Power-Aware Server Consolidation for Federated Clouds

Alessandro Ferreira Leite, Azzedine Boukerche, Alba Cristina Magalhaes Alves de Melo, Christine Eisenbeis, Claude Tadonki, Célia Ghedini Ralha

Power-Aware Server Consolidation for Federated Clouds

Alessandro Ferreira Leite[1],Azzedine Boukerche[2*],Alba Cristina Magalhaes Alves de Melo[1],Christine Eisenbeis[3,4],Claude Tadonki[5], Célia Ghedini Ralha[1]

[1]*Department of Computer Science, University of Brasilia, Brasilia, Brazil*
[2]*School of Information and Technology Engineering, University of Ottawa, Ottawa, Canada*
[3]*INRIA, Saclay, France*
[4]*Université Paris-Sud, France*
[5]*École Nationale Supérieure des Mines Paris, Fontainebleau, France*

SUMMARY

Cloud computing has evolved to provide computing resources on-demand through a virtualized infrastructure, letting applications, computing power, data storage, and network resources to be provisioned and managed over private networks or over the Internet. Cloud services normally run on large data centers and demand a huge amount of electricity. Consequently, the electricity cost represents one of the major concerns of data centers, since it is sometimes nonlinear with the capacity of the data centers, and it is also associated with a high amount of carbon emission (CO_2). However, energy-saving schemes that result in too much degradation of the system performance or in violations of service-level agreement (SLA) parameters would eventually cause the users to move to another cloud provider. Thus, there is a need to reach a balance between energy savings and the costs incurred by these savings in the execution of the applications. Therefore, in this paper we propose and evaluate a power and SLA-aware application consolidation solution for cloud federations. It comprises a multi-agent system (MAS) for server consolidation, taking into account service-level agreement, power consumption, and carbon footprint. Different for similar solutions available in the literature, in our solution, when a cloud is overloaded its data center needs to negotiate with other data centers before migrating the workload to another cloud. Simulation results show that our approach can reduce up to 46% of the power consumption while trying to meet performance requirements. Furthermore, we show that federated clouds can provide an adequate solution to deal with power consumption in the clouds.

KEY WORDS: Server Consolidation; Federated Clouds; Power-awareness Computing

1. INTRODUCTION

In the 1960s, time-sharing pushed up the development of computer networks [29]. Nowadays, cloud computing is being seen as the new time-sharing [14] due to characteristics such as on-demand, pay-per-usage, and elasticity. In addition, it has pushed down the cost the users pay for having their own large-scale computing infrastructures, and removed the need for up-front investments to establish these computing infrastructures. In other words, cloud computing has enabled a utility computing model, offering computing, storage, and software as a service.

This utility model yields the notion of resource democratization and provides the capability for a pool of resources accessible to anyone on the Internet in nearly real-time. This notion is the main difference between cloud computing and other paradigms (e.g., grid computing systems) that have tried to deliver computing resources over the Internet before cloud [66]. Besides that, clouds offer

services without knowing who the users are and unaware of the technology they use to access the services [23, 9]. Finally, cloud computing has also gained popularity since it can help data centers to reduce monetary costs and carbon footprint.

However, cloud's services usually run in big data centers, which contain a large number of computing nodes. The projections considering data centers' energy-efficiency [36, 27, 37] show that the total amount of electricity consumed by data centers in the next years will be extremely high, and it is like to overtake the airlines industry in terms of carbon emissions.

Additionally, depending on the efficiency of the data center infrastructure, the number of watts that it requires can be from three to thirty times higher than the number of watts needed for computations [52]. And it has a high impact on the total operation costs [4], which can be over 60% of the peak load. Nevertheless, energy-saving schemes that result in too much degradation of the system performance or in violations of service-level agreement (SLA) parameters would eventually cause the users to move to another cloud provider. Hence, reducing the energy consumption without sacrificing service-level agreement (SLA) is an important issue for economical reasons and also for making the data centers environment sustainable [7].

The power consumption of a system comprises two parts: (a) a static or leakage power and (b) a dynamic part. The leakage power depends on the system size and on the type of transistor. This power consumption is related to the leakage currents that are present in any powered system and it is independent of clock rates and usage scenarios. The dynamic part, on the other hand, depends on the activity of a circuit, the usage scenario, and the clock rates. This consumption is mainly composed of short-circuit current and switched capacitance [6, 55]. Energy consumption, on the other hand, is defined as the average power consumption over a period of time. Clearly, energy consumption and power consumption are closely related but it is easy to see that the reduction in power consumption does not necessarily imply the reduction in energy consumption [6].

Actually, there are two basic approaches to reduce power consumption in a data center. The first one comprises the method of power-aware hardware design, which can be carried out at various levels, such as device-level power reduction, circuit and logic level intelligent power management and architecture power reduction. The second one is the method of power-aware software design, known as dynamic voltage and frequency scaling (DVFS) [15, 54], including the operating system, the applications, and the resources allocation in general.

Therefore, many studies have been conducted to provide power reduction, and some of them are based on server consolidation [6]. Server consolidation combines several virtual machines into a single physical server, trying to minimize the number of physical servers required to host a group of virtual machines. By employing server consolidation, data centers can consolidate the workloads into fewer nodes and switch off unused ones or put them in a low power consumption state mode. Of course, the effectiveness of this technique depends on (a) how saturated the cloud system is and (b) how much slowdown the cloud applications will accept. If the infrastructure is not saturated, VMs can be moved to hosts that are close to each other and power efficiency gains can be obtained with small application slowdowns. However, if the infrastructure is saturated (i.e., there are very few nodes with idle capacity), the power efficiency gains provided by server consolidation will come at the expense of severe performance loss in the applications since, in this case, several virtual machines may share the same physical core.

In order to support a large number of users or to decentralize the management, clouds can be combined, forming a federated cloud environment. A cloud federation can be defined as a set of cloud providers, public and private, connected through the Internet [8, 13] to offer non trivial quality of service (QoS) services for the users, based on standard interfaces, and without a centralized coordination [22]. A federated cloud can move services and tasks among clouds in order to achieve its goals. These goals are usually described as QoS metrics, such as minimum execution time, minimum price, availability, minimum power consumption and minimum network latency, among others.

Another important issue in large data centers is the level of carbon emissions. The high amount of carbon emissions in data centers is associated with the amount of energy consumption. A recent study on cloud computing and climate change shows that the total electricity consumed by data

centers in 2020 will be $1,963$ billion kWh and the carbon emissions associated would reach $1,024$ megatonnes [27]. In this scenario, strategies that are aware of energy in terms of resource are gaining attention in academy and industry, since they can fulfill the promise of developing network applications that can be tuned to consume less energy [43].

Hence, saving power of large-scale systems with acceptable performance losses is an economical incentive for their operators, as well as a significant contribution to the environment. However, this requires the design of power-aware solutions. Power-awareness can be characterized by taking into account the amount of resources, the performance constraints of the applications, and the power requirements along their life cycle [43].

In this context, we have distinct participants with multiple objectives, preferences, and disposition to pay for services. In this scenario, a multi-agent system (MAS) may be used where each participant is an autonomous agent that incorporates market and negotiation capabilities. Agents are autonomous, proactive, and trigger actions by their own initiative [61]. They also make decisions by themselves according to their beliefs, desires and intentions. For these reasons, agents are suitable for coordinating the cloud market, detecting problems, opportunities and reacting to them, triggering the most appropriate action depending on the system status. This capability can be used to negotiate resources usage, and to negotiate power consumption and carbon emission.

Therefore, in this work, we propose the use of a multi-agent system (MAS) for federated cloud server consolidation taking into account service-level agreement (SLA), power consumption, and carbon footprint. In our approach, the users should pay according to the efficiency of their applications in terms of resource utilization and power consumption. Therefore, we propose that the price paid by the users should increase according to the whole energy consumption of the data center, specially when the users refuse to negotiate performance requirements. Experimental results show that our approach can reduce up to 46% of the power consumption, while meeting QoS requirements. The experiments were realized through the CloudSim [10] simulator with two clouds and 400 simultaneous virtual machines.

