An Energy Aware Framework for Virtual Machine Placement in Cloud Federated Data Centres

Corentin Dupont*
CREATE-NET
Trento, Italy
cdupont@create-net.org

Giovanni Giuliani
HP Italy Innovation Centre
Milan, Italy
giuliani@hp.com

Fabien Hermenier
OASIS Team, INRIA – CNRS – I3S
University of Sophia-Antipolis
fabien.hermenier@inria.fr

Thomas Schulze
University of Mannheim
Mannheim, Germany
schulze@informatik.uni-mannheim.de

Andrey Somov
CREATE-NET
Trento, Italy
asomov@create-net.org

ABSTRACT

Data centres are powerful ICT facilities which constantly evolve in size, complexity, and power consumption. At the same time users' and operators' requirements become more and more complex. However, existing data centre frameworks do not typically take energy consumption into account as a key parameter of the data centre's configuration. To lower the power consumption while fulfilling performance requirements we propose a *flexible* and *energy-aware* framework for the (re)allocation of virtual machines in a data centre. The framework, being independent from the data centre management system, computes and enacts the best possible placement of virtual machines based on constraints expressed through service level agreements. The framework's flexibility is achieved by decoupling the expressed constraints from the algorithms using the *Constraint Programming* (CP) paradigm and programming language, basing ourselves on a cluster management library called *Entropy*. Finally, the experimental and simulation results demonstrate the effectiveness of this approach in achieving the pursued energy optimization goals.

Categories and Subject Descriptors

D.4.7 [OPERATING SYSTEMS]: Organization and Design---Distributed systems

General Terms

Algorithms, Design, Performance, Experimentation.

Keywords

Constraint Programming, Cloud Computing, Data Centre, Resource Management, Energy Efficiency, Virtualization, Service Level Agreement.

1. INTRODUCTION

Data centres are ICT facilities aimed at information processing and computing/telecommunication equipment hosting purposes for scientific and/or business customers. Until recently, data centre operation management has been entirely focused on improving metrics like performance, reliability, and service availability. However, due to the rise of service demands, data centres evolve in complexity and size. This and the continuous increase of energy cost have prompted the ICT community to add energy efficiency as a new key metric for improving data centres facilities. This trend was further boosted by the acknowledgement that the ICT sector's carbon emissions are increasing faster than in any other domain [1]. Therefore researchers and IT companies have been solicited to find energy-aware strategies for the operation of data centres [2].

To tackle this problem, a number of energy-aware approaches have been recently proposed in the literature and research projects like e.g. workload consolidation [3][5], optimal placement of workload [6], scheduling of applications [1][7], detection of more power efficient servers [8], and the reduction of power consumption by cooling systems [4].

It should be noted, however, that most of the energy-aware approaches and resource management algorithms for data centres consider only specific research problems and integrate typical constraints not taking some important factors into account:

Data centres have complex and quickly changing configurations;

Data centres are not homogeneous in terms of performance, management capabilities, and energy efficiency;

Data centres must comply with a number of users' and operators' requirements.

Due to the growing number of constraints and their complexity we need to separate them from the resource management algorithm(s) to secure the two-folded objective:

Being able to add or modify a constraint without changing the algorithms

Being able to test and activate a new algorithm without having to re-implement every constraint within it.

In this paper we propose and discuss a flexible energy-aware framework to address the problem of energy-aware allocation/consolidation of Virtual Machines (VMs) in a cloud

e-Energy 2012, May 9-11 2012, Madrid, Spain.
Copyright 2012 ACM 978-1-4503-1055-0/12/05 ...$10.00.

[*] The authors are listed in alphabetic order.

data centre. The core element of the framework is the *optimizer* which is able to deal with (1) Service Level Agreement (SLA) requirements, (2) different data centres interconnected in a federation, each with their own characteristics, as well as (3) two different end objectives, namely minimizing energy consumption or CO_2 emissions.

This framework is developed and tested within the FIT4Green project [23], funded by the Commission of the European Union, whose main goal is reducing the direct energy consumption of ICT resources of a data centre by 20%. In practice, it relies on *Constraint Programming* (CP) paradigm and the *Entropy* open source library [13] to compute the energy-aware placement of VMs. This approach enables the adaptation of new constraints in a flexible manner (see Figure 1) without redesigning the underlying algorithm.

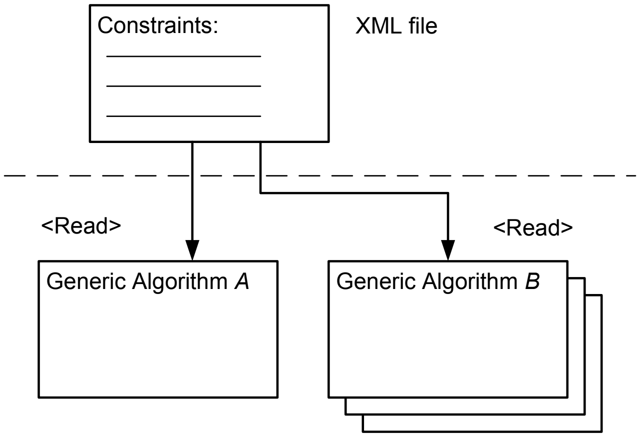

Figure 1. Constraint programming: The reuse of constraints in the algorithms and vice versa.

The CP paradigm provides a domain specific language to express the constraints. In our case it will be designed specifically for expressing data centre constraints. By using this language we can achieve the important goal of separating two different realms: (1) The realm of the data centre domain specific knowledge, expressed in the constraints, and (2) the realm of the optimization knowledge, expressed in the algorithms.

The optimizer aims at computing a configuration; an assignment of the VMs to the nodes; that minimizes the overall energy consumption of a federation of data centres while satisfying the different SLAs. In practice, the optimizer uses a power objective model to estimate the energy consumption of a configuration and extends Entropy, a flexible consolidation manager based on Constraint Programming, to compute the optimized configurations.

This paper is organized as follows: Section 2 introduces related work on the subject. Section 3 will present the proposed software architecture which includes the power objective model, heuristics search, and the translation of the SLA into constraints. The experimental results obtained in a cloud data centre testbed within Hewlett Packard premises as well as complementary scalability evaluation will be discussed in Section 4. Finally, we conclude the paper in Section 5 and discuss our future work in Section 6 .

