



# Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing

Anton Beloglazov<sup>a,\*</sup>, Jemal Abawajy<sup>b</sup>, Rajkumar Buyya<sup>a</sup>

<sup>a</sup> Cloud Computing and Distributed Systems (CLOUDS) Laboratory, Department of Computer Science and Software Engineering, The University of Melbourne, Australia

<sup>b</sup> School of Information Technology, Deakin University, Melbourne, Australia

## ARTICLE INFO

### Article history:

Received 15 July 2010

Received in revised form

21 March 2011

Accepted 28 April 2011

Available online 4 May 2011

### Keywords:

Energy efficiency

Green IT

Cloud computing

Resource management

Virtualization

Dynamic consolidation

## ABSTRACT

Cloud computing offers utility-oriented IT services to users worldwide. Based on a pay-as-you-go model, it enables hosting of pervasive applications from consumer, scientific, and business domains. However, data centers hosting Cloud applications consume huge amounts of electrical energy, contributing to high operational costs and carbon footprints to the environment. Therefore, we need Green Cloud computing solutions that can not only minimize operational costs but also reduce the environmental impact. In this paper, we define an architectural framework and principles for energy-efficient Cloud computing. Based on this architecture, we present our vision, open research challenges, and resource provisioning and allocation algorithms for energy-efficient management of Cloud computing environments. The proposed energy-aware allocation heuristics provision data center resources to client applications in a way that improves energy efficiency of the data center, while delivering the negotiated Quality of Service (QoS). In particular, in this paper we conduct a survey of research in energy-efficient computing and propose: (a) architectural principles for energy-efficient management of Clouds; (b) energy-efficient resource allocation policies and scheduling algorithms considering QoS expectations and power usage characteristics of the devices; and (c) a number of open research challenges, addressing which can bring substantial benefits to both resource providers and consumers. We have validated our approach by conducting a performance evaluation study using the CloudSim toolkit. The results demonstrate that Cloud computing model has immense potential as it offers significant cost savings and demonstrates high potential for the improvement of energy efficiency under dynamic workload scenarios.

© 2011 Elsevier B.V. All rights reserved.

## 1. Introduction

Cloud computing can be classified as a new paradigm for the dynamic provisioning of computing services supported by state-of-the-art data centers that usually employ Virtual Machine (VM) technologies for consolidation and environment isolation purposes [1]. Cloud computing delivers an infrastructure, platform, and software (applications) as services that are made available to consumers in a pay-as-you-go model. In industry these services are referred to as Infrastructure as a Service (IaaS), Platform as a Service (PaaS), and Software as a Service (SaaS) respectively. Many computing service providers including Google, Microsoft, Yahoo, and IBM are rapidly deploying data centers in various locations around the world to deliver Cloud computing services.

A recent Berkeley report [2] stated: “Cloud computing, the long-held dream of computing as a utility, has the potential to

transform a large part of the IT industry, making software even more attractive as a service”. Cloud offers significant benefit to IT companies by relieving them from the necessity in setting up basic hardware and software infrastructures, and thus enabling more focus on innovation and creating business value for their services. Moreover, developers with innovative ideas for new Internet services no longer require large capital outlays in hardware to deploy their service or human expenses to operate it [2].

To fully realize the potential of Cloud computing, Cloud service providers have to ensure that they can be flexible in their service delivery to meet various consumer requirements, while keeping the consumers isolated from the underlying infrastructure. Until recently, high performance has been the sole concern in data center deployments, and this demand has been fulfilled without paying much attention to energy consumption. However, an average data center consumes as much energy as 25,000 households [3]. As energy costs are increasing while availability dwindles, there is a need to shift the focus from optimizing data center resource management for pure performance to optimizing them for energy efficiency, while maintaining high service level performance.

Therefore, Cloud service providers need to adopt measures to ensure that their profit margin is not dramatically reduced due

\* Corresponding author.

E-mail addresses: [abe@csse.unimelb.edu.au](mailto:abe@csse.unimelb.edu.au) (A. Beloglazov), [jemal.abawajy@deakin.edu.au](mailto:jemal.abawajy@deakin.edu.au) (J. Abawajy), [raj@csse.unimelb.edu.au](mailto:raj@csse.unimelb.edu.au) (R. Buyya).

to high energy costs. The rising energy cost is a highly potential threat as it increases the Total Cost of Ownership (TCO) and reduces the Return on Investment (ROI) of Cloud infrastructures. There is also increasing pressure from governments worldwide aimed at the reduction of carbon footprints, which have a significant impact on the climate change. For example, the Japanese government has established The Japan Data Center Council to address the soaring energy consumption of data centers [4]. Recently, leading computing service providers have formed a global consortium known as The Green Grid [5] to promote energy efficiency for data centers and minimization of the environmental impact. Thus, providers need to minimize energy consumption of Cloud infrastructures, while ensuring the service delivery.

Lowering the energy usage of data centers is a challenging and complex issue because computing applications and data are growing so quickly that increasingly larger servers and disks are needed to process them fast enough within the required time period. *Green Cloud computing* is envisioned to achieve not only the efficient processing and utilization of a computing infrastructure, but also to minimize energy consumption [6]. This is essential for ensuring that the future growth of Cloud computing is sustainable. Otherwise, Cloud computing with increasingly pervasive front-end client devices interacting with back-end data centers will cause an enormous escalation of the energy usage. To address this problem and drive Green Cloud computing, data center resources need to be managed in an energy-efficient manner. In particular, Cloud resources need to be allocated not only to satisfy Quality of Service (QoS) requirements specified by users via Service Level Agreements (SLAs), but also to reduce energy usage.

The main objective of this work is to present our vision, discuss open research challenges in energy-aware resource management, and develop efficient policies and algorithms for virtualized data centers so that Cloud computing can be a more sustainable and eco-friendly mainstream technology to drive commercial, scientific, and technological advancements for future generations. Specifically, our work aims to:

- Define an architectural framework and principles for energy-efficient Cloud computing.
- Investigate energy-aware resource provisioning and allocation algorithms that provision data center resources to client applications in a way that improves the energy efficiency of a data center, without violating the negotiated SLAs.
- Develop autonomic and energy-aware mechanisms for self-managing changes in the state of resources effectively and efficiently to satisfy service obligations and achieve energy efficiency.
- Develop algorithms for energy-efficient mapping of VMs to suitable Cloud resources in addition to dynamic consolidation of VM resource partitions.
- Explore open research challenges in energy-efficient resource management for virtualized Cloud data centers to facilitate advancements of the state-of-the-art operational Cloud environments.

The rest of the paper is organized as follows. Section 2 discusses related work, followed by the Green Cloud architecture and principles for energy-efficient Cloud computing presented in Section 3. The proposed energy-aware resource allocation algorithms are discussed in Section 4. A performance analysis of the proposed energy-aware resource provisioning and allocation algorithms is presented in Section 5. In Section 6 we discuss our vision on open research challenges in energy-efficient Cloud computing. Section 7 concludes the paper with summary and future research directions.

## 2. Related work

One of the first works, in which power management has been applied at the data center level, has been done by Pinheiro et al. [7]. In this work the authors have proposed a technique for minimization of power consumption in a heterogeneous cluster of computing nodes serving multiple web-applications. The main technique applied to minimize power consumption is concentrating the workload to the minimum of physical nodes and switching idle nodes off. This approach requires dealing with the power/performance trade-off, as performance of applications can be degraded due to the workload consolidation. Requirements to the throughput and execution time of applications are defined in SLAs to ensure reliable QoS. The proposed algorithm periodically monitors the load of resources (CPU, disk storage and network interface) and makes decisions on switching nodes on/off to minimize the overall power consumption, while providing the expected performance. The actual load balancing is not handled by the system and has to be managed by the applications. The algorithm runs on a master node, which creates a Single Point of Failure (SPF) and may become a performance bottleneck in a large system. In addition, the authors have pointed out that the reconfiguration operations are time-consuming, and the algorithm adds or removes only one node at a time, which may also be a reason for slow reaction in large-scale environments. The proposed approach can be applied to multi-application mixed-workload environments with fixed SLAs.

Chase et al. [8] have considered the problem of energy-efficient management of homogeneous resources in Internet hosting centers. The main challenge is to determine the resource demand of each application at its current request load level and to allocate resources in the most efficient way. To deal with this problem the authors have applied an economic framework: services “bid” for resources in terms of volume and quality. This enables negotiation of the SLAs according to the available budget and current QoS requirements, i.e. balancing the cost of resource usage (energy cost) and the benefit gained due to the usage of this resource. The system maintains an active set of servers selected to serve requests for each service. The network switches are dynamically reconfigured to change the active set of servers when necessary. Energy consumption is reduced by switching idle servers to power saving modes (e.g. sleep, hibernation). The system is targeted at the web workload, which leads to a “noise” in the load data. The authors have addressed this problem by applying the statistical “flip-flop” filter, which reduces the number of unproductive reallocations and leads to a more stable and efficient control. The proposed approach is suitable for multi-application environments with variable SLAs and has created a foundation for numerous studies on power-efficient resource allocation at the data center level. However, in contrast to [7], the system deals only with the management of the CPU, but does not consider other system resources. The latency due to switching nodes on/off also is not taken into account. The authors have noted that the management algorithm is fast when the workload is stable, but turns out to be relatively expensive during significant changes in the workload. Moreover, likewise [7], diverse software configurations are not handled, which can be addressed by applying the virtualization technology.

