high-level grid brokering. In Filipe, J., Fred, A., and Sharp, B., editors, *Agents and Artificial Intelligence*, volume 129 of *Communications in Computer and Information Science*, pages 173–185. Springer Berlin Heidelberg.

Feller, E., Rilling, L., Morin, C., Lottiaux, R., and Leprince, D. (2010). Snooze: A scalable, fault-tolerant and distributed consolidation manager for large-scale clusters.

Feller, E., Rohr, C., Margery, D., and Morin, C. (2012). Energy management in iaaS clouds: A holistic approach. In *Proc. 2012 IEEE Fifth International Conference on Cloud Computing (5th IEEE CLOUD'12)*, pages 204–212, Honolulu, HI, USA. IEEE Computer Society.

Kecskeméti, G., Kertész, A., Marosi, A., and Kacsuk, P. (2012). Interoperable resource management for establishing federated clouds. In Villari, M., Brandic, I., and Tusa, F., editors, *Achieving Federated and Self-Manageable Cloud Infrastructures: Theory and Practice*, pages 18–35. IGI Global, Hershey.

Lee, Y. C., Wang, C., Zomaya, A. Y., and Zhou, B. B. (2010). Profit-driven service request scheduling in clouds. In *CCGRID*, pages 15–24. IEEE.

Lefèvre, L. and Orgerie, A.-C. (2009). Towards energy aware reservation infrastructure for large-scale experimental distributed systems. *Parallel Processing Letters*, 19(3):419–433.

Lucas-Simarro, J. L., Moreno-Vozmediano, R., Montero, R. S., and Llorente, I. M. (2013). Scheduling strategies for optimal service deployment across multiple clouds. *Future Generation Comp. Syst*, 29(6):1431–1441.

Park, K. and Pai, V. S. (2006). Comon: a mostly-scalable monitoring system for planetlab. *Operating Systems Review*, 40(1):65–74.

Prezi (2013). Prezi Inc. "scale contest" website, http://prezi.com/scale/, april 2013.

Salleh, S., Sanugi, B., and Jamaluddin, H. (1999). Fuzzy logic model for dynamic multiprocessor scheduling.

Schubert, L. and Jeffery, K. (2012). Advances in clouds - research in future cloud computing.

Sotiriadis, S., Bessis, N., and Antonopoulos, N. (2013). Towards inter-cloud simulation performance analysis: Exploring service-oriented benchmarks of clouds in simIC. In Barolli, L., Xhafa, F., Takizawa, M., Enokido, T., and Hsu, H.-H., editors, *AINA Workshops*, pages 765–771. IEEE Computer Society.

SPEC (2013). Spec website, http://www.spec.org, august 2013.

vor dem Berge, M., Costa, G. D., Kopecki, A., Oleksiak, A., Pierson, J.-M., Piontek, T., Volk, E., and Wesner, S. (2012). Modeling and simulation of data center energy-efficiency in coolemall. In Huusko, J., de Meer, H., Klingert, S., and Somov, A., editors, *E2DC*, volume 7396 of *Lecture Notes in Computer Science*, pages 25–36. Springer.