



**HAL**  
open science

## Seeking for the Optimal Energy Modelisation Accuracy to Allow Efficient Datacenter Optimizations

Edouard Outin, Jean-Emile Dartois, Olivier Barais, Jean-Louis Pazat

► **To cite this version:**

Edouard Outin, Jean-Emile Dartois, Olivier Barais, Jean-Louis Pazat. Seeking for the Optimal Energy Modelisation Accuracy to Allow Efficient Datacenter Optimizations. 16th International Symposium on Cluster, Cloud and Grid Computing, May 2016, Cartagena, Italy. 10.1109/CCGrid.2016.67 . hal-01356099

**HAL Id: hal-01356099**

**<https://hal.science/hal-01356099>**

Submitted on 24 Aug 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Seeking for the optimal energy modelisation accuracy to allow efficient datacenter optimizations

Edouard Outin, Jean-Emile Dartois  
*b-com*  
Rennes, France  
Email: *firstname.lastname@b-com.com*

Olivier Barais  
*b-com, IRISA, INRIA, Univ. Rennes 1*  
Rennes, France  
Email: *barais@irisa.fr*

Jean-Louis Pazat  
*b-com, IRISA, INRIA, INSA*  
Rennes, France  
Email: *jean-louis.pazat@irisa.fr*

**Abstract**—As cloud computing is being more and more used, datacenters play a large role in the overall energy consumption. We propose to tackle this problem, by continuously and autonomously optimizing the cloud datacenters energy efficiency. To this end, modeling the energy consumption for these infrastructures is crucial to drive the optimization process, anticipate the effects of aggressive optimization policies, and to determine precisely the gains brought with the planned optimization. Yet, it is very complex to model with accuracy the energy consumption of a physical device as it depends on several factors. Do we need a detailed and fine-grained energy model to perform good optimizations in the datacenter? Or is a simple and naive energy model good enough to propose viable energy-efficient optimizations? Through experiments, our results show that we don't get energy savings compared to classical bin-packing strategies but there are some gains in using precise modeling: better utilization of the network and the VM migration processes.

**Keywords**-cloud; placement optimization; energy models;

## I. INTRODUCTION

Optimizing datacenter efficiency is one of the main challenges of the cloud computing field. It can be seen as managing system resources (servers and networking) in an energy-efficient way, while handling the dynamicity of the resources demand and utilization. With the growth of the cloud adoption both in industry and individuals, the energy consumption and the CO<sub>2</sub> emissions are becoming very huge. Cloud providers are more and more using optimization techniques to increase the cloud efficiency, they aim at reducing their TCO while guaranteeing the same service. In order to adapt the system we first have to model the system. Indeed, these techniques are using a datacenter abstraction to investigate the available optimizations. It is therefore necessary to ask which granularity in the modelisation is necessary to run efficient algorithms for the energy optimization. Cloud datacenters are complex systems, running thousands of servers and networking equipments while hosting huge amounts of heterogeneous VMs. These VMs, have different workloads changing over time, and have different flavors and utilization levels. Moreover, determining the optimal cloud datacenter configuration, while meeting the SLA previously negotiated with the customers, is challenging. Actually, a

cloud user can specify some constraints on the deployment of its VMs (affinity, anti-affinity, location...) and the provider has to comply with these restrictions; otherwise some economic penalties can be applied. To tackle this challenge, we propose the use of metaheuristics to find out the near optimal solution to a datacenter and a set of virtual machines assigned to it. in order to reduce the energy consumption in the datacenter.

We design an autonomous adaptation engine to address the cloud datacenters issues, through the genetic algorithms. Yet, to be able to reason and seek for the best datacenter configuration, we need to represent the problem and all its characteristics. Indeed, we need to model the compute nodes and their resources, the virtual machines and their utilization, the networking equipments... but also, we have to model all the energy consumption of the physical devices. Then, the genetic algorithm is able to manipulate this model, perform recombinations and mutations on the model, and evaluate the quality of the new datacenter configurations (in terms of energy-efficiency, SLA violations, performance). Modeling the servers energy consumption is not trivial. It depends on a lot of factors such as CPU and RAM utilization, but disk and network activities are energy consuming too for instance. Therefore, we need to predict for each physical device, its energy consumption, based on its different utilization levels and features. Nevertheless, very precise modeling of the datacenters and its components can be costly in terms of computation time, energy consumption, and implementation. Whereas, a naive and simplistic model can have good performance and easy computation. But what about the results they provide? Are the optimizations computed with a fine-grained model more efficient than the ones with the coarse-grained model? There is a trade off to find to solve the datacenter configuration problem in an efficient way: accuracy in the model, but no additional computations, if it doesn't improve the solution's quality.