Elnozahy et al. [9] have investigated the problem of power-efficient resource management in a single web-application environment with fixed SLAs (response time) and load balancing handled by the application. As in [8], two power saving techniques are applied: switching power of computing nodes on/off and Dynamic Voltage and Frequency Scaling (DVFS). The main idea of the policy is to estimate the total CPU frequency required to provide the necessary response time, determine the optimal number

of physical nodes and set the proportional frequency to all the nodes. However, the transition time for switching the power of a node is not considered. Only a single application is assumed to be run in the system and, like in [7], the load balancing is supposed to be handled by an external system. The algorithm is centralized that creates an SPF and reduces the scalability. Despite the variable nature of the workload, unlike [8], the resource usage data are not approximated, which results in potentially inefficient decisions due to fluctuations. Nathuji and Schwan [10] have studied power management techniques in the context of virtualized data centers, which has not been done before. Besides hardware scaling and VMs consolidation, the authors have introduced and applied a new power management technique called “soft resource scaling”. The idea is to emulate hardware scaling by providing less resource time for a VM using the Virtual Machine Monitor’s (VMM) scheduling capability. The authors found that a combination of “hard” and “soft” scaling may provide higher power savings due to the limited number of hardware scaling states. The authors have proposed an architecture where the resource management is divided into local and global policies. At the local level the system leverages the guest OS’s power management strategies. However, such management may appear to be inefficient, as the guest OS may be legacy or power-unaware.

Raghavendra et al. [11] have investigated the problem of power management for a data center environment by combining and coordinating five diverse power management policies. The authors explored the problem in terms of control theory and applied a feedback control loop to coordinate the controllers’ actions. It is claimed that, similarly to [10], the approach is independent of the workload type. Like most of the previous works, the system deals only with the CPU management. The authors have pointed out an interesting outcome of the experiments: the actual power savings can vary depending on the workload, but “the benefits from coordination are qualitatively similar for all classes of workloads”. However, the system fails to support strict SLAs as well as variable SLAs for different applications. This results in the suitability for enterprise environments, but not for Cloud computing providers, where more comprehensive support for SLAs is essential. Kusic et al. [12] have defined the problem of power management in virtualized heterogeneous environments as a sequential optimization and addressed it using Limited Lookahead Control (LLC). The objective is to maximize the resource provider’s profit by minimizing both power consumption and SLA violation. Kalman filter is applied to estimate the number of future requests to predict the future state of the system and perform necessary reallocations. However, in contrast to heuristic-based approaches, the proposed model requires simulation-based learning for the application specific adjustments. Moreover, due to the complexity of the model the execution time of the optimization controller reaches 30 min even for 15 nodes, which is not suitable for large-scale real-world systems. On the contrary, our approach is heuristic-based allowing the achievement of reasonable performance even for a large scale as shown in our experimental study.

Srikantaiah et al. [13] have studied the problem of request scheduling for multi-tiered web-applications in virtualized heterogeneous systems to minimize energy consumption, while meeting performance requirements. The authors have investigated the effect of performance degradation due to high utilization of different resources when the workload is consolidated. They have found that the energy consumption per transaction results in a “U”-shaped curve, and it is possible to determine the optimal utilization point. To handle the optimization over multiple resources, the authors have proposed a heuristic for the multidimensional bin packing problem as an algorithm for the workload consolidation. However, the proposed approach is workload type and application dependent, whereas our algorithms are independent of

the workload type, and thus are suitable for a generic Cloud environment. Cardoso et al. [14] have proposed an approach for the problem of power-efficient allocation of VMs in virtualized heterogeneous computing environments. They have leveraged the min, max and shares parameters of VMM, which represent minimum, maximum and proportion of the CPU allocated to VMs sharing the same resource. Similarly to [11], the approach suits only enterprise environments as it does not support strict SLAs and requires the knowledge of application priorities to define the shares parameter. Other limitations are that the allocation of VMs is not adapted at run-time (the allocation is static) and no other resources except for the CPU are considered during the VM reallocation.

Verma et al. [15] have formulated the problem of power-aware dynamic placement of applications in virtualized heterogeneous systems as continuous optimization: at each time frame the placement of VMs is optimized to minimize power consumption and maximize performance. Like in [13], the authors have applied a heuristic for the bin packing problem with variable bin sizes and costs. Similarly to [10], live migration of VMs is used to achieve a new placement at each time frame. The proposed algorithms, on the contrary to our approach, do not handle strict SLA requirements: SLAs can be violated due to variability of the workload. Gandhi et al. [16] have considered the problem of allocating an available power budget among servers in a virtualized heterogeneous server farm, while minimizing the mean response time. To investigate the effect of different factors on mean response time, a queuing theoretic model has been introduced, which allows the prediction of the mean response time as a function of the power-to-frequency relationship, arrival rate, peak power budget, etc. The model is used to determine the optimal power allocation for every configuration of the above factors.

In contrast to the discussed studies, we propose efficient heuristics for dynamic adaption of VM allocation at run-time according to the current utilization of resources applying live migration, switching idle nodes to the sleep mode, and thus minimizing energy consumption. The proposed approach can effectively handle strict SLAs, heterogeneous infrastructure and heterogeneous VMs. The algorithms do not depend on a particular type of workload and do not require any knowledge about applications running in VMs.

Another resource that has been recognized by the research community as a significant energy consumer is network infrastructure. Gupta et al. [17] have suggested putting network interfaces, links, switches and routers into sleep modes when they are idle in order to save the energy consumed by the Internet backbone and consumers. Based on the foundation laid by Gupta et al. [17], a number of research works have been done on the energy-efficient traffic routing by ISPs and applying sleep modes and performance scaling of network devices [18,19]. Chiaraviglio and Matta [20] have proposed cooperation between ISPs and content providers that allows the achievement of an efficient simultaneous allocation of compute resources and network paths that minimizes energy consumption under performance constraints. Koseoglu and Karasan [21] have applied a similar approach of joint allocation of computational resources and network paths to Grid environments based on the optical burst switching technology with the objective of minimizing job completion times. Tomas et al. [22] have investigated the problem of scheduling Message Passing Interface (MPI) jobs in Grids considering network data transfers satisfying the QoS requirements.

Dodonov and de Mello [23] have proposed an approach to scheduling distributed applications in Grids based on predictions of communication events. They have proposed the migration of communicating processes if the migration cost is lower than the cost of the predicted communication with the objective of minimizing the total execution time. They have shown that the approach can be effectively applied in Grids; however, it is not viable for virtualized data centers, as the VM migration cost is higher



than the process migration cost. Gyarmati and Trinh [24] have investigated the energy consumption implications of data centers' network architectures. However, optimization of network architectures can be applied only at the data center design time and cannot be applied dynamically. Guo et al. [25] have proposed and implemented a virtual cluster management system that allocates the resources in a way satisfying bandwidth guarantees. The allocation is determined by a heuristic that minimizes the total bandwidth utilized. The VM allocation is adapted when some of the VMs are de-allocated. However, the VM allocation is not dynamically adapted depending on the current network load. Moreover, the approach does not explicitly minimize energy consumption by the network. Roderio-Merino et al. [26] have proposed an additional layer of infrastructure management in Clouds with the ability to automatically deploy services with multi-VM configurations. The proposed infrastructure management system applies the specified rules for scaling VM configurations in and out. However, the system does not optimize the network communication between VMs. Calheiros et al. [27] have investigated the problem of mapping VMs on physical nodes optimizing network communication between VMs; however, the problem has not been explored in the context of the optimization of energy consumption.

Recently, a number of research works have been done on the thermal-efficient resource management in data centers [28,29]. The studies have shown that the software-driven thermal management and temperature-aware workload placement bring additional energy savings. However, the problem of thermal management in the context of virtualized data centers has not been investigated. Moreover, to the best of our knowledge there is no study on a comprehensive approach that combines optimization of VM placement according to the current utilization of resources with thermal and network optimizations for virtualized data centers. Therefore, the exploration of such an approach is timely and crucial considering the proliferation of Cloud computing environments.

### 3. Green cloud architecture

#### 3.1. Architectural framework

Clouds aim to drive the design of the next generation data centers by architecting them as networks of virtual services (hardware, database, user-interface, application logic) so that users can access and deploy applications from anywhere in the world on demand at competitive costs depending on their QoS requirements [30]. Fig. 1 shows the high-level architecture for supporting energy-efficient service allocation in a Green Cloud computing infrastructure [6]. There are basically four main entities involved:

1. *Consumers/Brokers*: Cloud consumers or their brokers submit service requests from anywhere in the world to the Cloud. It is important to notice that there can be a difference between Cloud consumers and users of deployed services. For instance, a consumer can be a company deploying a web-application, which presents varying workload according to the number of "users" accessing it.
2. *Green Service Allocator*: Acts as the interface between the Cloud infrastructure and consumers. It requires the interaction of the following components to support the energy-efficient resource management:
  - (a) *Green Negotiator*: Negotiates with the consumers/brokers to finalize the SLAs with specified prices and penalties (for violations of the SLAs) between the Cloud provider and consumer depending on the consumer's QoS requirements and energy saving schemes. In case of web-applications, for instance, a QoS metric can be 95% of requests being served in less than 3 s.

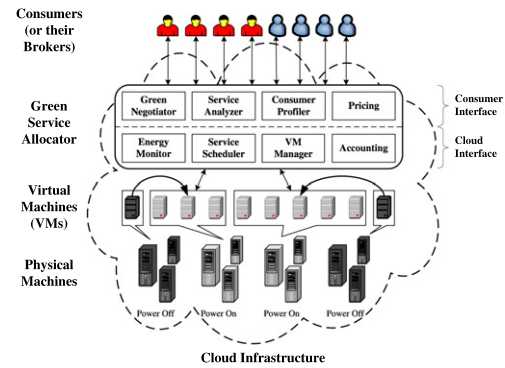


Fig. 1. The high-level system architecture.