The remainder of this paper is organized as follows. Section 2 presents some concepts of cloud computing. Section 3 discusses energy-aware strategies and energy green performance indicators. Section 4 presents some related work in the area of multi-agent server consolidation. The proposed multi-agent system for federated cloud server consolidation is presented in Section 5. In Section 6, experimental results are discussed. Finally, Section 7 presents final considerations and future work.

2. CLOUD COMPUTING

Many cloud computing definitions have been proposed over the last years. One reason for the existence of different perceptions about cloud computing is that cloud computing is not a new technology, but a new operational model that brings together a set of existing technologies in a different way [57, 64]. Therefore, cloud computing can be seen as a type of distributed system that dynamically provisions virtualized elastic and on-demand resources, respecting service-level agreements defined between the service provider and the consumers [23, 9]. It differs from traditional distributed computing systems because it can be encapsulated as an abstract entity that delivers different levels of services to customers outside the cloud and that is driven by economies of scale, dynamically configured with on-demand delivery model [42].

Several cloud architectures have been proposed in the literature. Generally, the architecture of a cloud computing system can be divided into four layers: hardware, infrastructure, platform and application layers.

The hardware layer contains the physical resources of the cloud, such as CPUs, disks and networks. It is usually confined in data centers which contain thousands of servers and storage systems interconnected by switches.

The infrastructure layer contains resources that have been abstracted typically by using virtualization techniques, creating a pool of computing resources to be exposed as integrated resources to the upper layer and end users [23]. This layer is an important component of cloud computing, since many features such as elastic resource assignment are made available in this layer [64].

On the one hand, the platform layer consists of application frameworks and a collection of specialized tools on top of the infrastructure layer to provide a development and/or deployment platform aiming to minimize the burden of deploying applications directly into virtual machines containers [23, 64].

On the other hand, the application layer contains the applications that run in the clouds. Different from traditional applications, cloud applications can leverage on automatic scaling to achieve better performance and availability in an on-demand usage.

Furthermore, a cloud service model is mapped to the cloud architecture. In such case, the cloud service model comprises three classes, defined according to the abstraction level and the service model of the providers. The classes are: infrastructure-as-a-service (IaaS), platform-as-a-service (PaaS), and software-as-a-service (SaaS) [42]. The main difference between these cloud service models relies on the kind of control that the users may have over the cloud infrastructure.

In the traditional approach (i.e., non-cloud scenario), the users are responsible for managing the whole stack (e.g., hardware, software, and data center facilities), which gives them full control over the infrastructure.

In the infrastructure-as-a-service (IaaS) model, the users request processing power, storage, network, and other computing resources such as the operating system and pay for what they use. In other words, the users pay for the use of resources, instead of having to setup them, and deploy their own software on physical machines, controlling and managing them. The amount of instances can be scaled dynamically to fill the users' need. Examples of IaaS providers are Amazon Elastic Compute Cloud (Amazon EC2)–(*aws.amazon.com/ec2*), Rackspace cloud (*rackspace.com/cloud*), GigaSpaces (*gigaspaces.com*), Microsoft Windows Azure (*windowsazure.com*), and Google Compute Engine (GCE)–(*cloud.google.com/compute*).

Platform-as-a-service (PaaS) are development platforms that allow the creation of applications with supported programming languages and tools hosted in the cloud and accessed through a browser. This model can slash development time, offering readily available tools and services. PaaS providers offer a higher-level software infrastructure, where the users can build and deploy particular classes of applications and services using the tools and programming languages supported by the PaaS provider. The users have no control over the underlying infrastructure, such as CPU, network, storage or operating system, as it is abstracted away below the platform [42]. Examples of PaaS services are Google App Engine (*cloud.google.com/appengine*), OpenShift (*openshift.com*), and Heroku (*heroku.com*).

Finally, in the software-as-a-service (SaaS) model, applications run on the cloud infrastructure and are accessible from various client devices. The users of these services do not control the underlying infrastructure and application platform, i.e., only limited user-configurations are available. The main architectural difference between the traditional software model and SaaS model is the number of tenants the applications support. From the user viewpoint, the SaaS model allows him/her to save money in servers and software licenses. Examples of SaaS are SalesForce (*salesforce.com*), NetSuite (*netsuite.com*) and Microsoft Office Web Apps (*office.com*).

Although cloud computing has emerged mainly from the appearance of public computing utilities, other deployments model have been adopted [42]. The cloud computing deployment models are public, private and hybrid. Public clouds are operated by organizations with common interests available in a pay-as-you-go model [2]. Private clouds are operated by internal data center and are not available to the general public. A hybrid cloud is a private cloud supplemented with computing capacity from public clouds. The approach of temporarily renting capacity to handle spikes in load is known as cloud-bursting [31].

2.1. Virtualization

The majority of cloud systems employ virtualization techniques for resource management and workload isolation.

Virtualization abstract the computing resources (e.g., CPU, storage, and network) from the applications to improve sharing of computer systems [26]. The use of virtualization exists since 1960s,

when it was first implemented by IBM to provide concurrent, interactive access to the mainframe 360/67.

A virtual machine (VM) is an environment provided by a virtualization software called virtual machine monitor (VMM), or hypervisor. In this case, the virtualization layer is placed between the bare hardware and the guest operating systems and gives the OSes a virtualized view of the hardware. The platform used by the hypervisor is named host machine, and the module that uses the virtual machine is named guest machine. An important function of the hypervisor is to provide the connection between virtual machines and the host machine. It also abstracts the resources of the host machine, which will be used by the operating system through the virtual machine, and it provides the isolation among virtual machines placed in the same host machine, guaranteeing the independence of each other. Furthermore, the hypervisor handles changes in the processor where the application is running on without affecting the user's OS or application [46, 51].

Consequently, one immediate benefit of virtualization is the option to run multiple operating systems and software stacks on a single physical platform.

The adoption of virtualization relies on some of characteristics such as workload isolation, consolidation and migration [3].

Workload isolation is the confinement of all program instructions into a virtual machine. In this case, better reliability can be achieved because software failures in one virtual machine (VM) do not affect other VMs [56]. Also, better performance control is attained since the execution of one VM should not affect the performance of other VMs. Virtualization makes it possible to consolidate heterogeneous workloads onto a single physical platform, reducing the total cost of ownership and leading to better utilization. This practice is also employed to overcome potential software and hardware incompatibilities in case of upgrades, allowing systems to run legacy and new operating systems concurrently [56]. Workload migration, also known as application mobility [26], can be employed in many cases, such as hardware maintenance, load balancing and disaster recovery. This is done by decoupling the guest operating system from the hardware and encapsulating its state into a VM, allowing it to be suspended, serialized and migrated to a different platform. A VM's state includes a disk partition image, configuration files, and a RAM image [32]. This capability delivers improved quality of service at a relatively low operational cost [56].

2.2. Workload and Server Consolidation

Consolidation is a technique that reallocates virtual machines to achieve some objectives. For instance, it can be used to reduce the amount of physical machines to run the virtual machines or to improve performance of a virtual machine, migrating it for a more powerful physical machine [59, 41].

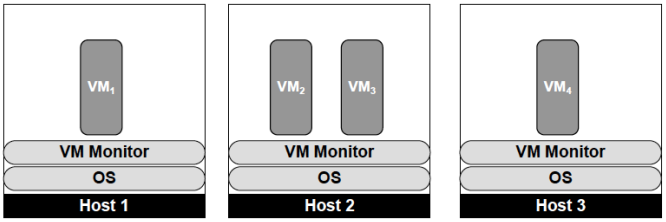
Virtualization and live migration make possible to consolidate heterogeneous workloads onto a single physical platform, reducing the total cost of ownership and leading to better utilization. This practice is also employed to overcome potential software and hardware incompatibilities in case of upgrades, allowing systems to run legacy and new operating systems concurrently [56]. Recently, consolidation has been used to reduce the number of underutilized servers.

Figure 1 illustrates the consolidation strategy. First, a power-inefficient allocation is shown in Figure 1(a). In this case, there are three active quad-core hosts, two of them with 25% of their capacity utilized and one with 50% of its capacity utilized. With consolidation, as shown in Figure 1(b), all virtual machines are allocated in one host and the other hosts can be turned off, reducing the power consumption of the whole system.