2. RELATED WORK

In this section we briefly review recent middleware and frameworks for SLAs translation into constraints and heuristic-based approaches. Besides, we overview some work based on the CP paradigm, aimed at power saving in data centres.

2.1 SLA CONSTRAINTS

SLAs are referred to as the textual contract signed by a costumer and a service provider that guarantee a certain Quality of Service (QoS). In the context of data centre services these encompass, among other terms, hardware related descriptions, performance related metrics and availability guarantees as well as prizes and penalties. Besides the textual document written in natural language, SLAs are nowadays more and more coped with in a machine readable manner. Reasons for this can be seen in the increase of complexity as well as, driven by the development of agent based technologies, a much higher automation of the bargaining process.

However, one downside of this evolution is the rise of a set of highly heterogeneous technologies and standards. XML schemas, RDF and other ontological languages have been defined to be used in the context of all parts of the SLA life-cycle. For monitoring and billing, several middleware (e.g. Globus Toolkit, Unicore) and resource manager (e.g. PBSPRO, Torque, Maui) have been developed over the last decades. Furthermore frameworks capturing the whole SLA lifecyle (e.g. BrEIN, SLA@SOI, NextGrid SLA Framework) were created all implementing SLA capabilities in their own way. In order to be in the broadest sense platform independent with our approach, we therefore have to find a solution coping with this heterogeneity.

2.2 SEARCH HEURISTICS

The problem of consolidating and rearranging the allocation of virtual machines in a datacenter in an energy efficient manner is described in [15]. It is known to be a NP-hard problem [16] with a large solution space. Even if one suppresses any constraints from the problem, the size of the solution space is equal to the number of VMs to the power of involved servers, which is a huge number leading to a combinatorial explosion. For example, if the number of Servers is 10^3 and the number of VMs is 10^4, then the solution space without considering any constraints is 10^7.

In the heuristics proposed in [15], for each VM to be moved we find the appropriate server that leads to minimize the current overall power consumption of a data centre. This is similar to the First Fit Decreasing (FFD) algorithm which has been used in previous works [17][18][19], with the addition of power-awareness for choosing the server. While these types of heuristics are fast, in many situations they cannot lead to the optimal solution, unless the data centre is homogeneous. The heuristics are searching a solution by finding a local optimum for each VM, which is known to not always lead to a global optimum for the datacenter.

2.3 CP-BASED FRAMEWORKS

The framework presented in [20] addresses the Service Consolidation Problem (SCP) in a data centre using the CP approach. The rule-based constraints are assessed by the Comet programming language. This framework, however, focuses on the experimental evaluation of time necessary to find a feasible solution using CP and Integer Linear Programming (ILP) approaches. The obtained results show that the CP paradigm is more effective to find a solution for a large number of constraints and instances with respect to time.

In [13], CP is applied to solve the bin repacking scheduling problem. The main idea of this work is to schedule the transitions of VMs considering both placement constraints and resource requirements. In contrast to [13], we allow a user/operator to derive automatically the constraints starting from existing SLA requirements. Furthermore, the objective of saving energy is stated explicitly in the model used by our framework, by using a runtime simulation and evaluation of the energy consumption for every component of a data centre.

CP-based approaches were also proposed to solve the data migration [21] and load rebalancing [22] problems. However, all the listed works model the specific constraints directly.

The usage of constraint programming technology for SLA negotiation and validation has recently been investigated in a variety of approaches. The concurrent constraint Pi-calculus [11] provides mechanisms to negotiate and validate contracts by extending the nominal process calculi, for instance. Another approach was introduced by [10] extending the soft concurrent constraint language in order to facilitate SLA negotiation. However, the focus of all research is the negotiation process.

3. FRAMEWORK DESIGN

In this section, we first describe the global design of the framework. We then present the translation of SLAs into constraints, followed by the description of the power objective model and discuss about the heuristics we use to increase the scalability of our framework and the quality of the computed configurations.

The optimizer based on the CP engine has, in our case, several inputs:

The complete current data centre configuration.
A number of constraints, described in Section 3.2.
An objective function - in our case it is called the Power Objective, described in sub-section 3.3.
A number of Search Heuristics, described in sub-section 3.4.

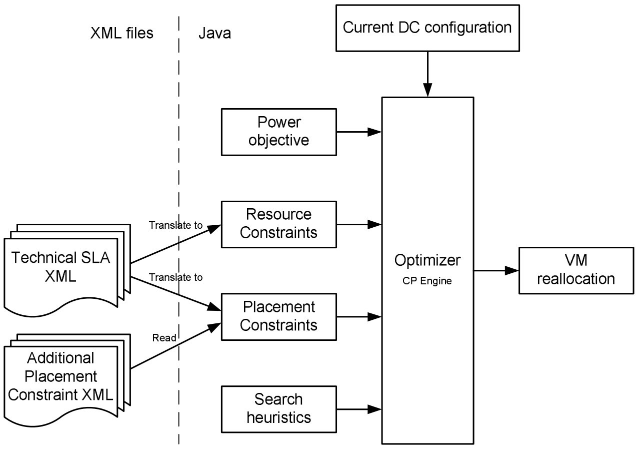

Figure 2. Framework architecture.

In the following, we outline the various input elements.

3.1 FRAMEWORK OVERVIEW

The framework extends the Entropy consolidation manager to compute an energy-efficient reconfiguration plan while Entropy itself is not energy-aware. It relies on the VM Repacking Scheduling Problem (VRSP), an abstract reconfiguration algorithm modeling the current memory and CPU demand of the VMs, the server's state and the future placement of the VMs. The VRSP can then be specialized to fit the datacenters and the VMs specificities.

The flexibility of Entropy comes from its usage of CP [24] to compute the new configuration and the reconfiguration plan. CP allows modeling and solving combinatorial problems where the problem is modeled by stating constraints (logical relations) that must be satisfied by its solution. Given sufficient time, the CP solving algorithm is guaranteed to determine a globally optimal solution, if one exists. The solving algorithm is independent of the constraints composing the problem and the order in which they are provided. This enables the framework to handle both the placement constraints and a power model independent from each other. In practice, Entropy embeds the CP solver Choco [25].

Figure 2 depicts the composition mechanism of the framework. Each call to the framework leads to 1) the generation of one Entropy VRSP based on the current configuration, 2) the translation and the injection of the external constraints, 3) the insertion of the power model, and 4) the insertion of the heuristics to guide the solver efficiently to a solution providing an optimized energy usage.