- (b) *Service Analyzer*: Interprets and analyzes the service requirements of a submitted request before accepting it. Hence, it needs the latest load and energy information from VM Manager and Energy Monitor respectively.
  - (c) *Consumer Profiler*: Gathers specific characteristics of consumers so that important consumers can be granted special privileges and prioritized over other consumers.
  - (d) *Pricing*: Decides how service requests are charged to manage the supply and demand of computing resources and facilitate in prioritizing service allocations effectively.
  - (e) *Energy Monitor*: Observes energy consumption caused by VMs and physical machines and provides this information to the VM manager to make energy-efficient resource allocation decisions.
  - (f) *Service Scheduler*: Assigns requests to VMs and determines resource entitlements for the allocated VMs. If the auto-scaling functionality has been requested by a customer, it also decides when VMs are to be added or removed to meet the demand.
  - (g) *VM Manager*: Keeps track of the availability of VMs and their resource usage. It is in charge of provisioning new VMs as well as reallocating VMs across physical machines to adapt the placement.
  - (h) *Accounting*: Monitors the actual usage of resources by VMs and accounts for the resource usage costs. Historical data of the resource usage can be used to improve resource allocation decisions.
3. *VMs*: Multiple VMs can be dynamically started and stopped on a single physical machine according to incoming requests, hence providing the flexibility of configuring various partitions of resources on the same physical machine to different requirements of service requests. Multiple VMs can concurrently run applications based on different operating system environments on a single physical machine. By dynamically migrating VMs across physical machines, workloads can be consolidated and unused resources can be switched to a low-power mode, turned off or configured to operate at low-performance levels (e.g. using DVFS) in order to save energy.
  4. *Physical Machines*: The underlying physical computing servers provide the hardware infrastructure for creating virtualized resources to meet service demands.

#### 3.2. Power model

Power consumption by computing nodes in data centers is mostly determined by the CPU, memory, disk storage and network interfaces. In comparison to other system resources, the CPU consumes the main part of energy, and hence in this work we focus on managing its power consumption and efficient usage. Moreover, the CPU utilization is typically proportional to the overall system load.

Recent studies [16,12,11,15] have shown that the application of DVFS on the CPU results in almost linear power-to-frequency

relationship for a server. The reason lies in the limited number of states that can be set to the frequency and voltage of the CPU and the fact that DVFS is not applied to other system components apart from the CPU. Moreover, these studies have shown that on average an idle server consumes approximately 70% of the power consumed by the server running at the full CPU speed. This fact justifies the technique of switching idle servers to the sleep mode to reduce the total power consumption. Therefore, in this work we use the power model defined in (1).

$$P(u) = k \cdot P_{\max} + (1 - k) \cdot P_{\max} \cdot u, \quad (1)$$

$P_{\max}$  is the maximum power consumed when the server is fully utilized;  $k$  is the fraction of power consumed by the idle server (i.e. 70%); and  $u$  is the CPU utilization. For our experiments  $P_{\max}$  is set to 250 W, which is a usual value for modern servers. For example, according to the SPECpower benchmark, for the fourth quarter of 2010, the average power consumption at 100% utilization for servers consuming less than 1000 W was approximately 259 W.<sup>1</sup>

The utilization of the CPU may change over time due to the workload variability. Thus, the CPU utilization is a function of time and is represented as  $u(t)$ . Therefore, the total energy consumption by a physical node ( $E$ ) can be defined as an integral of the power consumption function over a period of time as shown in (2).

$$E = \int_{t_0}^{t_1} P(u(t))dt. \quad (2)$$

#### 4. Energy-aware allocation of data center resources

Recent developments in virtualization have resulted in its proliferation across data centers. By supporting the movement of VMs between physical nodes, it enables dynamic migration of VMs according to the performance requirements. When VMs do not use all the provided resources, they can be logically resized and consolidated to the minimum number of physical nodes, while idle nodes can be switched to the sleep mode to eliminate the idle power consumption and reduce the total energy consumption by the data center.

Currently, resource allocation in a Cloud data center aims to provide high performance while meeting SLAs, without focusing on allocating VMs to minimize energy consumption. To explore both performance and energy efficiency, three crucial issues must be addressed. First, excessive power cycling of a server could reduce its reliability. Second, turning resources off in a dynamic environment is risky from the QoS perspective. Due to the variability of the workload and aggressive consolidation, some VMs may not obtain required resources under peak load, and fail to meet the desired QoS. Third, ensuring SLAs brings challenges to accurate application performance management in virtualized environments. All these issues require effective consolidation policies that can minimize energy consumption without compromising the user-specified QoS requirements.

##### 4.1. VM placement

The problem of VM allocation can be divided in two: the first part is the admission of new requests for VM provisioning and placing the VMs on hosts, whereas the second part is the optimization of the current VM allocation. The first part can be seen as a bin packing problem with variable bin sizes and prices. To solve it we apply a modification of the Best Fit Decreasing

(BFD) algorithm that is shown to use no more than  $\frac{11}{9} \cdot OPT + 1$  bins (where  $OPT$  is the number of bins given by the optimal solution) [31]. In our modification, the Modified Best Fit Decreasing (MBFD) algorithms, we sort all VMs in decreasing order of their current CPU utilizations, and allocate each VM to a host that provides the least increase of power consumption due to this allocation. This allows leveraging the heterogeneity of resources by choosing the most power-efficient nodes first. The pseudo-code for the algorithm is presented in Algorithm 1. The complexity of the allocation part of the algorithm is  $n \cdot m$ , where  $n$  is the number of VMs that have to be allocated and  $m$  is the number of hosts.

---

#### Algorithm 1: Modified Best Fit Decreasing (MBFD)

---

```

1 Input: hostList, vmList Output: allocation of VMs
2 vmList.sortDecreasingUtilization()
3 foreach vm in vmList do
4   minPower ← MAX
5   allocatedHost ← NULL
6   foreach host in hostList do
7     if host has enough resource for vm then
8       power ← estimatePower(host, vm)
9       if power < minPower then
10        allocatedHost ← host
11        minPower ← power
12 if allocatedHost ≠ NULL then
13   allocate vm to allocatedHost
14 return allocation

```

---

##### 4.2. VM selection

The optimization of the current VM allocation is carried out in two steps: at the first step we select VMs that need to be migrated, at the second step the chosen VMs are placed on the hosts using the MBFD algorithm. To determine when and which VMs should be migrated, we introduce three double-threshold VM selection policies. The basic idea is to set upper and lower utilization thresholds for hosts and keep the total utilization of the CPU by all the VMs allocated to the host between these thresholds. If the CPU utilization of a host falls below the lower threshold, all VMs have to be migrated from this host and the host has to be switched to the sleep mode in order to eliminate the idle power consumption. If the utilization exceeds the upper threshold, some VMs have to be migrated from the host to reduce the utilization. The aim is to preserve free resources in order to prevent SLA violations due to the consolidation in cases when the utilization by VMs increases. The difference between the old and new placements forms a set of VMs that have to be reallocated. The new placement is achieved using live migration of VMs [32]. In the following sections we discuss the proposed VM selection policies.

##### 4.2.1. The minimization of migrations policy

The Minimization of Migrations (MM) policy selects the minimum number of VMs needed to migrate from a host to lower the CPU utilization below the upper utilization threshold if the upper threshold is violated. Let  $V_j$  be a set of VMs currently allocated to the host  $j$ . Then  $\mathcal{P}(V_j)$  is the power set of  $V_j$ . The MM policy finds a set  $R \in \mathcal{P}(V_j)$  defined in (3).

$$R = \begin{cases} \left\{ S | S \in \mathcal{P}(V_j), u_j - \sum_{v \in S} u_a(v) < T_u, \right. \\ \left. |S| \rightarrow \min \right\}, & \text{if } u_j > T_u; \\ V_j, & \text{if } u_j < T_l; \\ \emptyset, & \text{otherwise} \end{cases} \quad (3)$$

<sup>1</sup> The SPECpower benchmark results for the fourth quarter of 2010. [http://www.spec.org/power\\_ssj2008/results/res2010q4/](http://www.spec.org/power_ssj2008/results/res2010q4/).

where  $T_u$  is the upper utilization threshold;  $T_l$  is the lower utilization threshold;  $u_j$  is the current CPU utilization of the host  $j$ ; and  $u_a(v)$  is the fraction of the CPU utilization allocated to the VM  $v$ .

The pseudo-code for the MM algorithm for the over-utilization case is presented in Algorithm 2. The algorithm sorts the list of VMs in the decreasing order of the CPU utilization. Then, it repeatedly looks through the list of VMs and finds a VM that is the best to migrate from the host. The best VM is the one that satisfies two conditions. First, the VM should have the utilization higher than the difference between the host's overall utilization and the upper utilization threshold. Second, if the VM is migrated from the host, the difference between the upper threshold and the new utilization is the minimum across the values provided by all the VMs. If there is no such a VM, the algorithm selects the VM with the highest utilization, removes it from the list of VMs, and proceeds to a new iteration. The algorithm stops when the new utilization of the host is below the upper utilization threshold. The complexity of the algorithm is proportional to the product of the number of over-utilized hosts and the number of VMs allocated to these hosts.

---

**Algorithm 2:** Minimization of Migrations (MM)

---

```

1 Input: hostList Output: migrationList
2 foreach  $h$  in hostList do
3   vmList  $\leftarrow$  h.getVmList()
4   vmList.sortDecreasingUtilization()
5   hUtil  $\leftarrow$  h.getUtil()
6   bestFitUtil  $\leftarrow$  MAX
7   while hUtil > THRESH_UP do
8     foreach vm in vmList do
9       if vm.getUtil() > hUtil - THRESH_UP then
10         t  $\leftarrow$  vm.getUtil() - hUtil + THRESH_UP
11         if t < bestFitUtil then
12           bestFitUtil  $\leftarrow$  t
13           bestFitVm  $\leftarrow$  vm
14       else
15         if bestFitUtil = MAX then
16           bestFitVm  $\leftarrow$  vm
17       break
18   hUtil  $\leftarrow$  hUtil - bestFitVm.getUtil()
19   migrationList.add(bestFitVm)
20   vmList.remove(bestFitVm)
21   if hUtil < THRESH_LOW then
22     migrationList.add(h.getVmList())
23     vmList.remove(h.getVmList())
24 return migrationList

```

---

#### 4.2.2. The highest potential growth policy

When the upper threshold is violated, the Highest Potential Growth (HPG) policy migrates VMs that have the lowest usage of the CPU relatively to the CPU capacity defined by the VM parameters in order to minimize the potential increase of the host's utilization and prevent an SLA violation, as formalized in (4).

$$R = \begin{cases} \left\{ S \mid S \in \mathcal{P}(V_j), u_j - \sum_{v \in S} u_a(v) < T_u, \right. \\ \left. \sum_{v \in S} \frac{u_a(v)}{u_r(v)} \rightarrow \min \right\}, & \text{if } u_j > T_u; \\ V_j, & \text{if } u_j < T_l; \\ \emptyset, & \text{otherwise} \end{cases} \quad (4)$$

where  $u_r(v)$  is the fraction of the CPU capacity initially requested for the VM  $v$  and defined as the VM's parameter. We do not provide the pseudo-code for the HPG algorithm, as it is similar to the MM algorithm presented earlier.