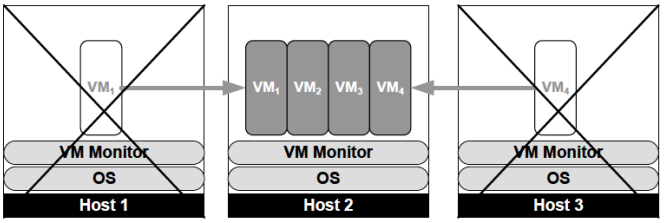
Server consolidation uses workload consolidation in order to reduce the number of active servers. It is the process of gathering several virtual machines into a single physical server. It is often used by data centers to increase resource utilization and to reduce electricity costs [59].

The consolidation process can be performed in a single step using the peak load demands, known as static consolidation, or in a dynamic manner, re-evaluating periodically the workload demand in each virtual machine (i.e., dynamic consolidation).

In static consolidation, once allocated, a virtual machine stays in the same physical server during its whole lifetime. In this case, live migration is not used. The utilization of the peak load demand ensures that the virtual machine does not overload. However, in a dynamic environment with different



(a) Before consolidation



(b) After consolidation

Figure 1. Example of workload consolidation using virtual machines

load patterns, the virtual machine state can be idle most of the time, resulting in an inefficient power allocation.

Dynamic consolidation usually yields better results since the allocation of virtual machines occurs according to the current workload demands. Dynamic consolidation may require migrating virtual machines between physical servers in order to [21]: (i) pull out physical servers from an overload state when the total number of virtual machines mapped to a physical server becomes higher than its capacity; (ii) or turn off a physical server when it is idle or when the virtual machines mapped to it can be moved to another physical server.

Consolidation influences utilization of resources in a non-trivial manner. Clearly, energy usage does not linearly add when workloads are combined. For example, in an Intel i7 machine (4 real cores and 4 cores emulated) an application using 100% of one core, with the other cores in the idle state, consumes 128W whereas the same application using 100% of eight cores consumes 170W [6]. Moreover, resource utilization and performance can also change in a non-trivial manner. Performance degradation occurs with consolidation because of internal conflicts among consolidated applications, such as cache conflicts, conflicts at functional units of the CPU, disk scheduling conflicts, and disk write buffer conflicts [60].

In a cloud computing environment, server consolidation presents some additional difficulties such as: (i) the cloud computing environment must provide reliable QoS, normally defined in terms of service-level agreement (SLA), which describe characteristics such as minimal throughput and maximal response time delivered by the deployed systems; (ii) there can be dynamic changes of the incoming requests for the services; (iii) the resource usage patterns are usually unpredictable; and (iv) the users have distinct preferences.

3. POWER-AWARE COMPUTING

Cloud computing solutions may have a potential impact on greenhouse gas (GHC), which include CO_2 emissions. Currently, with the energy costs increasing, the focus shifts from optimizing large-scale resource management for pure performance to optimize it for energy efficiency while maintaining the services level [9].

However, this represents a challenging problem, since there are many variables that contribute to the power consumption of a resource. First, large-scale computing infrastructures comprise different

layers, and characterizing the power consumed by each of them is usually a difficult task [48]. For instance, the power consumed by a resource may change according to its position in the data center, as well as the data center's temperature [49, 12]. Second, it is often difficult to determine the locations where a workload should be distributed while considering performance requirements. Third, in large scale, data centers' carbon footprint can vary significantly according to their load [24]. Finally, the environment impact of an application may change depending on the users' location, as electricity cost and carbon footprint are location specific [24, 40]. Moreover, it may be difficult to determine the overall power footprint of a workload since some companies still do not publish any information about their power source [17].

Green computing involves a set of methodologies, mechanisms and techniques that helps computing systems (hardware and/or software) to reduce power consumption or carbon footprint.

3.1. Green Performance Indicator

Various green data center metrics have been introduced to measure data center efficiency under the vision of achieving economical, environment and technological sustainability [35].

One example includes the green performance indicator (GPI). GPI defines a set of policies for data collection and analysis related to energy consumption. In [35], Kipp and colleagues classify the GPIs in four clusters: IT Resource Usage, Application Lifecycle, Energy Impact, and Organizational. In this work, we consider only two of them: the IT Resources Usage and the Energy Impact. The idea of GPIs is interesting because it can be adapted criteria to define SLAs, where requirements about energy efficiency versus the expected quality of the services are specified, and thus, need to be satisfied.

On the one hand, the IT resource usage GPIs characterize the IT resource used by the application. In this case, the energy consumption of an application is related to the resources that it uses. Examples of metrics include CPU usage, RAM memory usage, and I/O activity.

On the other hand, the energy impact GPIs metrics specify the impact of IT service centers and applications in the environment, considering power supply, consumed material, emissions, and other energy factors. Examples these metrics include:

1. application performance indicators: these indicators measure the energy consumption per computing unit depending on the application type. The selection of this indicator depends on the type of application. If it is a simulation application, FLOPS/kWh can be the chosen unit. If it is a web server application, we can use the number of transactions/kWh as a metric.

2. data center infrastructure efficiency (DCiE) & power usage effectiveness (PUE): determine the energy efficiency of a data center. It refers to how much energy the IT equipments consume from the total energy consumption.

3. compute power efficiency (CPE): this indicator measures how efficiently the data center power is used for computation. In this metric, the power consumed by the idle servers counts as overhead. It is computed as shown in Equation (1).

$$CPE = DCiE \ / \ Resource\ utilization \tag{1}$$

4. data center energy productivity (DCeP): this GPI indicates the number of bytes processed per kWh. It is computed as the ratio between the size of the output produced by a data center in bytes and the total energy used in the data center in kWh.

4. RELATED WORK

Many studies have tried to improve the power efficiency of a system by minimizing the static power consumption while trying to increase the performance proportionally to the dynamic power consumption [63, 38]. As a result, the hardware energy efficiency has significantly improved.

However, whereas hardware is physically responsible for mostly of the power consumption, hardware operations are guided by software, which is indirectly responsible for the energy consumption [1, 47].

Some of the studies reported in literature try to minimize the power consumption from the data center perspective, considering that the main reason for energy inefficiency is resource underutilization. One of the first approaches to try to solve this problem consists of shutting down idle nodes [45, 33, 55] and waking up them when the workload increases or the average QoS violation ratio exceeds a threshold.

At the hardware level, improvements are made turning off components, putting them to sleep or changing their frequency using dynamic voltage and frequency scaling (DVFS) techniques [25, 38, 53, 28, 55]. DVFS techniques assume that applications dominated by memory accesses or involving heavy I/O activities can be executed at lower CPU frequency with only a marginal impact on their execution time. In that case, the goal of a DVFS scheduler is to identify each execution phase of an application, quantify its workload characteristics, and then switch the CPU frequency to the most appropriate power/performance mode.

For example, in [34], a power-aware DVFS based cluster scheduling algorithm is presented taking into account performance constraints. The proposed algorithm selects the appropriate supply voltages that minimize energy consumption of the resources. Simulation results show that the scheduling algorithm can reduce the power consumption with an increase in the execution time.

In [30], the authors describe a virtual machine placement framework called Entropy. Entropy aims to minimize the number of physical hosts to allocate the virtual machines, without violating any constraints (e.g., memory size and number of CPUs). Its placement process comprises two phases. The first phase identifies the nodes that have sufficient resources (i.e., RAM and CPU) to host a VM, and the second one allocates the virtual machines trying to minimize both the number of physical hosts and the number of VM migrations. These two phases use constraint programming to find out a feasible global solution. Experimental realized in the Grid'5000 testbed show that constraint programming outperforms the first-fit decreasing (FDD) algorithm with regard to the number of VM migrations and power savings. In [19], the authors use Entropy to allocate virtual machines in a federated cloud environment, taking into account power consumption and CO_2 emissions. The experimental results considering two synthetic workloads and two federated clouds show a power saving of almost 22% when considering only power and a saving of almost 19% when the allocations considering both power and CO_2 emissions.