This paper is organized as follows. Section II presents our multi-objective genetic algorithm framework we developed, its model manipulations and its solutions evaluations. Section III reviews the related work in this field of cloud computing and presents the energy models we use in this

work. In the section IV, we deal with the cloud simulator and its modifications. Finally, section V and VI respectively deal with the experiment and the results.

## II. GENETIC ALGORITHM FRAMEWORK

### A. General

The evolutionary algorithm (EAs) is a class of stochastic optimization methods that imitates the process of natural evolution. The EA can be used for solving both constrained and unconstrained optimization problems that are based on natural selection. The optimization process starts by creating a random initial population (made of several individuals representing a potential cloud configuration). The optimization process then generates a sequence of new populations by cloning the model. The model is representing the hardware of an IaaS infrastructure and the mapping of the different virtual machines on the compute nodes. For instance, an IaaS infrastructure is composed of a set of hosts, which are responsible for managing VMs during their life cycles. At each step, the algorithm uses the individuals in the current generation to create the next population (going towards an elite population). There are several popular and efficient genetic algorithms in the available frameworks and the scientific literature. Among the algorithms we evaluated we can cite: *MOEA/D*, *NSGA-II* or  $\epsilon$ -*MOEA*. In our work we use *NSGA-II*, as it outperforms others algorithms and handles well both constrained and unconstrained problems.

### B. Adaptation to the cloud problem

In order to adapt the genetic algorithm framework to the cloud energy-efficiency optimization problem we must map the genetic engine with the underlying model. Indeed, the GA performs recombinations by manipulating the cloud configuration model through mutators. We propose several mutators to fully discover the search space. For example, among them there are:

- *MigrateRandomVms*: migrates a random number of VMs (random source and destination hosts)
- *PermuteOneVm*: permutes a VM from a given host with a VM from another host

Moreover, the genetic algorithm has to evaluate each individual of the generations in order to be able to rank them following different fitness functions. For instance, we can evaluate a given cloud configuration in terms of potential energy consumption, number of virtual machine migrations required or constraint violations (such as SLA or host capacity). To assess the quality of an individual, the algorithm uses fitness functions. Each fitness function corresponds to an objective of the optimization problem. And through these functions and the genetic algorithm we are able to find out the Pareto front of the cloud configuration problem. It contains the different individuals having the best global fitness scores (the optimal tradeoffs between the different objectives).

## III. ENERGY MODELS

In this section, we present and analyze the different energy models that are used in this paper. Indeed, to be able to improve the datacenters energy efficiency, we must model the energy consumption of the different physical devices and the parameters affecting this consumption. Then, we can tune and modify the current configuration through our model at runtime, and check whether the envisaged new configuration is interesting or not, in terms of energy savings. Indeed, by understanding how energy is consumed in the datacenter and what are the factors impacting this consumption, we aim at improving the energy efficiency. These energy models can be embedded in the genetic algorithm as fitness functions to evaluate the different solutions.

### A. Server energy model

Servers energy models allow us to predict the Watts consumed by compute servers based on their utilization metrics. For instance, a popular and widely used energy model is *SPECpower\_ssj2008* benchmark [1]. It is designed to measure the power characteristics of server-class computer equipment. Moreover, it allows comparing power and performance among different servers and serves as a toolset for use in improving server efficiency. Nevertheless, these SPEC energy models, only take into account the CPU utilization to determine the energy consumption of the whole system (as it is the resource that consumes most of the host's power [9]). This can lead to some approximations or errors. Actually, we know that disk and network operations, for instance, have an impact on the total server consumption [9]. In order to get more accurate energy prediction for the compute nodes, other metrics should be taken into account and not only the CPU. In [4], the authors propose a high-level non-linear model to predict the server energy consumption based on its current utilization levels. They design an energy model for a specific server type using regression and statistical machine learning techniques. Once the model is trained, it can predict energy consumption accurately, given the CPU and RAM utilization, and the disk and network activity.