A timeout can be provided to Entropy to make it stop solving after a given time. When no timeout is specified, Entropy computes and returns the reconfiguration plan that lead to the best solutions according to the power model and the placement constraints. Otherwise, it returns the best solution computed so far.

3.2 SLA CONSTRAINTS

FIT4Green is implemented as a plug-in to extend the existing management framework. Thus it does not cope with SLA creation, bargaining, execution and validation per se. However, in order to not violate the SLAs during the optimization process, information need to be injected to the model. Thus, in our approach the aforementioned problem of technological diversity of different management frameworks needed to be dealt with, too. The solution was to define an XML schema on a low, technical level of abstraction. This in turn is being used by the DC operator to supply the needed constraints in a both human and machine readable format.

3.2.1 SLA Schema Creation

As a starting point (see Figure 3) for the definition of the schema we have used the findings of [9] where the authors have analyzed nearly fifty SLAs and extracted common metrics.

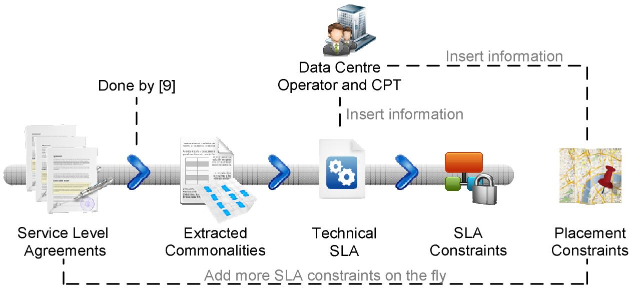

Figure 3. The development of constraints from natural language SLAs.

In a first step we deconstructed those high level Service Level Objectives (SLOs) concerning their impact on low level technical metrics. As a result we have identified four main categories:

Hardware-,
QoS related-,
Availability-, and
Additional metrics.

Within the first category all hardware related metrics like CPU frequency or RAM space is being captured.

The second category (QoS) encapsulates factors like the maximum amount of VMs that share a single CPU core, or the bandwidth. In modern data centres these metrics are often defined by the Capacity Planning Team (CPT) gaining their knowledge from past experience. Guaranteeing a certain service execution time for instance needs extensive knowledge about the process itself and the interplay with hardware resources. However, if past experience has shown, that the CPU is the bottleneck the CPT can decide to restrict the number of VMs per core. [14] has pointed out that automatic transformation of SLOs to technical SLAs are also possible in specific situations, eliminating the needed involvement of the CPT. This technique is in general also applicable in combination with our approach.

The availability of a service can in theory be used to shut down the services for specific time periods. However, in practice it heavily depends on the nature of the contract if a service provider really wants to make extensive use of these metrics. If service availability for instance is set to 99.9% the provider might not want to shut down the service for 0.1% of the time by purpose, as this might scare away his customers. Nevertheless, in a different scenario where a service can be shut down during weekends, the provider will certainly make use of it. Therefore, the third category was added to the XML schema.

Last, the category "additional metrics" contains, for example, guarantees concerning access possibility (e.g. VPN) or the guarantee of a dedicated server. Dedicated server in this context means that only one VM can be hosted on a server and is not related to a set of VMs. Whether a server supports a special access possibility is captured within the FIT4Green meta-model and is therefore handled as an 'attribute' of a server.

To conclude, the created technical XML schema can deal with all commonly known SLA metrics. However, as an addition we have created a second, low level placement schema with which the data centre operator can easily add "low level" constraints on the fly without writing a single line of code. It is based on the build-in placement constraints of entropy [13].

3.2.2 *Constraint Programming and SLAs*

In order to be used within Entropy, the technical constraints provided by the DC operator need to be translated. In general two approaches on different levels of abstraction can be used for this purpose:

The higher level placement constraints with a pre-selection process, and

The low level "posting"-method, which directly injects the rules and constraints to the Choco solver for consideration.

The first technique was used for most of the hardware related metrics. In a pre-selection process a set of servers have been extracted that satisfy the hardware requirements, which again is used in combination with the placement constraint 'fence' allowing an allocation only to be performed on this set of servers.

For the other metrics contained in the technical SLA, the low level "posting"-method is used in combination with the entropy model. It is more powerful and thus suitable for the creation of more complex constraints.

In Table 1 the different CP-approaches and their correlations with the technical constraints are presented. Besides, the number of lines of code needed for the implementation of each constraint is provided. Here, the number in brackets allegorizes the number of Lines of Code (LoC) needed to, on one hand transform the model used in FIT4Green to the one used in entropy, and on the other hand the LoC needed for the pre-selection process. Those generic methods are used for a variety of constraints and therefore listed separately.

Table 1. CP-Approach for Technical Constraints.

Category	Constraint	Approach	LoC
Hardware	HDD	Choco + ext. Entropy	121+(25)
	CPUCores	Entropy ('fence')	0+(25)
	CPUFreq	Entropy ('fence')	0+(25)
	RAM	Choco + ext. Entropy	123+(25)
	GPUCores	Entropy ('fence')	0+(25)
	GPUFreq	Entropy ('fence')	0+(47)
	RAIDLevel	Entropy ('fence')	0+(47)
QoS	MaxCPULoad	Choco + ext. Entropy	90+(25)
	MaxVLoadPerCore	Choco + ext. Entropy	109+(25)
	MaxVCPUPerCore	Choco + ext. Entropy	124+(25)
	Bandwidth	Entropy ('fence')	0+(49)
	MaxVMperServer	Entropy ('capacity')	0+(25)
Availability	PlannedOutages	Choco + ext. Entropy	Future Work
	Availability	Choco + ext. Entropy	Future Work
Additional Metrics	Dedicated Server	Entropy ('capacity')	0 + (25)
	Access	Entropy ('fence')	0 + (25)

3.3 POWER OBJECTIVE MODEL

As a basis for our model we use a component called "Power Calculator", which is also developed within the FIT4Green project and is being described in [15]. When provided with a description of the datacenter physical and dynamic elements, this component is able to simulate the power consumption of every part of the data center on a very fine level of granularity, in real time. While it is perfectly possible to call the Power Calculator component during the search of a reconfiguration plan, this has proven to be inefficient for our purpose. This is due to the complexity of the problem (NP-hard) as stated above. Here, we need to avoid calling the Power Calculator each time we are testing the placement of a VM in a server because this is very time consuming. As a result the CP engine must use a static version of the Power Calculator. This means that the necessary values are retrieved and stored in a vector before and not during the search and the engine has therefore all parameters directly "at hand".