In [58], the author present pMapper, a power and VM placement framework. Its architecture comprises three different managers: performance, migration, and power. This optimization process starts with a sensor collecting the current performance and power characteristics of both virtual and physical machines. Then, it sends these data for the performance and power managers. After, the performance manager analyses the data and based on SLA violations, it suggests to resize the virtual machines. Similarly, the power manager based on the current power consumption suggests power throttling actions (e.g. DVFS or CPU throttling). Based on these suggestions, an arbitrator component selects a configuration, and defines the physical machines to host the virtual machines, as well as the size of each VM. Finally, the managers resize and migrate the virtual machines. Due to heterogeneous platforms, each manager consults a knowledge base to determine the cost of a VM migration in the performance of its applications, as well as in the power consumption. Moreover, pMapper implements three algorithms called: min Power Parity (mPP), min Power Placement with history (mPPH), and PMaP. The mPP algorithm takes as input the VMs' sizes, their current allocations, and a power model of the physical machines. Then, it attempts to reallocate the virtual machines in order to minimize the total power consumption. The mPPH, on the other hand, extends the mPP to minimize VM migrations. Finally, the PMaP tries to find out an allocation that minimizes both power consumption and VM migrations. Experimental results show that the mPP and mPPH algorithms can reduce 25% of the power consumption when the utilization ratio is at most 75% of the cluster's capacity.

In [16], a market-based multi-agent resource allocation model is presented that aims to provide an effective resource allocation policy through genetic algorithm in a cloud environment. Buyer and service provider agents determine the bid and ask prices using interactions to find an acceptable

price considering the demands, the availability of cloud resources and constraints of the cloud user / service provider.

In [20], a Semantically Enhanced Resource Allocation (SERA) framework is used to process distributed resource allocations, using agents in a cloud environment. Agents negotiation involves the combination of customer and provider policies, trying to obtain scheduling results which satisfy both parts.

In [18], the authors address the coordination of multiple autonomic managers for power and performance trade-offs in a real data center environment, with a real HTTP traffic and time-varying demand. By turning off servers under low load condition, the proposed approach achieved power savings of more than 25% without incurring in SLA penalties.

In [39], two energy-conscious task consolidation heuristics (ECTC and MaxUtil) are used to maximize resource utilization for power saving. The cost of the ECTC heuristics is computed considering the energy consumption to run a group of parallel tasks. The MaxUtil heuristic tries to increase the consolidation density. Simulation results show that the proposed heuristics are able to save energy by 18%(ECTC) and 13%(MaxUtil).

In [10], CloudSim is used to simulate VM provisioning techniques. Experimental results compare the performance of two energy-conscious resource management techniques (DVFS and an extension of DVFS policy). In the DVFS policy, VMs were resized according with to host's CPU utilization. In the extension of DVFS, VMs were migrated every 5 seconds using a greedy algorithm that sorts VMs in decreasing order of CPU utilization. In both of them, each VM was migrated to hosts that have resources kept below an utilization threshold. Experimental results show that the total power consumption of a data center reduced up to 50%, but with a increase in the number of SLA violations.

In [67], Zhou and colleagues propose and evaluate a service scheduling approach, called Random Dynamic Scheduling Problem (RDSP), to reduce energy consumption in cloud computing environments. This approach uses Monte Carlo sample historical data to approximate to the predictable user demand and a probabilistic model to express QoS requirements in a homogeneous environment, where the servers' power consumption is constant. Using numeric validation and Monte Carlo sampling to estimate the users' demand, the results show that the proposed scheduling strategy could decrease the power consumption of the server when the user demand is predictable.

In [11], the authors present a virtual machine consolidation policy for cloud computing. The proposed policy aims to minimize power consumption taking into account QoS requirements. And, it extends the Minimum Power policy [5] in order to minimize VM migrations and to maximize resource usage. In this case, different from the previous policy, a VM is migrated only when its node is overloaded and with SLA violations. Experimental results show a reduction in the power consumption (up to 34%) and in the execution time (63%). Moreover, the new policy increases SLA guarantees. The experiments were realized through the CloudSim simulator considering one data center with 800 physical nodes.

Table I summarizes the eleven approaches discussed in the previous paragraphs. As can be seen in this Table, three approaches [16, 20, 18] use multi-agent systems to reduce power consumption and costs. One of them is targeted to a cloud environment and two execute in cluster computing environments. Five works [39, 10, 19, 67, 11] reduce power consumption in cloud computing considering SLAs. Only one proposal [19] deals with cloud federation to implement workload consolidation, without implementing negotiation mechanisms between in the data centers. None of these eleven proposals tackle federated cloud environments for power-aware allocations that are SLA-conscious and the workload migration requires negotiation between the data centers.

5. DESIGN OF A MULTI-AGENT SERVER CONSOLIDATION MECHANISM

The main goal of our approach, called federated application provisioning (FAP), is to reduce power consumption of data centers, trying to meet QoS requirements, with limited energy defined by a third party agent (carbon emission regulator agency). We consider that data centers are concerned by an energy threshold, and they are in a federated cloud computing environment, scheduling online the

Paper	Target	Federated	Multi-agent	Migration	Negotiation
[34]	Cluster	No	No	No	No
[30]	Cluster	No	No	Same DC	No
[19]	Cloud	Yes	No	Among DCs	No
[58]	Cluster	No	No	Same DC	No
[16]	Cloud	No	Yes	No	No
[20]	Cluster	No	Yes	No	No
[18]	Cluster	No	Yes	No	No
[39]	Cloud	No	No	No	No
[10]	Cloud	No	No	Same DC	No
[67]	Cloud	No	No	No	No
[11]	Cloud	No	No	Same DC	No
This work	Cloud	Yes	Yes	Among DCs	Yes

Table I. Comparative summary of cloud server consolidation strategies

execution of the users' applications. In this case, a multi-agent strategy is used to negotiate resources' allocations and the final price to execute the users' tasks.

We assume a federated cloud model composed of public and private clouds, and that the cost to transfer an application or to migrate VMs across the clouds is known by the cloud providers.

In our cloud environment there are four distinct agents: cloud service provider (CLSP), cloud user (CLU), electric power provider (EPP), and carbon emission regulator agency (CERA) as shown in Figure 2. In our design, the carbon emission regulator agency determines the amount of carbon emissions that both CLSP and EPP can emit in a period of time.

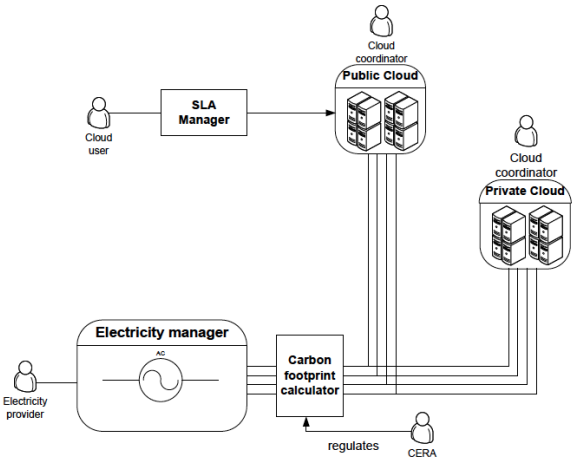


Figure 2. Agents of the cloud market

We also assume that each cloud is composed of one data center with one coordinator accountable for monitoring the metrics and for negotiating with the other agents (CLSP, CLU, EPP, and CERA). There are also sensors to monitor power consumption, resource usage, and SLA violation as depicted in Figure 3.

Finally, we consider that the cloud system has a communication layer such that any participant can exchange messages. Messages and QoS metrics are described in a format that is known by the agents, and a cloud provider cannot reject users' tasks.