### B. Network devices energy model

Compute servers consume a lot of energy in the datacenter context, but the networking devices can not be put aside [5]. Energy efficiency is becoming increasingly important in the operation of networking infrastructure, especially in enterprise and datacenter networks, where the switches and routers can have tremendous work to process. It is clear that live migration processes and inter-VMs communications consume variable network bandwidth depending on the datacenter cloud configuration. We need a comprehensive characterization of energy consumption for switches and routers to accurately quantify the savings from the various energy savings schemes. Energy management for networking devices in wired networks has not received much

attention in the literature, but we can find some interesting research work [8]. In [6], the authors propose a linear regression analysis on a switch in operation mode. They vary three different input parameter while measuring the energy consumption of the network device: bandwidth, number of connections and link load. They find out that the link load has barely an impact on the switch energy consumption while the bandwidth and the number of active connections are the determining inputs.

### C. VM migration energy model

Live migration of virtual machines is more and more used to consolidate and rearrange the current workload in the datacenters. Nevertheless, it is plain to see that this technique, consisting of migrating a virtual machine from one hypervisor to another with almost no downtime, has an energy overhead and might introduce some QoS degradations for the migrating VM. Moreover, we can note that the volume of data needed to be transferred to achieve a live migration can be up to a few gigabytes. Therefore, it is clear that triggering too many virtual machine migrations can lead to a traffic saturating network links. Then, the network being flooded, there might be a lack of resources to guarantee the optimal QoS to the customers and to fulfill the management of the infrastructure. To model the VM migration energy overhead in our experiment we decide to rely on the state of the art techniques, as detailed in this work [7]. The authors design a high-level linear model to estimate the energy required to migrate a virtual machine from one physical host to the other. Finally, we end up with the linear model below:

Let  $E_{mig}$  denote the total energy overhead of the migration (in Joules) and  $V_{mig}$  the total network traffic during migration (in Mb).

$$E_{mig} = 0.512 * V_{mig} + 20.165$$

With  $1J \approx 2.77778E-7$  kWh.

We are now able to characterize a virtual machine migration in terms of energy consumption. Note that we do not express the optional downtime that can be induced by the migration, nor the possible QoS degradation for the users. Nevertheless, migrations introduce additional bandwidth utilization and we express the corresponding link load of the connections in our network topology model.

## IV. CLOUDSIM SIMULATOR

### A. Presentation

CloudSim is a holistic framework for modeling, simulating and experimenting with cloud computing environments [3]. It is developed at the CLOUDS laboratory of the University of Melbourne. It is written in Java and is licensed under GNU LGPL v3. This simulation framework focuses on system design issues without having to worry about

the low level details inherent to cloud-based infrastructures and services. It has support for modeling and simulation of energy-aware computational resources. It is also widely used to evaluate strategies in cloud computing, such as policies, mapping or load balancing rules. Additionally, CloudSim is useful to simulate one's own cloud environment with the infrastructure, policies and algorithms, in order to check and validate the system behavior before deploying it in the real infrastructure. We decided to use this simulator as it is very popular in the research area, has energy-awareness and an active community.

### B. Algorithms in CloudSim

CloudSim provides a set of scheduling and placement algorithms [2]. There are several policies available to select the virtual machines to migrate and where to migrate them. Nevertheless, it is possible for a user to add its own algorithm in the simulation framework. Below are the classes we target:

*VmAllocationPolicy* – Class to be extended in order to implement new algorithms for deciding which host a new VM should be placed on.

*PowerVmAllocationPolicyMigrationAbstract* – A template class for implementing power-aware dynamic VM consolidation algorithms that use VM live migration to dynamically reallocate VMs at every time frame.

### C. Adding custom algorithm

We decide to embed our genetic algorithm framework in the CloudSim project in order to check and validate the results obtained through this simulator. Our genetic algorithm doesn't handle the scheduling and the initial placement of the virtual machine, so we let CloudSim manage the VM provisioning. To include our genetic engine in CloudSim we create a new class *VmAllocationPolicyMigrationGA* extending the *PowerVmAllocationPolicyMigrationAbstract* class, and we override the *optimizeAllocation* method in order to trigger the genetic algorithm initialization. Moreover, we need to be able to capture any cloud configuration in the simulator and to inject it in our model to allow reasoning and optimization on it through the genetic algorithms. Finally we also need effectors to apply the changes previously computed in the simulator.

### D. Workload

In order to evaluate and compare the different energy models and their respective effects on the datacenter optimization we use PlanetLab workload traces. The data contains CPU utilization data of multiple VMs. In every VM workload, we have the mean CPU utilization values for every 5 minutes, and for approximately 1200 VMs. We then define how much physical compute servers we deploy in the infrastructure to host this set of VMs, and then CloudSim schedules the incoming requests on the compute servers. The

PlanetLab workload contains 10 days of VM traces, taken from 2011 March and April.