#### 4.2.3. The random choice policy

The Random Choice (RC) policy relies on a random selection of a number of VMs needed to decrease the CPU utilization by a host below the upper utilization threshold. According to a uniformly distributed discrete random variable ( $X$ ), whose values index subsets of  $V_j$ , the policy selects a set  $R \in \mathcal{P}(V_j)$ , as shown in (5).

$$R = \begin{cases} \left\{ S \mid S \in \mathcal{P}(V_j), u_j - \sum_{v \in S} u_a(v) < T_u, \right. \\ \left. X \stackrel{d}{=} U(0, |\mathcal{P}(V_j)| - 1) \right\}, & \text{if } u_j > T_u; \\ V_j, & \text{if } u_j < T_l; \\ \emptyset, & \text{otherwise} \end{cases} \quad (5)$$

where  $X$  is a uniformly distributed discrete random variable used to select a subset of  $V_j$ .

The results of a simulation-based evaluation of the proposed algorithms in terms of power consumption, SLA violations and the number of VM migrations are presented in Section 5.

### 5. Performance analysis

In this section, we discuss a performance analysis of the energy-aware allocation heuristics presented in Section 4. In our experiments, we calculate the time needed to perform a live migration of a VM as the size of its memory divided by the available network bandwidth. This is justified as to enable live migration, the images and data of VMs must be stored on a Network Attached Storage (NAS); and therefore, copying the VM's storage is not required. Live migration creates an extra CPU load; however, it has been shown that the performance overhead is low [33]. Moreover, with advancements of the virtualization technology, the efficiency of VM migration is going to be improved. For the simulations, the utilization of the CPU by a VM is generated as a uniformly distributed random variable. This is appropriate due to unknown types of applications running on VMs, and as it is not possible to build the exact model of such a mixed workload. This approach has been justified by Verma et al. [15].

#### 5.1. Performance metrics

In order to compare the efficiency of the algorithms we use several metrics to evaluate their performance. The first metric is the total energy consumption by the physical resources of a data center caused by the application workloads. The second performance metric is called the SLA violation percentage, or simply the SLA violations, which is defined as the percentage of SLA violation events relatively to the total number of the processed time frames. We define that an SLA violation occurs when a given VM cannot get the amount of Million Instructions Per Second (MIPS) that are requested. This can happen in cases when VMs sharing the same host require a CPU performance that cannot be provided due to the consolidation. This metric shows the level by which the QoS requirements negotiated between the resource provider and consumers are violated due to the energy-aware resource management. It is assumed that the provider pays a penalty to the client in case of an SLA violation. The third metric is the number of VM migrations initiated by the VM manager during the adaptation of the VM placement. The last performance metric is the average SLA violation, which represents the average CPU performance that has not been allocated to an application when requested, resulting in performance degradation.



## 5.2. Experiment setup

As the targeted system is a generic Cloud computing environment, i.e. IaaS, it is essential to evaluate it on a large-scale virtualized data center infrastructure. However, it is extremely difficult to conduct repeatable large-scale experiments on a real infrastructure, which is required to evaluate and compare the proposed resource management algorithms. Therefore, to ensure the repeatability of experiments, simulations have been chosen as a way to evaluate the performance of the proposed heuristics. The CloudSim toolkit [34] has been chosen as a simulation platform as it is a modern simulation framework aimed at Cloud computing environments. In contrast to alternative simulation toolkits (e.g. SimGrid, GangSim), it supports modeling of on-demand virtualization-enabled resource and application management. It has been extended to enable energy-aware simulations as the core framework does not provide this capability. Apart from the energy consumption modeling and accounting, the ability to simulate service applications with workloads variable over time has been incorporated.

We have simulated a data center comprising 100 heterogeneous physical nodes. Each node is modeled to have one CPU core with the performance equivalent to 1000, 2000 or 3000 MIPS, 8 GB of RAM and 1 TB of storage. Power consumption by the hosts is defined according to the model described in Section 3.2. According to this model, a host consumes from 175 W with 0% CPU utilization, up to 250 W with 100% CPU utilization. Each VM requires one CPU core with 250, 500, 750 or 1000 MIPS, 128 MB of RAM and 1 GB of storage. The users submit requests for provisioning of 290 heterogeneous VMs that fill the full capacity of the simulated data center. Each VM runs a web-application or any kind of application with variable workload, which is modeled to generate the utilization of CPU according to a uniformly distributed random variable. The application runs for 150,000 MI that is equal to 10 min of the execution on 250 MIPS CPU with 100% utilization. Initially, the VMs are allocated according to the requested characteristics assuming 100% CPU utilization. Each experiment has been run 10 times.

## 5.3. Simulation results

For the benchmark experimental results we have used the Non Power-Aware (NPA) policy. This policy does not apply any power-aware optimizations and implies that all hosts run at 100% CPU utilization and consume maximum power all the time. The second policy applies DVFS, but does not perform any adaptation of the VM allocation at run-time. For the described simulation setup, the NPA policy leads to the total energy consumption of 9.15 kWh with 95% Confidence Interval (CI): (9.04, 9.27), whereas DVFS decreases this value to 4.40 kWh with 95% CI: (4.39, 4.41).

Another benchmark VM selection policy is a VM migration-aware policy called Single Threshold (ST). It is based on the idea of setting the upper utilization threshold for hosts and placing VMs, while keeping the total utilization of CPU below this threshold. At each time frame all VMs are reallocated using the MBFD algorithm with additional condition of keeping the upper utilization threshold not violated. To evaluate the ST policy we have conducted several experiments with different values of the utilization threshold. The simulation results are presented in Fig. 2. The results show that energy consumption can be significantly reduced relatively to the NPA and DVFS policies—by 77% and 53% respectively with 5.4% of SLA violations. The results show that with the growth of the utilization threshold energy consumption decreases, whereas the percentage of SLA violations increases. This is due to the fact that a higher utilization threshold allows more aggressive consolidation of VMs by the cost of the increased risk of SLA violations.

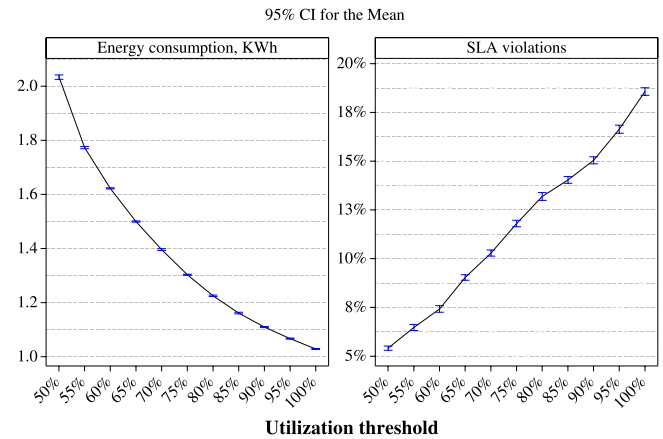


Fig. 2. The energy consumption and SLA violations by the ST policy.

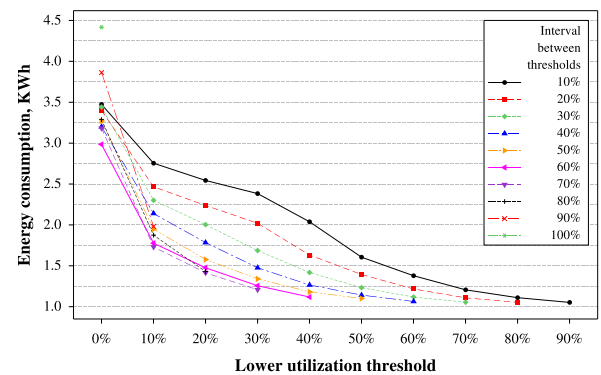


Fig. 3. The mean energy consumption by the MM policy for different values of the utilization thresholds.

To evaluate the double-threshold policies it is necessary to determine the best values for the thresholds in terms of the energy consumption and QoS delivered. We have chosen the MM policy to conduct the analysis of the utilization thresholds. We have simulated the MM policy varying the absolute values of the lower and upper thresholds as well as the interval between them. First of all, it is important to determine which threshold has higher influence on the energy consumption. We have performed a regression analysis of the relationship between the energy consumption and values of the utilization thresholds. To achieve the normality of the residuals we have applied the  $\log(\log(X))$  transformation. The Ryan–Joiner normality test has resulted in the  $P$ -Value  $> 0.1$ . The regression analysis has shown the adjusted  $R^2$  is 86.6%, the coefficient for the lower threshold is  $-2.79$ , and the coefficient for the upper threshold is  $-1.28$ . The value of the adjusted  $R^2$  shows that the obtained regression represents the relationship with a high precision. The values of the coefficients show that the lower threshold has approximately two times higher influence on the energy consumption than the upper threshold. This can be explained by the fact that an increase of the lower threshold eliminates the low utilization of the resources leading to higher energy savings; however, possibly increasing the number of VM migrations and SLA violations.

The results showing the mean energy consumption achieved using the MM policy for different values of the lower utilization threshold and the interval between the thresholds are presented in Fig. 3. The graph shows that an increase of the lower utilization threshold leads to decreased energy consumption. However, the low level of energy consumption can be achieved with different intervals between the thresholds. Therefore, to determine the best interval we have to consider another factor—the level of SLA violations.