The proposed scenario includes a set of data centers (clouds) composed by a set of virtual machines, which are mapped to a set of interconnected physical servers deployed across the clouds. Let $R = \{r_1, r_2, \cdots, r_n\}$ be the set of resources in data center i with a capacity c_i^k, where $k \in R$. The power consumption (P_i) can be defined as [39]:

$$P_i = (p_{max} - p_{min}) * U_i + p_{min} \tag{2}$$

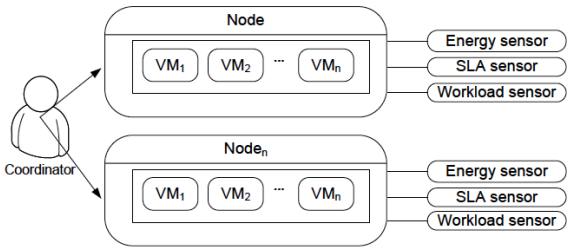


Figure 3. Detailed view of a data center

where p_{max} is the power consumption for the data center i at the peak load, p_{min} is the minimum power consumption in active mode, and U_i is the resource utilization of data center i as defined in Equation (3) [39]:

$$U_i = \sum_{j=1}^{n} u_{i,j} \tag{3}$$

where $u_{i,j}$ is the resource usage of resource j in the data center i.

The relation between a cloud provider and a cloud user is determined by a set of QoS requirements described in the SLA. Furthermore, data centers are subjected to an energy consumption threshold agreed among the CLSP, the EPP, and the CERA. When the energy consumption threshold is violated, this implies additional costs. To calculate the carbon footprint of the CLSP and the EPP, the CERA uses the following metrics: application performance indicators (FLOPS/kWh), data center DCiE, PUE, and CPE.

Let T represent a set of independent tasks to be executed, which is subject to a set of QoS constraints such as minimum RAM memory, minimum CPU utilization, and minimum execution time. In this case, the following steps are executed:

1. when a task t_i is submitted, the cloud provider calculates the price of t_i's execution (σ_i) based on the power consumption (Equation (2)).

2. the cloud provider tries to place t_i in an available resource, using consolidation techniques to reduce the number of active physical servers.

3. if the cloud provider does not have enough available resources or the energy threshold will be violated, the cloud provider first contacts another cloud provider and negotiates with it the execution of this task. In this case, the price of this execution (C_t) is defined as shown in Equation (4).

$$C_t = \sigma_t + \epsilon_t + \lambda_t \tag{4}$$

   where σ_t is the financial cost of executing task t based on its power consumption, ϵ_t is the cost of the power impact of a task t in the environment, and λ_t is the cost to transfer a task t to another cloud provider.

4. if the cloud provider does not succeed, it tries to consolidate its VMs considering the service-level agreements.

5. If not possible, it tries to negotiate the energy threshold with the CERA and with the EPP agents.

6. If all negotiations fail, the cloud provider finds the SLA whose violation implies in lower cost, terminates the associated task, and executes the task t_i. In this case, the price to execute the tasks is defined as shown in Equation (5).

$$V_t = C_t + \gamma + \delta \tag{5}$$

where γ is the cost to violate the QoS requirements of other tasks and δ is the cost associated with the power consumption violation.

To control tasks' allocation, each cloud provider has a 3-dimensional matrix representing the tasks ($t_i \in T$), the virtual machines (vm_j), and physical servers ($r_z \in R$), where $r(i, j, z) = 1$ iff task t_i is allocated at virtual machine vm_j in resource r_z; 0 indicates that the task can be allocated in vm_j; and finally, -1 represents that the allocation is impossible.

In order to illustrate our strategy, consider a federated cloud environment with 2 clouds (DC1 and DC2) and one user that contracted one cloud to execute him/her applications. Consider that the contracted cloud (DC1) is overloaded and that the QoS requirements described in the SLAs are based on response time. In this scenario, when the user submits a set of tasks to execute, the cloud provider of DC1 first tries to execute it locally considering power consumption and its available resources. Since DC1 is overloaded, its cloud provider contacts another data center (DC2) and negotiates the execution of the tasks. If DC2 accepts, the cost of the tasks execution is calculated using Equation (4). If DC2 refuses, then DC1 tries to consolidate its virtual machines and, if not possible, it tries to negotiate the energy threshold with the carbon emission regulator agency (CERA) and with the electric power provider (EPP) considering the following metrics: application performance indicators (FLOPS/kWh), DCiE, PUE, and CPE. If all negotiations fail, then DC1 finds the SLA whose violations implies in lower cost and terminates the execution of its associated task. Then, the cost to execute the tasks is calculated using Equation (5).

6. EXPERIMENTAL RESULTS

In this section, we evaluate the proposed server consolidation mechanism for federated clouds. We use the cloud simulator CloudSim [10], which is a well-established cloud simulator that has been used in many previous works [50, 65, 62], among others, for simulating resource management strategies. CloudSim enables modeling and simulation of cloud computing systems. It provides support fot cloud system components such as data centers, virtual machines, and resource provisioning policies.

6.1. Modifications in CloudSim

In order to enable federation and energy regulation capabilities, we added 4 classes to CloudSim, which are described below. CloudSim already implemented the support to measure the power consumption of the nodes.

The **CloudEnergyRegulation** class represents the behavior of the carbon emission regulator agency (CERA) agent. The CERA communicates with the data center cloud coordinator to inform the power consumption threshold.

The **DatacenterEnergySensor** class implements the **Sensor** interface that monitors the power consumption of the data center and informs the coordinator. When the power consumption is close to the limit, this sensor creates an event (i.e., CloudSim event) and notifies the coordinator. In this case, the coordinator first tries to contact another data center to transfer the virtual machines and if the data center does not accept, then the coordinator tries to consolidate them (Section 5).

The **FederatedPowerVmAllocationPolicy** class extends the **VmAllocationPolicy** class to implement our strategy to allocate the virtual machines across the data centers.

Finally, the **CustomerDatacenterBroker** class models the QoS requirements customer behavior, negotiates with the cloud coordinator, and requests the resources.

6.2. Simulation Environment

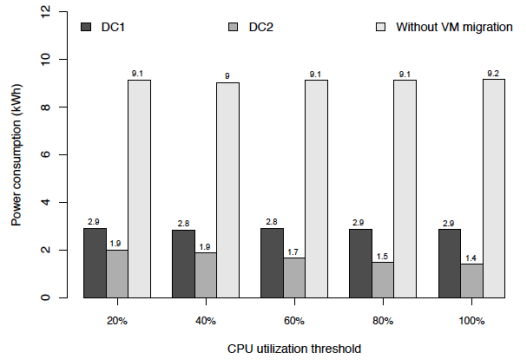
In order to evaluate the effectiveness of our federated application provisioning (FAP) technique, we used a simulation setup that is similar to the one described in [10]. Our simulation environment

included two clouds, each one with one data center (DC1 and DC2) that had 100 hosts each. These hosts were modeled to have one CPU with four cores with 1000 MIPS, 2GB of RAM and 1TB of storage. The workload model comprises the provisioning and allocating for 400 virtual machines. In this case, each virtual machine requested one CPU core, 256MB of RAM and 1GB of storage. The CPU utilization distribution was generated according to the Poisson distribution, where each virtual machine required 150 MIPS and 1 to 10 minutes to complete execution, assuming a CPU utilization of 20, 40, 60, 80 and 100% and a global energy consumption threshold of 3 kWh of energy per data center. Initially, the provisioner allocates as many as possible virtual machines in a single host, without violating any constraint of the host. The SLA was defined in terms of response time (10 minutes).

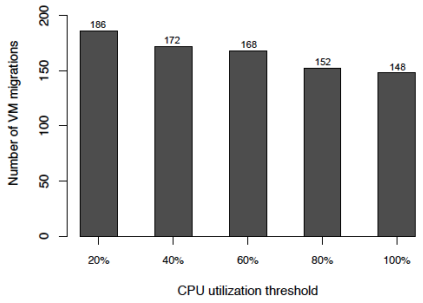
The energy consumption threshold of 3 kWh of energy per data center was chosen based in the results of the power management technique, presented in [10].

6.3. Scenario 1: workload submission to a single data center under power consumption threshold

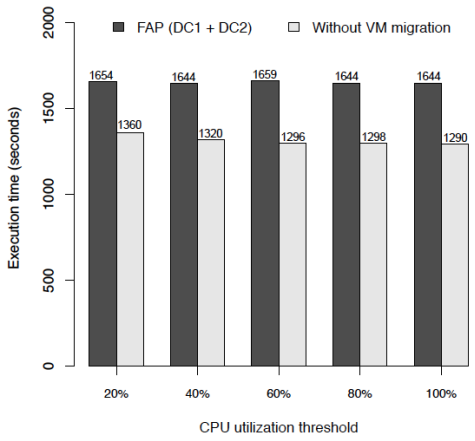
In this scenario, tasks are always submitted to data center 1 (DC1). If needed, VMs are migrated from DC1 to DC2. The simulation was repeated 10 times and the mean values for energy consumption without our mechanism using only DC1 (trivial), and with our federated application provisioning (FAP) approach are presented in Figure 4(a).