In order to benefit from the fine granularity provided by the Power Calculator and at the same time gain from the advantages of CP programming we have used the following approach in our work.

In a first step we have grouped all servers s_i into families S_k that share similar characteristics, where $i \in I$ is the index of the server in the data centres, and $k \in K$ is the index of the family. The VMs v_i are also grouped into V_l families that share similar characteristics, where $j \in J$ is the index of the VM in the data centres, and $l \in L$ is the index of the family. Note that such an assumption is possible since it is common for a data centre to have families of similar equipment and because VMs often share similar run-time characteristics as well.

Furthermore we have defined a vector $H_i = <h_{i1}, \dots , h_{ij}, \dots , h_{in}>$ for each server s_i that denotes the set of VMs assigned to that server, where $h_{ij} = 1$ if the node s_i is hosting the VM v_j and 0 otherwise. The whole array H therefore represents how the VMs are assigned on servers in the different data centres. Now, the power consumed by server *I* depends on its physical components as well as on the set of VMs present:

$$P_i = f(s_i, H_i \quad v) \tag{1}$$

Here, *f* is the power consumption function provided by the Power Calculator. The dot is the vectorial product and *v* is the vector of all VMs. Thus, $H_i \bullet v$ is the vector representing all the VMs that are located on server *i*.

Next, we extend the function by a factor representing the fact that, if there are no VMs on a server, it can be switched off meaning that it is not consuming any energy any more. For this purpose let X_i be a variable which has a value of 1 if there is at least one VM in a server *i*, 0 otherwise:

$$X_i = \begin{cases} 1, \exists j \in J \mid h_{ij} = 1 \\ 0, otherwise \end{cases} \tag{2}$$

Then

$$P_i = X_i * f(s_i, H_i \quad v) \tag{3}$$

In the next step, the static version of the Power Calculator is included. Here, function *f* is split in two parts:

The calculation of the idle power of a server in the family S_k (i.e. power without any VM running), called $P_{idle}(S_k)$.

The calculation of the power consumed by a VM in the family V_l if the latter is running on a server in the family S_k, called PVM (S_k, V_l).

The idle power as well as the power per VM for each server can be computed before the search. Let *α* denote the vector of the idle powers of the families of servers, and *β* denotes the array of the power consumption per VM in each family:

$$\alpha_k = P_{idle}(S_k) \tag{4}$$

$$\beta_{kl} = P_{VM}(S_k, V_l) \tag{5}$$

Then we can obtain the static version of the server's power by using the following equation:

$$P_i = X_i * \alpha_k + \sum_{j \in J, l \mid v_j \in V_l} h_{ij} * \beta_{kl} \quad , k \mid s_i \in S_k \tag{6}$$

When P_0 is the power of the data centre before the execution of the plan, as computed by the Power Calculator, then the power saved is calculated as:

$$P_1 = \sum_{i \in I} P_i \tag{7}$$

$$P_{save} = P_0 - P_1 \tag{8}$$

As a last step to obtain the global energy figure of our solution, we need to integrate the cost of the network movements. For this purpose we first need to know which VMs are moving. This is done by subtracting the two matrixH_0 (initial state of the data centre) and H_1 (final state of the data centre), and analyzing the resulting matrix. We obtain a vector of the moves M_k = < (S_{from}, S_{to})$_{VM1}$, … , (S_{from}, S_{to})$_{VMk}$, … , (S_{from}, S_{to})$_{VMn}$>, where S_{from} and S_{to} are the source and destination servers of the VM, respectively and $k \in$ [1..*n*] is the index of the VM.

We can retrieve the energy cost of a move, providing the characteristics of the source server, the destination server and the VM from the power calculator. This cost includes the energy spent by moving the VM through the network, but can also include the overhead incurred in term of CPU load and RAM IO.

$$Emove_k = EnergyCost(s_i, s_j, v_k) \tag{9}$$

Here, *i* and *j* are the indexes of the source and destination servers of the VM v_k, respectively. We obtain the energy cost of the plan by summing the cost of every movement:

$$Emove = \sum_{k \in K} Emove_k \quad M_k \tag{10}$$

If we know the end time of a VM, we can compute its remaining life time (LT). This information can be combined with the cost of the network and equation (8), to get the total energy saving that we can expect by moving a VM:

$$E_k = (P_{0k} - P_{1k}) \times LT_k - Emove_k \tag{11}$$

The global energy saved by the plan, at federation level is therefore:

$$E_{total} = \sum_{k \in K} E_k \tag{12}$$

In practice, these energy formulas are written in the Choco modeling language within the "Power Objective" component of our framework.

3.4 HEURISTICS

As mentioned previously computing a solution for the VRSP using the Optimizer may be time consuming for large infrastructures as selecting a satisfying server for each running VMs while maximizing the infrastructure energy efficiency is a NP-Hard problem. A CP solver, such as Choco, provides a customizable branching heuristic to guide the solver to a solution. A branching heuristic indicates an order to instantiate the variables and a value to try for each variable. For a given problem, the branching heuristic helps the solver by indicating variables that are critical to compute a solution and values that are supposed to be the best. A branching heuristic is then highly coupled with the exact objective of the problem as it relies to the variables semantic, an information that is initially out of the CP solver concern. For the Optimizer, the branching heuristic helps at instantiating the variables in a priority descending order denoting the VM placement to a value that point to an energy-efficient server. In practice, VMs are sorted in the increasing order of their energy efficiency and the solver will try to place each VM to a server that will provide the best energy gain. The energy gain is provided by the variable E_k described in the last section. As this metric includes both the energy cost of the VM on its destination server and the energy cost related to its migration, this approach tends also to reduce the number of migrations to a minimum to provide a fast reconfiguration process.

4. FRAMEWORK EVALUATION

In this section we first evaluate the energy saving due to our approach on a cloud testbed hosting workload inspired by a corporation. We then evaluate the scalability of the Optimizer.

4.1 Experiments on Cloud Testbed

In order to validate the proposed approach in an environment as close as possible to a cloud data centre, a trial has been performed at Hewlett Packard (HP) Italy Innovation Center facilities, inside the Cloud Computing Initiative lab environment. The facility is used to offer "hands-on" experience on a cloud demo infrastructure and to setup Proof of Concepts (PoC) configurations.