## V. EXPERIMENT

### A. Goal

Through these experiments we want to validate the necessity of modeling the cloud datacenter and its features in order to optimize the energy efficiency of the infrastructure. Moreover, we aim at discovering which level of accuracy is necessary in the model: model of the cloud datacenter and the network, as well as the energy models evaluating the different potential energy consumption (servers, network, live migrations). To this end, we propose to apply our dynamic VM placement algorithm and its models into CloudSim simulation with a PlanetLab workload. The algorithm we designed can be run using two different models:

**Coarse-grained model:** simple model of the cloud datacenter: physical nodes, virtual machines and the mapping of the virtual machines onto the physical hosts. Energy consumption of a host node is determined using a linear model based on the CPU utilization of the host.

**Fine-grained model:** more precise model of the cloud datacenter. Network links and devices are modeled. Energy consumption of a host node is predicted through complex model derived from machine learning. This model takes as input variables several metrics to predict the energy consumed by the host. We also use a VM migration energy model to be able to determine how much does a virtual machine migration consume. Finally, we also include a network devices energy model to take into account the energy consumption of the different switches and routers.

### B. Experiment protocol

For this set of experiments we use CloudSim 3.0.3 and the provided PlanetLab workloads. Our genetic algorithm relies on the MOEA Framework 2.6 and all the experiments are done using a regular GNU/Linux desktop with 3.13 kernel. The experiment is set to 24h and we trigger an optimization in the infrastructure every 60min. We evaluate three different algorithms and models:

- CloudSim bin-packing (LrMu) [2]
- Genetic algorithm with fine-grained model
- Genetic algorithm with coarse-grained model

For each experiment, we deploy 500 compute nodes hosting the virtual machines. The physical servers set is homogeneous and is composed of hosts with 20 GB RAM and 10 physical processors. We set a compute time limit of 40 seconds to our genetic algorithm to let it explore the search space and discover new cloud datacenter configurations.

## VI. RESULTS

### A. Measurements

The results of the different experiments are summarized in the table I. Below are the different acronyms used in

the display of the measurements: **CG** and **FG** respectively correspond to the coarse-grained model with the NSGA-II algorithm and to the fine-grained model with the NSGA-II algorithm. Whereas **LrMu** stands for CloudSim placement algorithm with Local Regression allocation policy and Maximum Utilization. Note that the energy consumption unit is the *kWh*.

### B. Built-in algorithms vs. model based heuristic

Comparing the results obtained through the CloudSim bin-packing algorithm (LrMu) with the results from the genetic algorithm (CG and FG), we notice that in terms of energy consumption the LrMu outperforms the GA algorithms. Nevertheless, we can note that to achieve this energy savings, the LrMu triggers a lot of virtual machine migrations, resulting in more physical hosts that can be shutdown. LrMu performs around two or three times the migrations that the genetic algorithm would have done for the same workload. Moreover, the gains in terms of servers freed from VM instances is not proportional. This huge amount of virtual machine migrations leads to an energy consumption overhead, a huge utilization of the datacenter's network and a potential QoS degradation. This massive network utilization is not represented accurately in CloudSim. It is clear that performing around 2000 VM migrations out of around 1400 VMs over a 24 hours period can affect a lot the network and is quite hard to fulfill. For every VM migration, the volume of transmitted data between the source and destination host can be from 200 Mb to a few gigabytes. Thus, it is plain that 2000 migration of 1400 VMs over a 24 hours period of time, with 500 hosts having 1Gbps network connectivity is intractable. Here we can see that the objective of the cloud configuration optimization can not only be the minimization of the running physical servers. We must also take into account the number of virtual machine migrations, and their effect on the system. Indeed, we must find the right tradeoff between the number migrations and the servers consolidation. Our genetic algorithm can find the near-optimal tradeoff between the number of VM migrations and the number of active servers. It proposes a few solutions of the Pareto front which are more or less aggressive regarding the changes required.