(a) Power consumption of the data centers with and without the FAP approach



(b) Number of VM migrated from DC1 to DC2 using the FAP approach



(c) Execution time of the tasks with and without the FAP approach

Figure 4. Power consumption with 2 data centers under limited power consumption

Figure 4(a) shows that the proposed provision technique can reduce the total power consumption of the data centers, without SLA violation. In this case, an average reduction of 46% in the power consumption was achieved since the data center 1 (DC1) consumed more than 9 kWh with the trivial approach (without VM migration) and no more than 4.8 kWh was consumed in total by both data centers with our approach (2.9 kWh for DC1 and 1.9 kWh for DC2). In order to achieve this, data center 1 (DC1) tried first to maximize the usage of its resources and to consume the limit of its energy power threshold, without violating the SLAs. Hence, the data center 2 (DC2) was only used in imminence of SLA violation or when the energy consumption was close to violate the limit. In all cases, the energy consumption for DC1 remained close to the limit.

Figure 4(b) presents the number of VMs migrated when our mechanism is used. It can be seen that the number of migrations decreases as the threshold of CPU usage increases. This result was expected since with more CPU capacity, the allocation policy tends to use it and to allocate more virtual machines in the same host.

In Figure 4(c), we measured the wallclock time needed to execute 400 tasks, with and without our mechanism (FAP). It can be seen that the proposed provision technique increases the whole execution time. This occurs because of the overhead caused by VMs migrations between the data centers, and the negotiations between the CLU and the CLSP. Nevertheless, this increase is less than 22%, where the wallclock execution times without and with the FAP mechanism are 21.5 minutes and 27.4 minutes, respectively, when using the whole CPU capacity. We consider that this increase in the execution time is compensated by the reduction in the power consumption (Figure 4(a)).

6.4. Scenario 2: distinct workload submission to different overloaded data centers

In this scenario, we consider two users, with distinct SLAs and each user submits 400 tasks to different data centers (DC1 and DC2). Our goal is to observe the rate of SLA violation when the workload of both data centers is high. The energy consumption of the data centers is presented in Figure 5(a).

In Figure 5(a), we can see that, even in a scenario with overloaded data centers, our mechanism can maintain the power consumption below the threshold (3 kWh) for each data center. Using the whole CPU capacity, the power consumption decreased from 9.2 kWh to 5.5 kWh (DC1 + DC2), reaching a reduction of 40% in the power consumption.

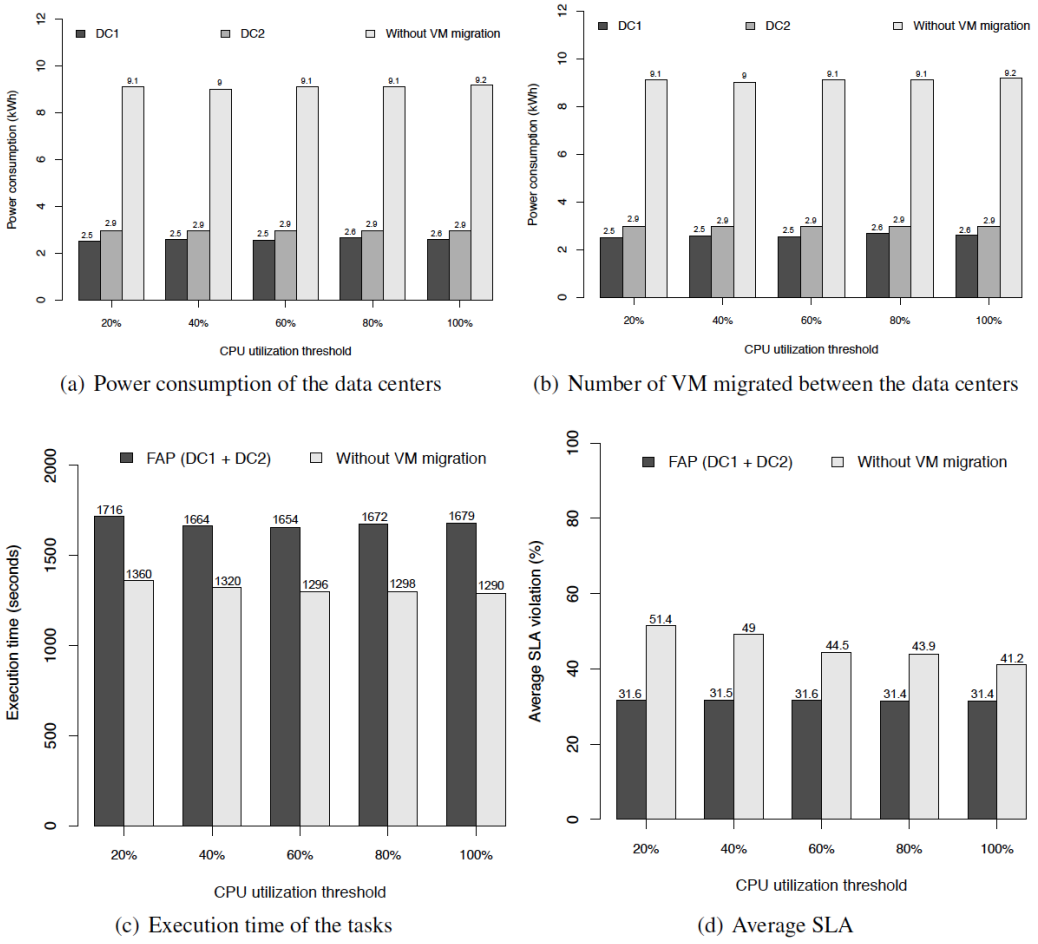
Figures 5(b) and 5(c) show the number of VM migrations between the data centers and the wallclock time to execute 800 tasks when both data centers are overloaded. Comparing with the scenario with one overloaded data center (DC1), the number of VM migrations decreased, keeping almost the same penalty in the execution time (23%) due to the negotiations overhead between the agents and by server consolidations.

The number of SLA violations with two overloaded data centers was lower than with just one data center (DC1) as we can see in Figure 5(d). With the CPU utilization threshold of 80%, the SLA violation decreased from 43.9% (DC1) to 31.4% (DC1 + DC2), reaching 28% of reduction in the SLAs violations. This shows the appropriateness of VM migration between different data centers in an overloaded scenario.

In order to analyze the impact of energy threshold in the SLA violation, we increase it for 4 kWh per data center. As we can see in Figure 6(a) the energy consumption of both data centers was close to limit but with a reduction in the SLA violation. With the CPU utilization threshold of 80%, the SLA violation decreases from 44% (DC1) to 18% (DC1 + DC2) reaching 59% of reduction in SLAs violations. Increasing the energy threshold from 3 kWh to 4 kWh decreases the SLA violation from 31.4% (DC1 + DC2) to 18% (DC1 + DC2).