Two different workloads have been setup for an Infrastructure-as-a-Service private cloud: the first one simulates a typical week load pattern and a second one – more challenging – focuses on a single work day.

4.1.1 Lab trial resources

Inside HP Italy Innovation Center, two racks, with an HP C7000 blade enclosure each, have been used to simulate two separate data centres; the first one (DC1) has 4 BL 460c blades dedicated to host Virtual Machines using VMWare ESX v4.0 native hypervisor, and 3 additional blades for Cluster and Cloud Control, and the scheduler of the workload tasks (VM creation and load generation). The second one (DC2) hosts 3 BL460c blades to host Virtual Machines again using VMWare ESX v4.0 native hypervisor, and 2 other blades for Cluster Control and the Data Collector of the Power and Monitoring System. The racks are connected to a LAN and use a SAN device to store all data, including VM images. The Virtual Connect modules inside the Blade enclosures offer a fast internal 1GB network.

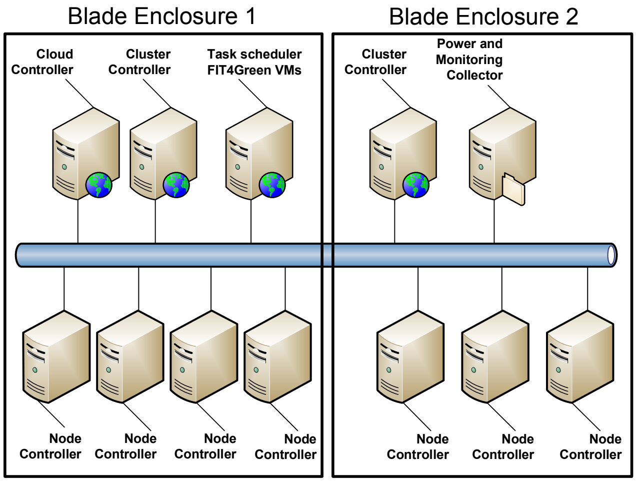

Figure 4. The logical view of the main hardware resources.

The characteristics of the processors in the two racks/enclosures are listed in Table 2.

4.1.2 Workload

The system has been tested using two synthetic workloads built to reply with high accuracy to the load patterns recorded in a real case PoC. The next figure represents the pattern of total number of active virtual machines during full week of work inside a small-medium size private cloud used by a corporation in Italy during a Proof of Concept performed with HP Innovation Center in Italy. The first synthetic test reproduces the 7 days compressed in time into 24 hours, while the second one focuses only on a single work day has been reproduced in 12 hours. The following picture schematically describes the weekly load pattern (number of active VMs on the Y axis) and the red box identifies the single work day for the second test.

Table 2. Characteristics of the Racks/Enclosures.

	Enclosure 1	Enclosure 2
Processor model	Intel Xeon E5520	Intel Xeon E5540
CPU frequency	2.27GHz	2.53GHz
Cpu& Cores	Dual cpu – Quad core	Dual cpu – Quad core
RAM	24 GB	24GB

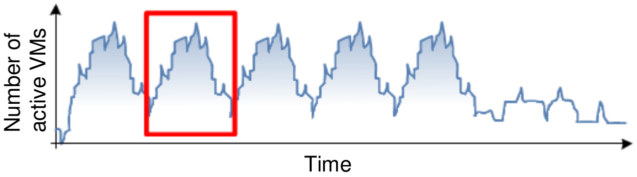

Figure 5. Schematic View on the Weekly load patterns.

The first workload considers also week-ends with low load, but it's an interesting approximation of a real case; the second one is more challenging and therefore it has been used more extensively in the federated trial. The workload execution is performed through an open source scheduler application (Task Scheduler – running inside a VM on the blade server in DC1), while system and power monitoring is performed through open source Collectd (inside another VM on blade server in DC2).

The explicit SLAs configured in the trial – and mapped into constraints – are related to the Number of Virtual CPUs per Core ratio (2 in the trial) and to the description of the topology of nodes in the federation. The parameters related to policies applied to the data centres are the same on each clusters: i.e. always guarantee at least 3 free 'VM slots', where a 'VM slot' is the necessary amount of free resources to accommodate a new VM, and keep at most 6 VM slots.

4.1.3 Single Site Trial

The trial for a Single Site scenario has been performed using only the first rack (DC1) and both workloads. The Task Scheduler allocates VMs through Cloud and Cluster Controller primitives only the nodes in DC1; data collection for power and system monitoring runs on a blade server in DC2. Table 3 shows the results in terms of overall energy consumed by the node controllers (the servers with native hypervisors where VMs are allocated); due to the lab-grade configuration, the number of cloud control servers (cloud and cluster controller, monitoring and scheduler) wrt. node controllers is far too high compared to a real cloud environment, therefore cloud control servers have been omitted from the computation to allow a clearer interpretation of the results. For test 1, the energy data refer to the average consumption per day of 4 node controllers inside DC1.

Table 3. Single Site Trial.

Scenario	Average Day for Week Workload	Single work day workload
Without FIT4Green	6029 Wh	6621 Wh
With FIT4Green – no migration	4867 Wh saving 19.2%	5938 saving 10.3%
With FIT4Green – using migration	4592 Wh saving 23.8%	5444 saving 17.7%

The trial shows an energy saving of approximately 24% in the average week workload, and almost 18% for the week day workload. As expected the second workload is more challenging than the first one, moreover the effect of VM migration capability for the optimization strategy is very important, especially in the most critical case.

After the runs, the monitored system data are analyzed to double check that the specified SLAs have not been violated.

4.1.4 Federated Sites Trial

The trial for the federated case has been performed using a data centre hosting one cluster of 4 nodes, and the second data centre hosting another independent cluster of 3 nodes. The workload for the first data centre is the same one as the single work day test of the single site case, while the workload for the second data centre is scaled by a factor ¾ (to cope with the smaller amount of computing resources) and has its peak shifted in time of approx. 1/24 in the time scale (1 hour in the 24 hours scenario) to simulate a slight work-time differences of the users of the second data centre.