### C. Fine-grained model vs. coarse-grained model

Comparing the results of CG and FG, running the same algorithm but with different underlying models, we can see that the gains in terms of energy efficiency are limited. Modeling the network topology and the different networking devices helps the algorithm to have a better vision of the datacenter system. It drives the migrations allocations policies by fostering the best virtual machines to migrate: the VMs with the least memory volume to transmit over the network. Moreover it fosters better physical servers (in terms of network topology, link utilization...) to host the VMs

Table I  
MEASUREMENTS

Workload	2011-03-22			2011-03-25			2011-04-03			2011-04-09		
Algorithm	CG	FG	LrMu	CG	FG	LrMu	CG	FG	LrMu	CG	FG	LrMu
Num. of hosts	500	500	500	500	500	500	500	500	500	500	500	500
Num. of VMs	1516	1516	1516	1078	1078	1078	1463	1463	1463	1358	1358	1358
Num. of VM migrations	1186	1258	2035	603	842	1678	478	568	1929	555	667	1940
Num. of hosts shutdown	371	376	429	395	401	442	354	356	418	375	379	432
Energy consumption	211.10	210.65	163.82	253.72	251.66	136.50	349.76	349.39	173.08	309.15	308.97	158.15

being migrated. Nevertheless, these effects are not visible through the energy consumption metric. The QoS levels and the network utilization should be taken into account too. With the different experiments results we notice that the placement configurations computed by FG are slightly different from the ones from CG.

### VII. PERSPECTIVES

The results of the different simulation experiments look promising, but the gains are limited in terms of energy-efficiency. The next step would be to run these experiments on a real cloud infrastructure to validate the benefits of using fine-grained energy models. Moreover, the workload provided by PlanetLab only consists in CPU intensive VMs, but to be more realistic we need to experiment with various and realistic VMs doing different tasks and utilizing various resources. Moreover, energy-efficiency is very important on the provider side, but we must also take QoS into consideration while optimizing the cloud datacenter. Indeed, it is a crucial metric for the customers of the cloud. If we consolidate and shrink the VMs too much, we might end up with VMs having not enough resources to cope with the customers requests, and thus a very bad QoS leading to bad user experience.

### VIII. CONCLUSION

Through this work we could discover the effects of different granularity modelisation to solve the cloud VM placement optimization problem: a fine-grained modelisation, a coarse-grained modelisation and no modelisation at all. With the simulation results, we discover that the results with no energy modelisation can induce a potential massive amount of virtual machine migrations, and thus they foster the consolidation process leading to energy savings in the datacenter infrastructure. Nevertheless, these very aggressive optimization policy are often intractable in a real cloud system. Triggering too much migrations can lead to SLA violation and bad user experience. By modeling the datacenter network, its topology and utilization links, we aim at minimizing the virtual machine migrations while consolidating the VMs in the datacenter infrastructure. Moreover, by using accurate server energy models we are able to characterize with accuracy the changes, in terms of energy, planned in the cloud optimization.

### ACKNOWLEDGMENT

This work has been achieved within the Institute of Research & Technology b<>com, dedicated to digital technologies. It has been funded by the French government through the National Research Agency (ANR) Investment referenced ANR-A0-AIRT-07.

### REFERENCES

- [1] SPEC.org ssj2008. [https://www.spec.org/power\\_ssj2008/](https://www.spec.org/power_ssj2008/), 2008. [Online; accessed 09-Nov-2015].
- [2] Anton Beloglazov and Rajkumar Buyya. 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*, 2012.
- [3] Rodrigo N Calheiros, Rajiv Ranjan, Anton Beloglazov, César AF De Rose, and Rajkumar Buyya. Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Software: Practice and Experience*, 2011.
- [4] Outin Edouard, Dartois Jean-Emile, Barais Olivier, and Pazat Jean-Louis. Enhancing cloud energy models for optimizing datacenters efficiency. ICCAC, 2015.
- [5] Helmut Hlavacs, Georges Da Costa, and Jean-Marc Pierson. Energy consumption of residential and professional switches. In *Computational Science and Engineering, 2009. CSE'09. International Conference on*, 2009.
- [6] Md Mohaimenul Hossain, Eric Rondeau, Jean-Philippe Georges, and Thierry Bastogne. Modeling the power consumption of ethernet switch. In *International SEEDS Conference 2015: Sustainable Ecological Engineering Design for Society*, 2015.
- [7] Haikun Liu, Hai Jin, Cheng-Zhong Xu, and Xiaofei Liao. Performance and energy modeling for live migration of virtual machines. *Cluster computing*, 2013.
- [8] Priya Mahadevan, Puneet Sharma, Sujata Banerjee, and Parthasarathy Ranganathan. A power benchmarking framework for network devices. In *NETWORKING 2009*. Springer, 2009.
- [9] Suzanne Rivoire, Mehul A Shah, Parthasarathy Ranganathan, Christos Kozyrakis, and Justin Meza. Models and metrics to enable energy-efficiency optimizations. 2007.