7. FINAL CONSIDERATION AND FUTURE WORK

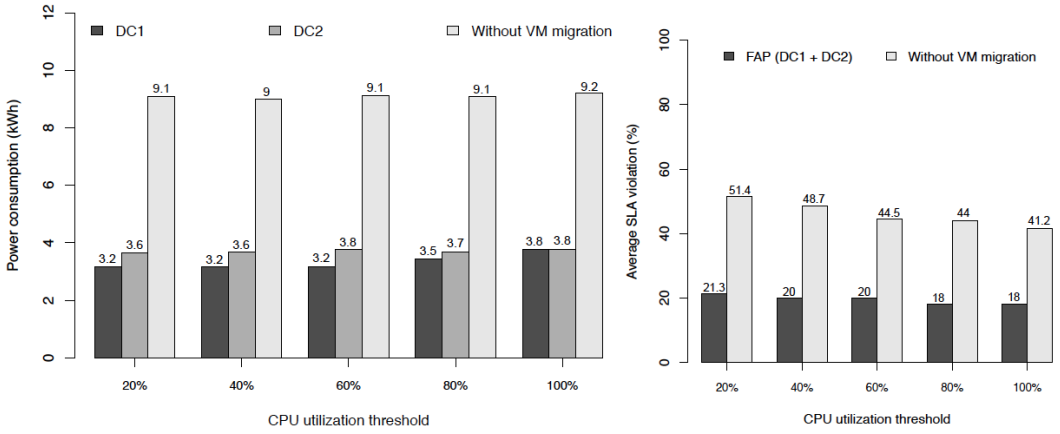
In this paper, we proposed and evaluated a server consolidation strategy to reduce power consumption on cloud federations. Our server consolidation strategy aims to reduce power consumption on cloud federations while trying to meet QoS requirements. We assumed that clouds'



(a) Power consumption of the data centers (b) Number of VM migrated between the data centers

(c) Execution time of the tasks (d) Average SLA

Figure 5. Power consumption of two overloaded data centers under limited power consumption of 3 kWh



(a) Power consumption of the data centers (b) Average SLA violation

Figure 6. Power consumption of two overloaded data centers under limited power consumption of 4 kWh

data centers have a limited power consumption defined by a third party agent. In this case, we

addressed applications' workloads, considering the costs to turn servers on/off and to migrate the virtual machines in the same data center and between different data centers. Simulation results showed that our strategy could reduce up to 46% of the power consumption, with a slowdown of 22% in the execution time. Similar to other works [50, 65, 62], the experiments were realized through the CloudSim [10] simulator with two clouds and 400 simultaneous virtual machines. Altogether, the results demonstrated that cloud federation can provide an interesting solution to deal with power consumption, by using the computing infrastructure of other clouds when a cloud runs out of resources or when other clouds have power-efficient resources. Even though we achieved very good results with our strategy, we noticed that other variables should also be considered such as the workload type, the data center characteristics (i.e., location, power source), and the network bandwidth as these variables may impact the whole power consumption of a data center. In addition, since the CPU no longer dominates the nodes' power consumption [44], the power consumption of other components (e.g., memory, disks) must be taken into account. Moreover, resource heterogeneity should also be considered, as data centers usually comprise heterogeneous resources that can have different power consumption and curves. This requires energy and performance-aware load distribution strategies. We leave these extensions for future work.

REFERENCES

[1] G. Agosta, M. Bessi, E. Capra, and C. Francalanci. Dynamic memoization for energy efficiency in financial applications. In *International Green Computing Conference and Workshops*, pages 1–8, 2011.

[2] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andrew Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. Above the clouds: A berkeley view of cloud computing. Technical Report UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Feb 2009.

[3] Paul Barham, Boris Dragovic, Keir Fraser, Steven Hand, Tim Harris, Alex Ho, Rolf Neugebauer, Ian Pratt, and Andrew Warfield. Xen and the art of virtualization. In *SOSP*, pages 164–177, 2003.

[4] Luiz André Barroso, Jimmy Clidaras, and Urs Hölzle. *The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines*. Synthesis Lectures on Computer Architecture. Morgan and Claypool Publishers, 2nd edition, 2013.

[5] 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*, 24(13), 2012.

[6] Anton Beloglazov, Rajkumar Buyya, Young Choon Lee, and Albert Zomaya. A taxonomy and survey of energy-efficient data centers and cloud computing systems. In *Advances in Computers*, volume 82, pages 47–111. Elsevier, 2011.

[7] Andreas Berl, Erol Gelenbe, Marco Di Girolamo, Giovanni Giuliani, Hermann De Meer, Minh Quan Dang, and Kostas Pentikousis. Energy-efficient cloud computing. *Comput. J.*, 53:1045–1051, 2010.

[8] Rajkumar Buyya, Rajiv Ranjan, and Rodrigo N. Calheiros. Intercloud: utility-oriented federation of cloud computing environments for scaling of application services. In *10th International Conference on Algorithms and Architectures for Parallel Processing*, pages 13–31, 2010.

[9] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25:599–616, 2009.

[10] Rodrigo N. Calheiros, Rajiv Ranjan, Anton Beloglazov, César A. F. 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*, 41(1):23–50, 2011.

[11] Zhibo Cao and Shoubin Dong. An energy-aware heuristic framework for virtual machine consolidation in cloud computing. *The Journal of Supercomputing*, 69(1):429–451, 2014.

[12] Eugenio Capra, Chiara Francalanci, and Sandra A. Slaughter. Is software "green"? application development environments and energy efficiency in open source applications. *Information and Software Technology*, 54(1):60–71, 2012.

[13] A. Celesti, F. Tusa, M. Villari, and A. Puliafito. How to enhance cloud architectures to enable cross-federation. In *IEEE 3rd International Conference on Cloud Computing*, pages 337–345, 2010.

[14] Vinton G. Cerf. ACM and the professional programmer. *Queue*, 12(7):10:10–10:11, 2014.

[15] Anantha P. Chandrakasan and Robert W. Brodersen. Minimizing power consumption in digital cmos circuits. *Proceedings of the IEEE*, 83(4):498–523, 1995.

[16] Yee Ming Chen and Hsin-Mei Yeh. An implementation of the multiagent system for market-based cloud resource allocation. *Journal of Computing*, 2(11):27–33, 2010.

[17] Gary Cook, Tom Dowdall, David Pomerantz, and Yifei Wang. Clicking Clean: How companies are creating the green Internet 2014. Technical report, Greenpeace, April 2014. Last accessed in December 2015.

[18] Rajarshi Das, Jeffrey O. Kephart, Charles Lefurgy, Gerald Tesauro, David W. Levine, and Hoi Chan. Autonomic multi-agent management of power and performance in data centers. In *7th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 107–114, 2008.

[19] Corentin Dupont, Thomas Schulze, Giovanni Giuliani, Andrey Somov, and Fabien Hermenier. An energy aware framework for virtual machine placement in cloud federated data centres. In *3rd International Conference on Future Energy Systems: Where Energy, Computing and Communication Meet*, pages 4:1–4:10, 2012.

[20] J. Ejarque, R. Sirvent, and R.M. Badia. A multi-agent approach for semantic resource allocation. In *Second IEEE International Conference on Cloud Computing Technology and Science*, pages 335–342, 2010.

[21] Tiago C. Ferreto, Marco A. S. Netto, Rodrigo N. Calheiros, and César A. F. De Rose. Server consolidation with migration control for virtualized data centers. *Future Generation Computer Systems*, 27(8):1027–1034, 2011.

[22] Global Inter-Cloud Technology Forum. Use cases and functional requirements for inter-cloud computing. Technical report, Global Inter-Cloud Technology Forum, 2010.

[23] I. Foster, Yong Zhao, I. Raicu, and S. Lu. Cloud computing and grid computing 360-degree compared. In *Grid Computing Environments Workshop*, pages 1–10, 2008.

[24] Peter Xiang Gao, Andrew R. Curtis, Bernard Wong, and Srinivasan Keshav. It's not easy being green. In *Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 211–222, 2012.

[25] Rong Ge, Xizhou Feng, Wu-chun Feng, and Kirk W. Cameron. CPU MISER: A Performance-Directed, Run-Time System for Power-Aware Clusters. In *International Conference on Parallel Processing*, pages 18–26, 2007.

[26] R. P. Goldberg. Survey of virtual machine research. In *Computerlinguistik*, pages 34–45, 1974.

[27] Greenpeace. Make it green: Cloud computing and its contribution to climate change. Technical report, Greenpeace International, Mar 2010.

[28] Vinay Hanumaiah and Sarma Vrudhula. Energy-efficient operation of multi-core processors by dvfs, task migration and active cooling. *IEEE Transactions on Computers*, 99, 2012.

[29] Michael Hauben and Ronda Hauben. *Netizens: On the History and Impact of Usenet and the Internet*. Wiley-IEEE Computer Society, 1997.