Results have been collected in different configurations:

Without FIT4Green with independent allocation of the workload on the two DCs (clusters); each workload item has been statically pre-assigned to one cluster

With FIT4Green with independent allocation of the workload on the two DCs (clusters)

With FIT4Green with dynamic allocation of the workload on the two DCs (clusters); when a workload item needs to be started FIT4Green is queried to decide on which cluster to run it

FIT4Green with dynamic allocation of the workload on the two DCs (clusters) and optimized policies; in this case the "buffer" of free slots of each cluster has been reduced capitalizing on the availability of additional resources in the other cluster – practically the minimum VM slot number has been reduced to 2 and the maximum to 5 on each cluster because the VM allocation can be satisfied by any one of the clusters

Table 4 presents, for the different configurations, the numerical results in term of global energy consumed by each datacenter node controllers (cluster nodes) and the total for the federation.

In the case of FIT4Green Static Allocation each data centre is considered separately, in the next case allocation is decided based on the energy saving optimizations. The ability to use the federation as a single pool of resources at allocation time allows saving to grow from 16.7% to 18.5%. Moreover the tuning of policies reducing the free amount of resources (min. VM slots) to be kept free to cope with load peaks at cluster level, allows saving to grow up to 21.7%.

Table 4. Federated Sites Trial.

Configuration	Data Centre 1	Data Centre 2	Energy for Federation
Without FIT4Green	6350 Wh	4701 Wh	11051 Wh
With FIT4Green Static Allocation	5190 Wh	4009 Wh	9199 Wh Saving 16.7%
With FIT4Green Dynamic Allocation	5068 Wh	3933 Wh	9001 Wh Saving 18.5%
With FIT4Green Optimized Policies	4860 Wh	3785 Wh	8645 Wh Saving 21.7%

4.1.5 Energy vs. Emissions Optimization

In the previous tests the two data centres were assumed to have exactly the same characteristics in terms of energy and emissions efficiency (as in the reality, since they're co-hosted in the same site). The goal is to evaluate the effectiveness of the optimizer when dealing with a federation of data centres heterogeneous with respect to energy and emissions efficiencies.

In order to simulate the scenario with data centres with different energy and emissions features, the last test has been run in two additional work modes, by modifying the meta-model Power Usage Effectiveness (PUE) and Carbon Usage Effectiveness (CUE) attributes of the data centre configuration:

DC1 with PUE=2.1 and DC2 with PUE=1.8 (more efficient) – optimizing for total energy of the federation

DC1 with CUE=0.772 g/Wh and DC 2 with CUE=0.797 g/Wh – optimizing for to total emissions of the federation; the two values for CUE simulate DC1 getting energy by Enel at 443 CO2 g/kWh and DC2 gets energy by A2A at CO2 368 g/kWh.

Figure 6 reports the final test results for the various configurations in visual format, while Table 5 contains the corresponding numerical values.

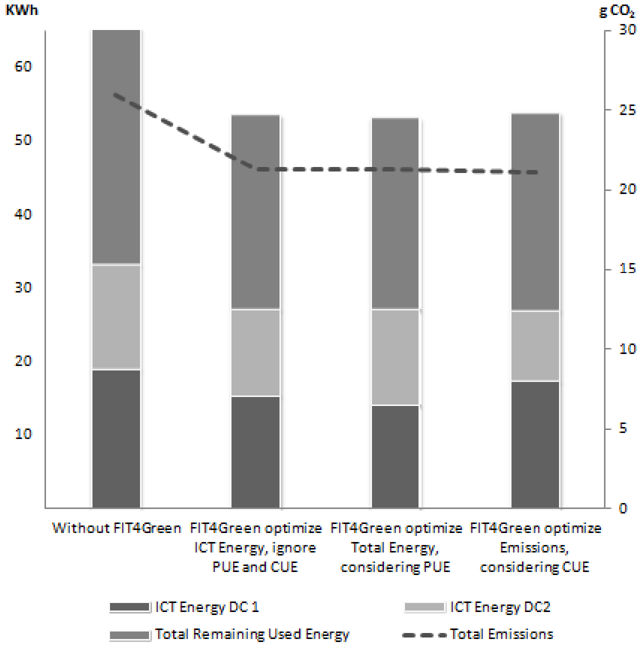

Figure 6. Graphical Representation of the Trial Results.

It's worth to notice that when FIT4Green optimizes for Energy, it saves 0.6% more in addition to the ICT energy optimization, because it's capitalizing on the energy efficiency difference of the two data centres, by relatively loading more DC2 that has a better PUE value.

When optimizing for Emissions, DC1 is relatively more loaded because it has better emissions efficiency (lower value of CUE) and the total improvement is 0.6% better than the ICT energy optimization case.

Table 5. Energy vs. Emissions.

Configuration	ICT Energy DC 1	ICT Energy DC2	Total Energy	Total Emissions
Without FIT4Green	19050 Wh	14103 Wh	65390 Wh	25.94 g CO2
FIT4Green optimize ICT Energy, ignore PUE and CUE	15486 Wh	11663 Wh	53514 Wh Saving 18.16%	21.25 g CO2 Saving 18.10%
FIT4Green optimize Total Energy, considering PUE	14188 Wh	12953 Wh	53110 Wh Saving 18.78%	21.27 g CO2 Saving 17.99%
FIT4Green optimize Emissions, considering CUE	17381 Wh	9624 Wh	53823 Wh Saving 17.68%	21.08 g CO2 Saving 18.72%

4.2 Scalability Evaluation

In order to show the correctness of our approach with a high number of servers and VMs, we made experimentation in simulation as a complement to the experimentation done within HP premises. Indeed, while the experimentation has been done on real equipment for a low number of servers, high scale experimentation can only be done through simulation from a practical point of view.

The simulation has been run using a DELL Latitude E6410 laptop with an Intel i7 Dual Core processor at 2.67GHz and 4GB of RAM. For the simulation we have varied the number of servers, with each server having 1 CPU with 4 cores at 1GHz, 8GB of RAM and 4 virtual machines instances already activated on it.

Each VM has 1 Virtual CPU used at 70%. The memory used by the VMs is set to 100MB. For each simulation run, we measured the time taken by the search to find a first solution and verified it for all the VMs and given the constraints. We have repeated the experiment 3 times: with one datacenter and no placement constraints, with one datacenter with an overbooking factor constraint set to 2, and with two federated datacenters.