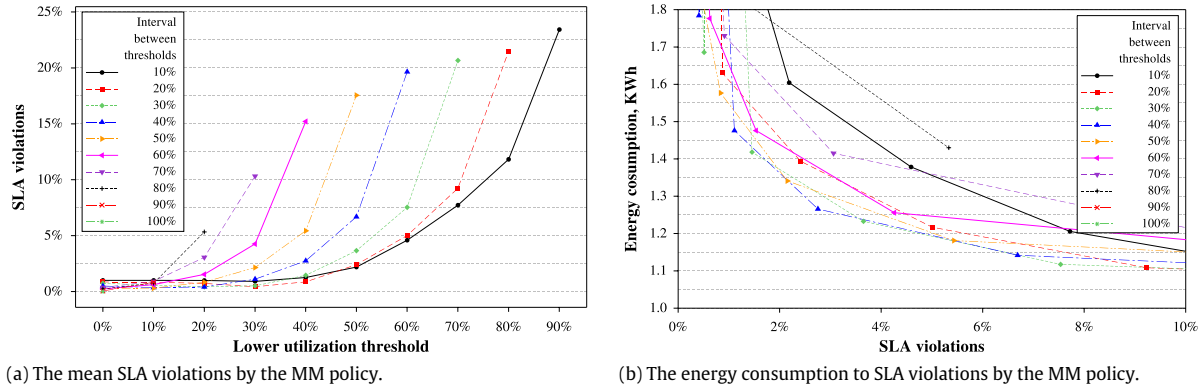


Fig. 4. The energy consumption and SLA violation analysis for the MM policy.

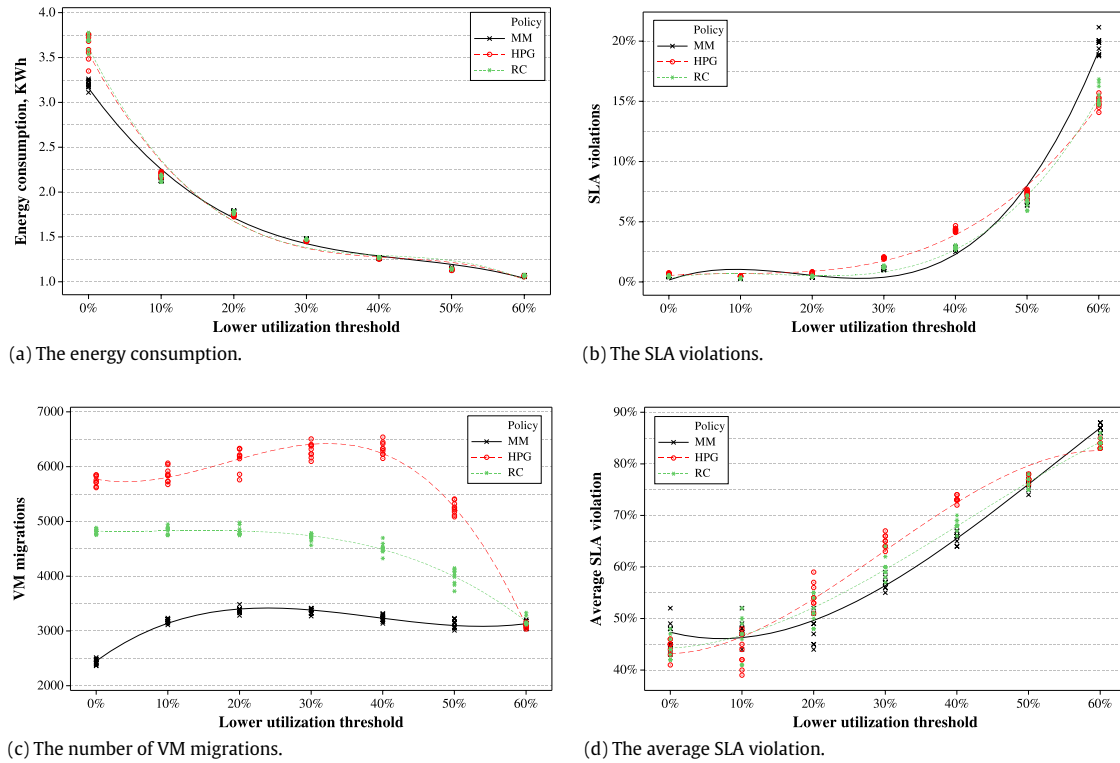


Fig. 5. The comparison of the double-threshold algorithms.

The results presented in Fig. 4(a) show that an increase of the lower utilization threshold leads to an increase of the SLA violations level for all the intervals. Therefore, it is necessary to determine the interval that will provide the optimal trade-off between energy consumption and SLA violations. Fig. 4(b) shows that for the considered simulation setup the best combination of energy consumption and SLA violations is achieved with the 40% interval between the thresholds.

We have used the interval between the thresholds of 40% with different values of the lower utilization threshold to compare the MM policy with the HPG and RC policies. The graphs with fitted lines of the energy consumption, SLA violations, number of VM migration and average SLA violation achieved by the policies with the 40% interval between the thresholds are presented in Fig. 5. To compare the policies by each of these factors we have performed a two-way ANOVA test. To meet the assumptions of the model we have transformed the energy consumption using  $\log(\log(X))$  transformation and the SLA violations using  $\log(X)$ . After the transformation the residuals are normally distributed with  $P$ -Value

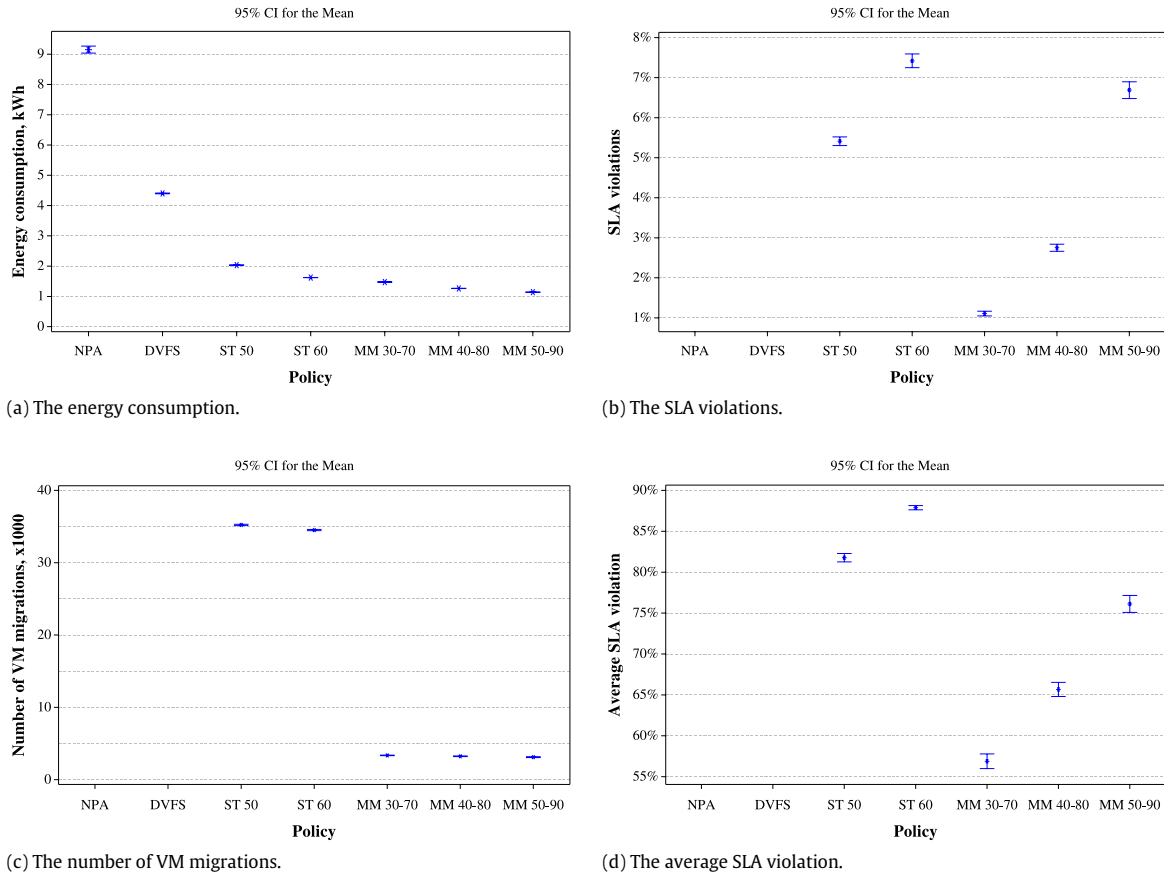
$>0.1$ . The variances of the distributions are approximately equal according to the plot of the standardized residuals against the fitted values. According to the results of the two-way ANOVA test, the data are consistent with the null hypothesis as  $P$ -Value = 0.16. Therefore, we can conclude that all the policies produce results with a not statistically significant difference.

For the comparison of SLA violations for the double-threshold policies we get  $P$ -value = 0.005 that allow us to conclude that the data are not consistent with the null hypothesis. Using pair wise comparisons, we have found that the difference between the results obtained using the RC and MM policies is not statistically significant ( $P$ -Value  $>0.9$ ). However, the  $P$ -Value for the difference between the MM and HPG policies is 0.01, and for the difference between RC and HPG the  $P$ -Value is 0.009. The difference of means for the MM and HPG policies is  $-0.297$  with 95% CI:  $(-0.526, -0.068)$ . The difference of means for the RC and HPG policies is  $-0.312$  with 95% CI:  $(-0.540, -0.083)$ . This means that the MM and RC policies lead to significantly less SLA violations than the HPG policy.