[30] Fabien Hermenier, Xavier Lorca, Jean-Marc Menaud, Gilles Muller, and Julia Lawall. Entropy: A consolidation manager for clusters. In *ACM SIGPLAN/SIGOPS International Conference on Virtual Execution Environments*, pages 41–50, 2009.

[31] Paul T. Jaeger, Jimmy Lin, Justin M. Grimes, and Shannon N. Simmons. Where is the cloud? geography, economics, environment, and jurisdiction in cloud computing. *First Monday*, 14(5–4), 2009.

[32] K. Keahey, I. Foster, T. Freeman, and X. Zhang. Virtual workspaces: Achieving quality of service and quality of life in the grid. *Sci. Program.*, 13:265–275, Oct 2005.

[33] Hamzeh Khazaei, Jelena Misic, and Vojislav B. Misic. Performance Analysis of Cloud Computing Centers Using M/G/m/m+r Queuing Systems. *IEEE Transactions on Parallel and Distributed Systems*, 23(5):936–943, 2012.

[34] Kyong Hoon Kim, Wan Yeon Lee, Jong Kim, and Rajkumar Buyya. Sla-based scheduling of bag-of-tasks applications on power-aware cluster systems. *IEICE Transactions on Information and Systems*, E93-D(12):3194–3201, 2010.

[35] Alexander Kipp, Tao Jiang, Mariagrazia Fugini, and Ioan Salomie. Layered green performance indicators. *Future Generation Computer Systems*, 28(2):478–489, February 2011.

[36] Jonathan G Koomey. Estimating total power consumption by servers in the u.s. ad the world. *Lawrence Berkley National Laboratory*, 2007.

[37] Jonathan G Koomey. Growth in data center electricity use 2005 to 2010. *Oakland, CA: Analytics Press*, 2011.

[38] Wan Yeon Lee. Energy-saving dvfs scheduling of multiple periodic real-time tasks on multi-core processors. In *13th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications*, pages 216–223, 2009.

[39] Young Choon Lee and Albert Y. Zomaya. Energy efficient utilization of resources in cloud computing systems. *The Journal of Supercomputing*, pages 1–13, 2010.

[40] Zhenhua Liu, Yuan Chen, Cullen Bash, Adam Wierman, Daniel Gmach, Zhikui Wang, Manish Marwah, and Chris Hyser. Renewable and cooling aware workload management for sustainable data centers. In *12th ACM SIGMETRICS/PERFORMANCE Joint International Conference on Measurement and Modeling of Computer Systems*, pages 175–186, 2012.

[41] Moreno Marzolla, Ozalp Babaoglu, and Fabio Panzieri. Server consolidation in clouds through gossiping. In *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks*, pages 1–6. IEEE Computer, 2011.

[42] Peter Mell and Timothy Grance. The NIST definition of cloud computing, national institute of standards and technology. Technical Report SP800-145, NIST Information Technology Laboratory, 2011.

[43] Alexandre Mello Ferreira, Kyriakos Kritikos, and Barbara Pernici. Energy-aware design of service-based applications. In *Service-Oriented Computing*, volume 5900, pages 99–114. 2009.

[44] Laurl Minas and Brad Ellison. *Energy Efficiency for Information Technology: How to Reduce Power Consumption in Servers and Data Centers*. Intel Press, 2009.

[45] Isi Mitrani. Service center trade-offs between customer impatience and power consumption. *Performance Evaluation*, 68(11):1222–1231, 2011.

[46] Susanta Nanda and Tzi cker Chiueh. A survey of virtualization technologies. Technical Report TR–179, Stony Brook University, February 2005. Last accessed in December 2015.

[47] Mais Nijim, Xiao Qin, Meikang Qiu, and Kenli Li. An adaptive energy-conserving strategy for parallel disk systems. *Future Generation Computer Systems*, 29(1):196–207, 2013.

[48] Anne-Cecile Orgerie, Marcos Dias de Assuncao, and Laurent Lefevre. A survey on techniques for improving the energy efficiency of large-scale distributed systems. *ACM Computing Surveys*, 46(4):47:1–47:31, 2014.

[49] Anne-Cecile Orgerie, Laurent Lefevre, and Jean-Patrick Gelas. Demystifying energy consumption in grids and clouds. In *International Conference on Green Computing*, pages 335–342, 2010.

[50] Yuxiang Shi, Xiaohong Jiang, and Kejiang Ye. An energy-efficient scheme for cloud resource provisioning based on cloudsim. In *IEEE International Conference on Cluster Computing*, pages 595–599, 2011.

[51] Dinkar Sitaram and Geetha Manjunath. *Moving To The Cloud: Developing Apps in the New World of Cloud Computing*. Syngress Publishing, 2011.

[52] Edward Stanford. Environmental trends and opportunities for computer system power delivery. In *20th International Symposium on Power Semiconductor Devices and IC's*, pages 1–3, 2008.

[53] Vaibhav Sundriyal, Masha Sosonkina, Fang Liu, and Michael W. Schmidt. Dynamic frequency scaling and energy saving in quantum chemistry applications. In *IEEE International Symposium on Parallel and Distributed Processing Workshops and PhD Forum*, pages 837–845, 2011.

[54] G. Terzopoulos and H.D. Karatza. Dynamic voltage scaling scheduling on power-aware clusters under power constraints. In *17th International Symposium on Distributed Simulation and Real Time Applications*, pages 72–78, 2013.

[55] George Terzopoulos and Helen D. Karatza. Maximizing performance and energy efficiency of a real-time heterogeneous 2-level grid system using dvs. In *16th IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications*, pages 185–191, 2012.

[56] R. Uhlig, G. Neiger, D. Rodgers, A.L. Santoni, F.C.M. Martins, A.V. Anderson, S.M. Bennett, A. Kagi, F.H. Leung, and L. Smith. Intel virtualization technology. *Computer*, 38(5):48–56, May 2005.

[57] Luis M. Vaquero, Luis Rodero-Merino, Juan Caceres, and Maik Lindner. A break in the clouds: Towards a cloud definition. *ACM SIGCOMM Computer Communication Review*, 39(1):50–55, 2008.

[58] Akshat Verma, Puneet Ahuja, and Anindya Neogi. pmapper: Power and migration cost aware application placement in virtualized systems. In *ACM/IFIP/USENIX International Conference on Middleware*, pages 243–264, 2008.

[59] Werner Vogels. Beyond server consolidation. *Jounal of ACM Queue*, 6(1):20–26, 2008.

[60] William Voorsluys, James Broberg, Srikumar Venugopal, and Rajkumar Buyya. Cost of virtual machine live migration in clouds: A performance evaluation. In *CloudCom*, pages 254–265, 2009.

[61] Michael Wooldridge and Nicholas R. Jennings. Intelligent agents: theory and practice. *The Knowledge Engineering Review*, 10:115–152, 6 1995.

[62] Linlin Wu, Saurabh Kumar Garg, and Rajkumar Buyya. Sla-based resource allocation for software as a service provider (saas) in cloud computing environments. In *11th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pages 195–204, 2011.

[63] D. Zachhuber, J. Doppler, A. Ferscha, C. Klein, and J. Mitic. Simulating the potential savings of implicit energy management on a city scale. In *12th IEEE/ACM International Symposium on Distributed Simulation and Real-Time Applications*, pages 207–216, 2008.

[64] Qi Zhang, Lu Cheng, and Raouf Boutaba. Cloud computing: state-of-the-art and research challenges. *Journal of Internet Services and Applications*, 1:7–18, 2010.

[65] Qi Zhang, Eren Gürses, Raouf Boutaba, and Jin Xiao. Dynamic resource allocation for spot markets in clouds. In *11th USENIX Conference on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services*, pages 1–6, 2011.

[66] Shuai Zhang, Xuebin Chen, Shufen Zhang, and Xiuzhen Huo. The comparison between cloud computing and grid computing. In *International Conference on Computer Application and System Modeling*, volume 11, pages V11–72–V11–75, 2010.

[67] Liang Zhou, Baoyu Zheng, Jingwu Cui, and Sulan Tang. Toward green service in cloud: From the perspective of scheduling. In *International Conference on Computing, Networking and Communications*, pages 939–943, 2012.