In Table 6 the placement constraints activated to realize each configuration is detailed. With one datacenter, no placement constraint is activated: the VMs are free to move in the datacenter, they just need to respect the default constraints that enforces that a valid configuration is found with respect to the consumption of the VM in term of CPU, RAM and HDD and the available resources on the servers. The "overbooking factor" set to 2 corresponds to a constraint called "MaxVCPUPerCore", which enforces that no more than 2 virtual CPU is attributed to one core. The "2 federated datacenters" configuration is translated into « Fences » constraints disallowing the VMs to migrate from one datacenter to another, which is usually not feasible in practice.

Table 6. Constraints activated in each configuration.

#	Configuration	Placement constraints activated
1	1 datacenter	none
2	1 datacenter with overbooking factor=2	"MaxVCPUPerCore" constraint set on each server
3	2 federated datacenters	"Fence" constraint set on each VM

Table 7. Solving duration of the Optimizer to compute the first solution.

number of servers	1 datacenter (ms)	1 datacenter with Overbooking factor=2 (ms)	2 federated datacenters (ms)
25	301	401	200
50	801	701	501
100	2504	1802	901
200	14214	10644	2810
250	25727	18718	4504
300	44152	29533	7004
400	87443	64095	14757
500	162207	120885	26630
600	269487	194193	42409
700	403893	307067	63671

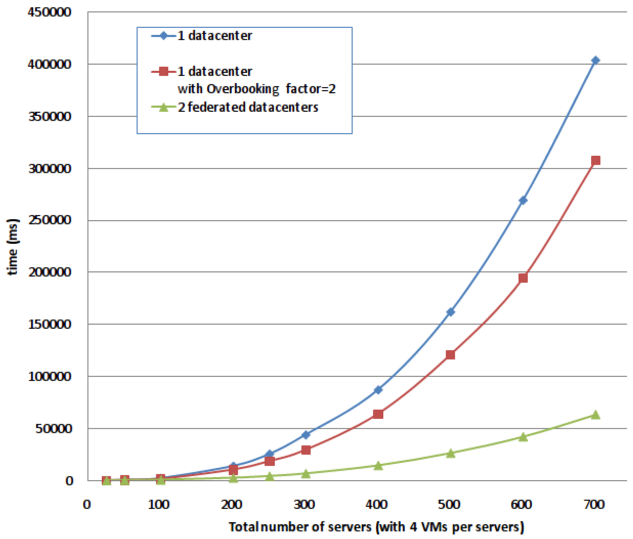

Figure 7. Graphical representation of the Optimizer' solving duration to compute the first solution.

First of all, the results presented in Table 7 show that for 700 servers and 2800 VMs the search completes in 6.7 minutes on the worst case. If the servers are split in two datacenters, the time drops to nearly 1 minute. Interestingly enough, adding new constraints don't increase the search time as one could expect: globally the search times are inferior with the placement constraints activated to the ones without. This is because adding new constraints, while it adds a small overhead in processing the constraint, also greatly reduces the problem search space. This shows that the engine prunes incorrect sub-trees in the search tree using the new constraints. For example in the Table 7 we see that the time for 400 servers split in 2 DC (14757ms) is nearly equal to the time in single DC for 200 servers (14214 ms). The times show that the engine is effectively separating the problem in 2, and that the two are then computed in parallel. The little time difference may be due to the overhead of parallel computing and the slightly increased time for the preparation the problem. The result would be the same if the VMs are separated in two clusters in the same data centre, which is also a common practice. The addition of the overbooking constraint reduces also the time, for the same reasons.

5. CONCLUSION

In this paper we have presented an approach for energy-aware resource allocation in datacenters using constraint programming. We addressed the problem of extensibility and flexibility by decoupling constraints and algorithms. Using this feature and easy-to-extend XML schemas, we were able to implement 16 frequently used SLA parameters in the form of constraints. The results of the tests executed in a cloud environment have shown that the presented approach is capable of saving both a significant amount of energy and CO_2 emissions in a real world scenario on average 18%within our test case. Furthermore, our scalability experiment showed that splitting the problem in several parts to enable parallel computation is very efficient in reducing the total computation time to find a solution. Indeed, we were able to find the first allocation solution for 2800 VMs in 700 servers split in two clusters in approximately 1 minute.

6. FUTURE WORK

Encouraged by the results of our test we will continue our research in this area. One enhancement will address the research in the area of SLAs. Even though current metrics do not directly relate to energy saving or environmental metrics, they play a major role in the process of energy saving strategies. As mentioned in [12] a key in lowering the energy consumption in data centres, without replacing hardware- or infrastructural-components, is to tweak SLAs in a way that guarantee the needed QoS for the customer, but at the same time widening the range of flexibility for the data centre operator to apply certain energy saving strategies. For that reason, in order to apply this approach the fixed structures of current SLAs either need to be enhanced by the possibility to express preferences in a "fuzzy" manner or to use a dynamic, preferable autonomous, re-negotiation process by using software agents, for instance (www.all4green-project.eu). In the context of FIT4Green we do neither replace a complete data centre management framework nor postulate agent based SLA negotiation. Therefore, the first approach is more appropriate. In the context of our framework this concludes in the extension of entropy to use so called soft constraints. In addition Klingert et. al. in [12] mention the need for new 'green' metrics. In the current state the entropy library provides only a limited model of the data centre infrastructure and VMs. Therefore, we will additionally explore the needs of new 'green' metrics in a technical aspect.

Besides the consideration of GreenSLAs we plan to investigate new heuristics and algorithms to first improve the efficiency of the optimizer and second the quality of the proposed solutions. We also plan to extend the concepts developed in this paper to other components involved in delivering an Internet service, suc as the network.

ACKNOWLEDGMENTS

This research has been partly (Corentin Dupont, Giovanni Giuliani, Thomas Schulze, Andrey Somov) carried out within the European Project FIT4Green (FP7-ICT-2009-4). Details on the project, its goals and results can be found at: http://www.fit4green.eu.

The authors would also like to thank Marco Di Girolamo (HP Italy Innovation Centre, Milan) for his valuable comments and fruitful discussions.