**Table 1**  
The final experiment results.

Algorithm	Energy, kWh	SLA violation	VM migrations	Average SLA
NPA	9.151 (9.036, 9.265)	–	–	–
DVFS	4.402 (4.390, 4.414)	–	–	–
ST 50%	2.034 (2.026, 2.042)	5.4% (5.3%, 5.5%)	35 226 (35 135, 35 317)	81.8% (81.3%, 82.3%)
ST 60%	1.622 (1.620, 1.625)	7.4% (7.2%, 7.6%)	34 519 (34 438, 34 600)	87.9% (87.6%, 88.1%)
MM 30%–70%	1.476 (1.472, 1.480)	1.1% (1.0%, 1.2%)	3360 (3322, 3397)	56.9% (56.0%, 57.8%)
MM 40%–80%	1.266 (1.262, 1.270)	2.8% (2.7%, 2.8%)	3241 (3194, 3288)	65.7% (64.8%, 66.5%)
MM 50%–90%	1.142 (1.134, 1.150)	6.7% (6.5%, 6.9%)	3121 (3054, 3188)	76.1% (75.1%, 77.2%)



**Fig. 6.** The final experiment results.

For the analysis of the number of VM migrations the extreme point for the lower utilization threshold at 60% and the 40% interval has been removed. The reason is that the usage of these utilization thresholds leads to the same behavior of all the evaluated policies, as at 100% of the upper utilization threshold there is no space to handle peak loads. The two-way ANOVA test for the comparison of the number of VM migrations gives a statistically significant result with  $P$ -Value  $< 0.001$ . According to the pair wise comparisons, the usage of the RC policy results in the mean of 1322 fewer VM migrations than the HPG policy with 95% CI: (848, 1795). The usage of the MM policy results in the mean of 1495 fewer VM migrations than the RC policy with 95% CI: (1022, 1968). From these results we can conclude that the MM policy leads to a significant reduction of the number of VM migrations compared to the other double-threshold policies.

A two-way ANOVA test for the comparison of the average SLA violation gives a not statistically significant result with  $P$ -Value = 0.417. This means that we accept the null hypothesis: the difference in results that can be obtained using the MM, HPG and RC policies is negligible.

In summary, all the evaluated policies cause approximately the same energy consumption and average value of SLA violation.

However, the MM and RC policies lead to significantly less SLA violations than the HPG policy. Moreover, the usage of the MM policy results in a significantly reduced number of VM migrations in comparison to the other policies. Therefore, we have chosen the MM policy as the best among the double-threshold policies. We have chosen three representative threshold pairs for the MM policy and two values of the threshold for the ST policy to conduct a final comparison. The mean values of the energy consumption, SLA violations, number of migration and average SLA violation along with 95% CI for the NPA, DVFS, ST and MM policies are presented in Table 1 and in Fig. 6.

The results show that the dynamic reallocation of VMs according to the current CPU utilization can bring higher energy savings in comparison to static resource allocation policies. According to the  $T$ -test, for the simulated scenario the MM 50%–90% policy leads to 0.48 kWh less energy consumption on average than ST 60% with approximately the same level of SLA violations with 95% CI: (0.47, 0.49). Moreover, the MM policy leads to more than 10 times less VM migrations than ST 60%. The  $T$ -test for the comparison of means of the energy consumption caused by the MM and DVFS policies show that the MM policy on average leads to 3.26 kWh less energy

consumption than DVFS with 95% CI: (3.25, 3.27). In comparison to the NPA policy, for the simulated scenario MM on average leads to 8.00 kWh less energy consumption than NPA with 95% CI: (7.89, 8.12).

From the presented results we can conclude that the usage of the MM policy provides the best energy savings with the least SLA violations and number of VM migrations among the evaluated policies for the simulated scenario. Moreover, the results show the flexibility of the MM algorithm, as the thresholds can be adjusted according to the SLAs requirements. 1.1% (1.0%, 1.2%) of the SLA violations allows the achievement of the energy consumption of 1.476 kWh (1.472, 1.480). However, if the SLAs are relaxed, for example allowing 6.7% (6.5%, 6.9%) performance degradation, the energy consumption is further reduced to 1.142 kWh (1.134, 1.150).

According to our model, the service provider pays a penalty to the client in cases of SLA violations. The actual penalty amount depends on the contract terms negotiated between the provider and the client. Nevertheless, the energy savings of 66% achieved using the MM policy in comparison to the non-migration aware DVFS policy justify the penalty caused by 1.1% SLA violations. Moreover, the performance requirements set as the SLAs for our experiments imply the 100% performance delivery. In a real-world environment, the provider can define in the contract terms the allowed mean performance degradation of 1%–5%, and thus avoiding the penalty if the performance degradation does not exceed the specified value.

#### 5.4. Transitions to the sleep mode

We have collected the data on the number of times the hosts have been switched to the sleep mode caused by the proposed MM algorithm during the simulations. The distribution of the number of transitions obtained from 10 simulation runs is depicted in Fig. 7.

According to the data, the experiment setup described in Section 5.2 has led to the mean of 201.5 transitions to the sleep mode with 95% CI: (188.62, 214.38). The distribution of the time duration until a host is switched to the sleep mode is shown in Fig. 8.

The data show that the mean time before a host is switched to the sleep mode is 20.1 s with 95% CI: (18.6, 21.6). In other words, for our experiment setup and the workload generated, on average a host switches to the sleep mode after approximately 20 s of being active. This value is effective for real-world systems, as modern servers allow low-latency transitions to the sleep mode consuming low power. Meisner et al. [35] have shown that a typical blade server consuming 450 W in the fully utilized state consumes approximately 10.4 W in the sleep mode, while the transition delay is 300 ms.

## 6. Open challenges

The virtualization technology, which Cloud computing environments heavily rely on, provides the ability to transfer VMs between physical nodes using live or offline migration. This enables the technique of dynamic consolidation of VMs to the minimum of physical nodes according to the current resource requirements. As a result, the idle nodes can be switched off or put to a power saving mode (e.g. sleep, hibernate) to reduce the total energy consumption by the data center. In this paper we have proposed algorithms that leverage this technique showing its efficiency. However, there are many open challenges that have to be addressed in order to take advantage of the full potential of energy conservation in Cloud data centers. In this section we identify and discuss key open research challenges that should be addressed at the level of managing data center's resources.

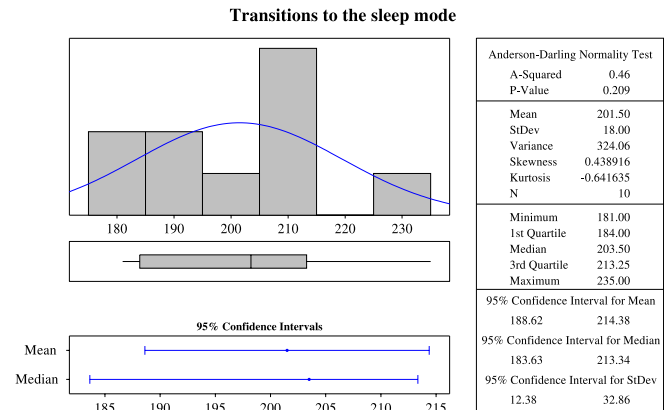


Fig. 7. The distribution of the number of host transition to the sleep mode.

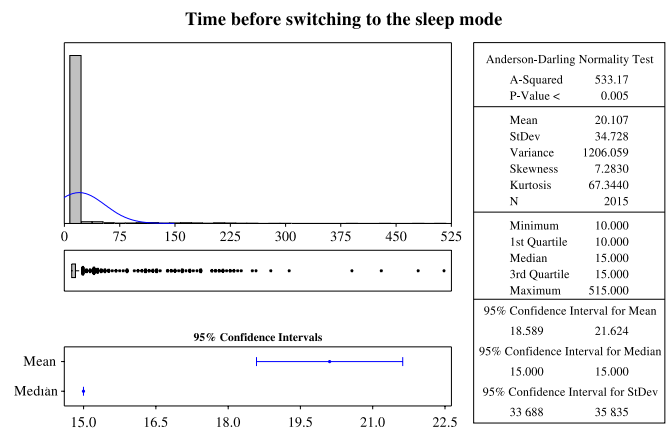


Fig. 8. The distribution of the time until a host is switched to the sleep mode.

### 6.1. Optimization of VM placement according to the utilization of multiple system resources

The CPU consumes the major part of power in a server followed by the next largest power consumer—memory. However, modern multi-core processors are much more power-efficient than previous generations, whereas the memory technology does not show any significant improvements in energy efficiency. The increased number of cores in servers combined with the rapid adoption of virtualization technologies creates the ever growing demand to memory and makes memory one of the most important components of focus in the power and energy usage optimization [36]. The same applies to network and disk storage facilities in modern data centers. These facts unveil that it is essential to take into account the usage of multiple system resources in the energy-aware resource management.

We have already investigated the problem of energy-aware dynamic consolidation of VMs according to the current CPU utilization. However, to allow a better VM placement optimization, the VMs should be reallocated according to the current utilization of multiple system resources, including the CPU, RAM, and network bandwidth as shown in Fig. 9. Disk storage is usually centralized (e.g. NAS) to enable live migration of VMs and, therefore, requires specific energy-efficient management techniques. The problem arises when it comes to providing strict SLAs ensuring no performance degradation, which is required for a Cloud data center. A generic Cloud computing environment (IaaS) is built to serve multiple applications for multiple users, creating mixed workloads and complicating the workload characterization. How to predict performance peaks? How to determine which VMs, when and where should be migrated to prevent the performance

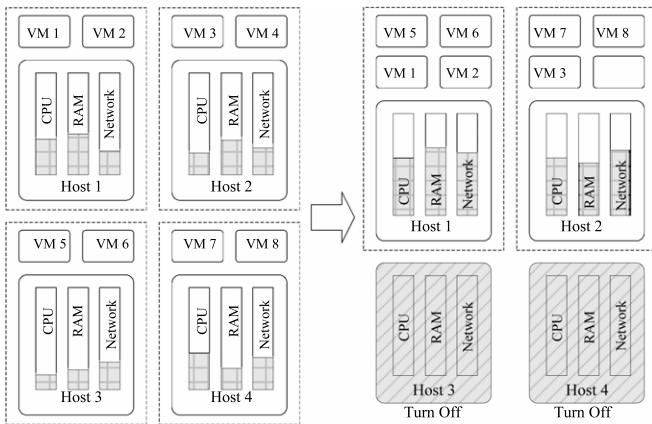


Fig. 9. The optimization over multiple resources.