REFERENCES

[1] Berral, J. L., Goiri, I., Nou, R., Julia, F., Guitart, J., Gavalda, R., and Torres, J. 2010. Towards energy-aware scheduling in data centers using machine learning. In *Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking* (Passau, Germany, April 13-15, 2010). e-Energy'10. ACM, New York, NY 215-224. DOI=10.1145/1791314.1791349

[2] Green Grid Consortium, http://www.thegreengrid.org

[3] Banerjee, A., Mukherjee, T., Varsamopoulos, G., Gupta, S. K. S.. 2010. Cooling-aware and thermal-aware workload placement for green HPC data centers. In *Proceedings of International Green Computing Conference*(Chicago, IL, USA, August 15-18, 2010). 245-256. DOI=10.1109/DREENCOMP.2010.5598306.

[4] Pakbaznia, E. and Pedram, M. 2009. Minimizing data center cooling and server power costs. In *Proceedings of the 14th ACM/IEEE International Symposium on Low Power Electronics and Design* (San Francisco, CA, USA, August 19-21). ISPLED'09. ACM, New York, NY 145-150.DOI = http://doi.acm.org/10.1145/1594233.1594268

[5] Meisner, D., Gold, B. T., and Wenisch, T. F. 2009. PowerNap: Eliminating server idle power. In *proceedings of the 14th International Conference on Architectural Support for Programming Languages and Operating Systems* (Washington, DC, USA, March 7-11, 2009). ASPLOS'09. ACM, New York, NY 205-216. DOI = 10.1145/1508244.1508269

[6] Carrol, R., Balasubramaniam, S., Donnelly, W., and D. Botvich. 2011. Dynamic optimization solution for green service migration in data centres. In *Proceedings of IEEE International Conference on Communications* (Kyoto, Japan, June 5-9, 2011). ICC'11.pp. 1-6, 2011. DOI = 10.1109/icc.2011.5963030.

[7] Garg, S. K., Yeo, C. S., Anandasivam, A., and Buyya, R. 2010. Environment-conscious scheduling of HPC applications on distributed cloud-oriented data centers. *Journal of Parallel and Distributed Computing.* 71 (2010), 732-749.

[8] Barbagallo, D., Nitto, E., Dubois, D. J., and Mirandola, R. 2010. A Bio-inspired algorithm for energy optimization in a self-organizing data center. In *Proceedings of the FirstSelf-organizing architectures* (Cambridge, UK). SOAR'09. Springer-Verlag Berlin, Heidelberg, 127-151. ISBN:3-642-14411-X 978-3-642-14411-0.

[9] Paschke, A., Schnappinger-Gerull, E. 2006. A categorization scheme for SLA metrics.In *Proceedings of Service Oriented Electronic Commerce*, Vol 80, p. 25-40

[10] Bistarelli, S., Santini, F. 2008. A nonmonologic soft concurrent constraint language for sla negotiation, Proc. CILC'08

[11] Buscemi, M.,Montanari, U. 2007. Cc-pi: A constraint-based language for specifying service level agreements. In *Proceedings of the 16th European conference on programming.* (Braga, Portugal). ESOP'07. Springer Verlag Berlin, Heidelberg 18-32. ISBN: 978-3-540-71314-2

[12] Klingert, S., Schulze, T., Bunse, C. 2011. GreenSLAs for the Energy-effcient Management of Data Centres. In *Proceedings of the Second International Conference on Energy-effcientComputing and Networking* (New York, USA, May 31-June 1, 2011). e-Energy'11. ACM, New York, NY.

[13] Hermenier, F.,Demassey, S., Lorca, X. 2011. Bin repacking scheduling in virtualized datacenters. In *Proceedings of the 17th International Conference on Principles and Practice of Constraint Programming* (Perugia, Italy). CP'11. Jimmy Lee (Ed.). Springer-Verlag, Berlin, Heidelberg, 27-41.

[14] Chen, Y.,Iyer, S., Liu, X.,Milojicic, D., Sahai, A. 2007.SLA decomposition: Translating service level objectives to system level thresholds. In *Proceedings of the Fourth International Conference on Autonomic Computing* (Washington, DC, USA, 2007). ICAC'07. IEEE Computer Society. DOI=10.1109/ICAC.2007.36.

[15] Quan, D.-M.,Basmadjian, R., De Meer, H., Lent, R.,Mahmoodi, T.,Sannelli, D.,Mezza, F.,Dupont, C. 2011.Energy efficient resource allocation strategy for cloud data centres.In *Proceedings of the 26th International Symposium on Computer and information Sciences* (London, UK, September 26-28, 2011). ISCIS'11.Springer, 133-141.

[16] Lawler, E. 1983. Recent results in the theory of machine scheduling. In *Mathematical Programming: The State of the Art.* Springer-Verlag, Berlin, Germany.

[17] N. Bobroff, A. Kochut, and K. Beaty. Dynamic placement of virtual machines for managing SLA violations. Integrated Network Management, 2007. IM '07. 10th IFIP/IEEE International Symposium on, pages 119–128, May 2007.

[18] Wood, T.,Shenoy, P. J.,Venkataramani,A.Yousif, M. S. 2007.Black-box and gray-box strategies for virtual machine migration. In *Proceedings of the 4th ACM/USENIX Symposium on Networked Systems Design and Implementation* (Cambridge, MA, USA). NSDI '07. USENIX Association, Berkeley, CA, USA,17-17.

[19] Verma, A.,Ahuja, P.,Neogi, A. 2008. Power-aware dynamic placement of hpc applications. In *Proceedings of the 22nd Annual International Conference on Supercomputing*(Island of Kos, Greece).ICS '08. ACM, New York, NY,175–184. DOI=10.1145/1375527.1375555.

[20] Dhyani, K., Gualandi, S., Cremonesi, P. 2010.A Constraint programming approach for the service consolidation problem. In *Lecture Notes of Computer Science,* 6140(2010). Springer, 97-101. DOI=10.1007/978-3-642-13520-0-13.

[21] Anderson, E., Hall, J., Hartline, J., Hobbes, M., Karlin, A., Saia, J., Swaminathan,R., Wilkes, J. 2010. Algorithms for data migration.*J.Algorithmica*, 57(2), 349–380.

[22] Fukunaga, A. 2009. Search spaces for min-perturbation repair. In*Proceedings of the 15th International Conference on Principles and Practice of Constraint Programming* (Lisbon, Portugal). CP'09. Springer Verlag Berlin, Heidelberg, 383–390.

[23] FIT4Green EU Project, `http://www.fit4green.eu`

[24] Rossi, F., Van Beek, P., Walsh, T. 2006.*Handbook of Constraints Programming*. Elsevier Science Inc.

[25] Choco,`http://choco.emn.fr`