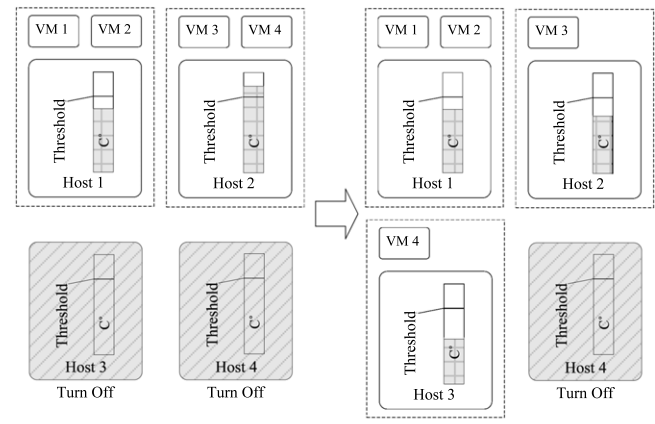


Fig. 11. The temperature optimization.

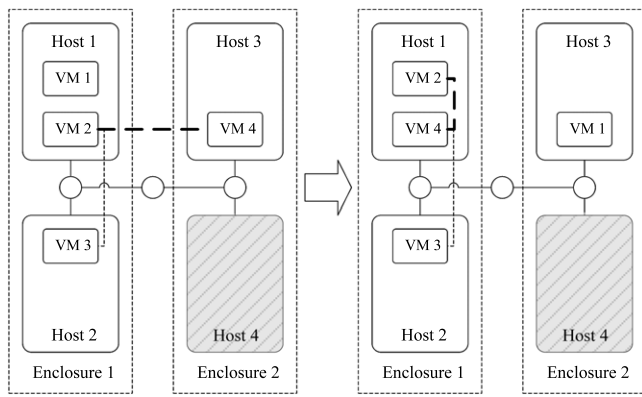


Fig. 10. The network optimization.

degradation considering multiple system resources? How to develop fast and effective algorithms for the VM placement optimization across multiple resources for large-scale systems? These are the challenges that have to be addressed to provide a viable solution for modern Cloud computing environments.

### 6.2. Optimization of virtual network topologies

In virtualized data centers VMs often communicate with each other, establishing virtual network topologies. However, due to VM migrations or a non-optimized allocation, the communicating VMs may end up hosted on logically distant physical nodes providing costly data transfers between each other. If the communicating VMs are allocated to the hosts in different racks or enclosures, the network communication may involve additional network switches and links, which consume significant amount of energy [37]. There have been recent research efforts on the optimization of the allocation of communicating applications to minimize the network data transfer overhead [20–25]. However, these works have not directly addressed the problem of energy consumption by the network infrastructure. Moreover, the proposed approaches do not optimize the placement of VMs at run-time depending on the current network load, which is effective for variable communication patterns and should be applied to virtualized data centers. To eliminate data transfer overheads and minimize energy consumption, it is necessary to monitor the communication between VMs and dynamically adapt their placement depending on the communication between them as shown in Fig. 10.

To provide effective reallocations, we propose the application of the proposed energy models of network devices [37] to the development of VM allocation adaption strategies for Cloud data

centers. As migrations consume additional energy and have a negative impact on the performance, before initiating a migration, the reallocation controller has to ensure that the cost of migration does not exceed the benefit. The energy-aware optimization of the network communication between VMs in Cloud data centers is essential to investigate, as modern service applications are often deployed in multi-VM configurations. The optimal VM placement and its dynamic adaptation can substantially reduce the data transfer overheads, and thus energy consumed by the network infrastructure.

### 6.3. Optimization of thermal states and cooling system operation

A significant part of electrical energy consumed by computing resources is transformed into heat. High temperature leads to a number of problems, such as reduced system reliability and availability, as well as decreased lifetime of devices. In order to keep the system components within their safe operating temperature and prevent failures and crashes, the emitted heat must be dissipated. The cooling problem becomes extremely important for modern blade and 1-unit rack servers, which lead to a high density of computing resources and complicate the heat dissipation. For example, for a 30,000 ft<sup>2</sup> data center with 1000 standard computing racks, each consuming 10 kW, the initial cost of purchasing and installing the infrastructure is \$2–\$5 million; whereas the annual costs for cooling is around \$4–\$8 million [38]. Therefore, apart from hardware improvements, it is essential to optimize the cooling system operation from the software side. There has been work on modeling the thermal topology of a data center that can be applied to achieve a more efficient workload placement [39]. New challenges include how and when to reallocate VMs to minimize the power drawn by the cooling system, while preserving a safe temperature of the resources and minimizing the migration overhead and performance degradation.

We propose the investigation and development of new thermal management algorithms that monitor the thermal state of physical nodes and reallocate the workload (VMs) from overheated nodes as shown in Fig. 11. In this case, the cooling systems of the overheated nodes can be slowed down allowing natural power dissipation. The temperature variations caused by different workloads can be leveraged to swap VMs at an appropriate time to control the temperature and energy consumption. In addition, hardware level power management techniques, such as DVFS, can lower the temperature when it surpasses the thermal threshold. To meet the requirements of Cloud data centers, this problem should be explored for a case when multiple diverse applications with different QoS requirements are executing in the system simultaneously.

#### 6.4. Efficient consolidation of VMs for managing heterogeneous workloads

Cloud infrastructure services provide users with the ability to provision virtual machines and execute any kinds of applications on them. This leads to the fact that different types of applications (e.g., enterprise, scientific, and social network applications) can be allocated to a single physical computing node. However, it is not obvious how these applications can influence each other, as they can be data, network or compute intensive, thus creating variable or static load on the resources. The problem is to determine which kinds of applications can be allocated to a single host to provide the most efficient usage of the resources. Moreover, data center resources may deliver different levels of performance to their clients; hence, performance-aware resource selection plays an important role in Cloud computing.

Current approaches to energy-efficient consolidation of VMs in data centers do not investigate the problem of combining different workload types. These approaches usually focus on one particular workload type, or do not consider different kinds of applications assuming a uniform workload. In contrast to the previous works, we propose an intelligent consolidation of VMs with different workload types. For example, a compute intensive (scientific) application can be effectively combined with a web-application (file server), as the former mostly relies on CPU performance, whereas the latter utilizes disk storage and network bandwidth. It is necessary to investigate which kinds of applications can be effectively combined and which parameters influence the efficiency. Additionally, Cloud applications can present varying workloads. It is therefore essential to carry out a study of Cloud services and their workloads in order to identify common behaviors, patterns, and explore load forecasting approaches that can potentially lead to more efficient resource provisioning and consequently improved energy efficiency. Another research problem is to develop methods for the automatic determination of compatible applications and application of VM migration to adapt the placement when the application behavior changes.

In this context, we propose the investigation of sample applications and correlations between workloads, and attempt to build performance models that can help exploring the trade-offs between QoS and energy savings. This knowledge can be used to develop workload-aware resource allocation algorithms, which can be incorporated into energy-efficient resource management strategies in data centers to achieve more optimal allocation of resources. For resource providers, the optimal allocation of VMs will result in higher utilization of resources and, therefore, reduced operational costs. End-users will benefit from decreased prices for the resource usage. Knowledge of the efficient combination of different types of workloads will advance resource management strategies in energy-aware computing environments, where consolidation of VMs is one of the most productive energy saving techniques.

#### 6.5. A holistic approach to energy-aware resource management

Although all of the optimization techniques discussed above are important, some of them are contradictory. For example, the technique discussed in Section 6.1 is aimed at the consolidation of VMs and increases the amount of physical resources in cases of workload peaks. On the other hand, the technique discussed in Section 6.3 de-consolidates VMs in cases of node overheating incorporating additional constraints. Therefore, the problem of combining different optimization techniques presents a significant research challenge creating a multi-objective optimization problem.

One of the limitations of current practices is that the optimization algorithm is slow due to complex computations or requires

machine-learning that is not suitable for a large-scale data center environment that has to be able to quickly respond to changes in the workload. Usually the optimization controller is centralized that creates a single point of failure and limits the scalability. The proposed holistic approach can be highly resource intensive, as it will incorporate a multi-objective optimization. Therefore, to provide scalability and fault-tolerance, the crucial requirements to the optimization algorithm are decentralization and distributed nature. This implies that reallocation controllers are distributed over multiple physical nodes in a data center and do not have the complete view of the system at any point of time. The research challenges in this case are: How to efficiently propagate the data between controllers? How to effectively combine different optimization techniques? How to ensure that the solution is close to the global optimum? These questions have to be answered in order to develop an approach suitable for real-world Cloud data centers.

### 7. Concluding remarks and future directions

This work advances the Cloud computing field in two ways. First, it plays a significant role in the reduction of data center energy consumption costs, and thus helps to develop a strong and competitive Cloud computing industry. Second, consumers are increasingly becoming conscious about the environment. A recent study shows that data centers represent a large and rapidly growing energy consumption sector of the economy and a significant source of CO<sub>2</sub> emissions [40]. Reducing greenhouse gas emissions is a key energy policy focus of many countries around the world. We have presented and evaluated our energy-aware resource allocation algorithms utilizing the dynamic consolidation of VMs. The experiment results have shown that this approach leads to a substantial reduction of energy consumption in Cloud data centers in comparison to static resource allocation techniques. We are aiming at putting in a strong thrust on open challenges identified in this paper in order enhance the energy-efficient management of Cloud computing environments.

The research work is planned to be followed by the development of a software platform that supports the energy-efficient management and allocation of Cloud data center resources. In order to reduce the cost of software engineering, we will extensively reuse existing Cloud middleware and associated technologies. We will leverage third party Cloud technologies and services offerings including (a) VM technologies, such as open-source Xen and KVM, and commercial products from VMware; (b) Amazon's Elastic Compute Cloud (EC2), Simple Storage Service (S3), and Microsoft's Azure. We will also leverage our own technologies such as Aneka, which is a.NET-based platform for building enterprise Clouds [41]. We will implement a generic resource manager and plug-in software adaptors to allow the interaction with different Cloud management systems such as Aneka and Amazon EC2.

### Acknowledgments

This is a substantially extended version of the keynote paper presented at PDPTA 2010 [6]. We thank Yoganathan Sivaram (Melbourne University), external reviewers and the Guest Editor of this special issue for their suggestions on enhancing the quality of the paper.

### References

- [1] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, A. Warfield, Xen and the art of virtualization, in: Proceedings of the 19th ACM Symposium on Operating Systems Principles, SOSP 2003, Bolton Landing, NY, USA, 2003, p. 177.



- [2] M. Armbrust, A. Fox, R. Griffith, A.D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, M. Zaharia, A view of cloud computing, *Communications of the ACM* 53 (4) (2009) 50–58.
- [3] J. Kaplan, W. Forrest, N. Kindler, *Revolutionizing Data Center Energy Efficiency*, McKinsey & Company, Tech. Rep..
- [4] Ministry of Economy, Trade and Industry, Establishment of the Japan data center council, Press Release.
- [5] The green grid consortium, 2011. URL: <http://www.thegreengrid.org>.
- [6] R. Buyya, A. Beloglazov, J. Abawajy, Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges, in: *Proceedings of the 2010 International Conference on Parallel and Distributed Processing Techniques and Applications, PDPTA 2010*, Las Vegas, USA, 2010.
- [7] E. Pinheiro, R. Bianchini, E.V. Carrera, T. Heath, Load balancing and unbalancing for power and performance in cluster-based systems, in: *Proceedings of the Workshop on Compilers and Operating Systems for Low Power*, 2001, pp. 182–195.
- [8] J.S. Chase, D.C. Anderson, P.N. Thakar, A.M. Vahdat, R.P. Doyle, Managing energy and server resources in hosting centers, in: *Proceedings of the 18th ACM Symposium on Operating Systems Principles*, ACM, New York, NY, USA, 2001, pp. 103–116.
- [9] E. Elnozahy, M. Kistler, R. Rajamony, Energy-efficient server clusters, *Power-Aware Computer Systems* (2003) 179–197.
- [10] R. Nathuji, K. Schwan, Virtualpower: coordinated power management in virtualized enterprise systems, *ACM SIGOPS Operating Systems Review* 41 (6) (2007) 265–278.
- [11] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, X. Zhu, No “power” struggles: coordinated multi-level power management for the data center, *SIGARCH Computer Architecture News* 36 (1) (2008) 48–59.
- [12] D. Kusic, J.O. Kephart, J.E. Hanson, N. Kandasamy, G. Jiang, Power and performance management of virtualized computing environments via lookahead control, *Cluster Computing* 12 (1) (2009) 1–15.
- [13] S. Srikantaiah, A. Kansal, F. Zhao, Energy aware consolidation for cloud computing, *Cluster Computing* 12 (2009) 1–15.
- [14] M. Cardosa, M. Korupolu, A. Singh, Shares and utilities based power consolidation in virtualized server environments, in: *Proceedings of the 11th IFIP/IEEE Integrated Network Management, IM 2009*, Long Island, NY, USA, 2009.
- [15] A. Verma, P. Ahuja, A. Neogi, pMapper: power and migration cost aware application placement in virtualized systems, in: *Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware*, Springer, 2008, pp. 243–264.
- [16] A. Gandhi, M. Harchol-Balter, R. Das, C. Lefurgy, Optimal power allocation in server farms, in: *Proceedings of the 11th International Joint Conference on Measurement and Modeling of Computer Systems*, ACM, New York, NY, USA, 2009, pp. 157–168.
- [17] M. Gupta, S. Singh, Greening of the internet, in: *Proceedings of the ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, SIGCOMM 2003, New York, NY, USA, 2003, pp. 19–26.
- [18] N. Vasic, D. Kostic, Energy-aware traffic engineering, in: *Proceedings of the 1st ACM International Conference on Energy-Efficient Computing and Networking*, e-Energy 2010, Passau, Germany, 2010, pp. 169–178.
- [19] C. Panarello, A. Lombardo, G. Schembra, L. Chiaraviglio, M. Mellia, Energy saving and network performance: a trade-off approach, in: *Proceedings of the 1st ACM International Conference on Energy-Efficient Computing and Networking*, e-Energy 2010, Passau, Germany, 2010, pp. 41–50.
- [20] L. Chiaraviglio, I. Matta, GreenCoop: cooperative green routing with energy-efficient servers, in: *Proceedings of the 1st ACM International Conference on Energy-Efficient Computing and Networking*, e-Energy 2010, Passau, Germany, 2010, pp. 191–194.
- [21] M. Koseoglu, E. Karasan, Joint resource and network scheduling with adaptive offset determination for optical burst switched grids, *Future Generation Computer Systems* 26 (4) (2010) 576–589.
- [22] L. Tomas, A. Caminero, C. Carrión, B. Caminero, Network-aware meta-scheduling in advance with autonomous self-tuning system, *Future Generation Computer Systems* 27 (5) (2010) 486–497.
- [23] E. Dodonov, R. de Mell, A novel approach for distributed application scheduling based on prediction of communication events, *Future Generation Computer Systems* 26 (5) (2010) 740–752.
- [24] L. Gyarmati, T. Trinh, How can architecture help to reduce energy consumption in data center networking? in: *Proceedings of the 1st ACM International Conference on Energy-Efficient Computing and Networking*, e-Energy 2010, Passau, Germany, 2010, pp. 183–186.
- [25] C. Guo, G. Lu, H. Wang, S. Yang, C. Kong, P. Sun, W. Wu, Y. Zhang, Secondnet: a data center network virtualization architecture with bandwidth guarantees, in: *Proceedings of the 6th International Conference on Emerging Networking Experiments and Technologies*, CoNEXT 2010, Philadelphia, USA, 2010.
- [26] L. Rodero-Merino, L. Vaquero, V. Gil, F. Galan, J. Fontan, R. Montero, I. Llorente, From infrastructure delivery to service management in clouds, *Future Generation Computer Systems* 26 (8) (2010) 1226–1240.
- [27] R.N. Calheiros, R. Buyya, C.A.F.D. Rose, A heuristic for mapping virtual machines and links in emulation testbeds, in: *Proceedings of the 38th International Conference on Parallel Processing*, Vienna, Austria, 2009.
- [28] R.K. Sharma, C.E. Bash, C.D. Patel, R.J. Friedrich, J.S. Chase, Balance of power: dynamic thermal management for internet data centers, *IEEE Internet Computing* (2005) 42–49.
- [29] J. Moore, J. Chase, P. Ranganathan, R. Sharma, Making scheduling “cool”: temperature-aware workload placement in data centers, in: *Proceedings of the Annual Conference on USENIX Annual Technical Conference*, Anaheim, CA, USA, 2005.
- [30] R. Buyya, C.S. Yeo, S. Venugopal, J. Broberg, I. Brandic, Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility, *Future Generation Computer Systems* 25 (6) (2009) 599–616.
- [31] M. Yue, A simple proof of the inequality  $FFD(L) < 11/9 OPT(L) + 1$ , for all  $L$  for the FFD bin-packing algorithm, *Acta Mathematicae Applicatae Sinica (English Series)* 7 (4) (1991) 321–331.
- [32] C. Clark, K. Fraser, S. Hand, J.G. Hansen, E. Jul, C. Limpach, I. Pratt, A. Warfield, Live migration of virtual machines, in: *Proceedings of the 2nd Symposium on Networked Systems Design and Implementation*, NSDI 2005, USENIX, Boston, MA, USA, 2005.
- [33] W. Voorsluys, J. Broberg, S. Venugopal, R. Buyya, Cost of virtual machine live migration in clouds: a performance evaluation, in: *Proceedings of the 1st International Conference on Cloud Computing*, CloudCom 2009, Springer, Beijing, China, 2009.
- [34] R.N. Calheiros, R. Ranjan, A. Beloglazov, C.A.F.D. Rose, R. Buyya, CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, *Software: Practice and Experience* 41 (1) (2011) 23–50.
- [35] D. Meisner, B. Gold, T. Wenisch, Powernap: eliminating server idle power, *ACM SIGPLAN Notices* 44 (3) (2009) 205–216.
- [36] L. Minas, B. Ellison, *Energy Efficiency for Information Technology: How to Reduce Power Consumption in Servers and Data Centers*, Intel Press, 2009.
- [37] J. Chabarek, J. Sommers, P. Barford, C. Egan, D.T.S. Wright, Power awareness in network design and routing, in: *Proceedings of the 27th IEEE Conference on Computer Communications*, INFOCOM 2008, Phoenix, AZ, USA, 2008, pp. 457–465.
- [38] C.D. Patel, C.E. Bash, A.H. Beitelmal, Smart cooling of data centers, Google Patents, 2001, US Patent App. 09/970,707.
- [39] J. Moore, J. Chase, P. Ranganathan, Weatherman: automated, online, and predictive thermal mapping and management for data centers, in: *Proceedings of the 3rd International Conference on Autonomic Computing*, ICAC 2006, Dublin, Ireland, 2006.
- [40] P. Johnson, T. Marker, Data centre energy efficiency product profile, Tech. Rep. 2009/05, 2009.
- [41] C. Vecchiola, X. Chu, R. Buyya, Aneka: a software platform for Net-based cloud computing, *Advances in Parallel Computing* 18 (2009) 267–295.



**Anton Beloglazov** is a Ph.D. student at the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the Computer Science and Software Engineering department of the University of Melbourne, Australia. He has completed his Bachelor's and Master's degrees in Informatics and Computer Science at the faculty of Automation and Computer Engineering of Novosibirsk State Technical University, Russian Federation. Under his Ph.D. studies, Anton Beloglazov is actively involved in research on energy- and performance-efficient resource management in virtualized data centers for Cloud computing. He has been contributing to the development of the CloudSim toolkit, a modern open-source framework for modeling and simulation of Cloud computing infrastructures and services. Anton Beloglazov has publications in internationally recognized conferences and journals, such as the 7th International Workshop on Middleware for Grids, Clouds and e-Science (MGC 2009); the 10th IEEE/ACM International Symposium on Cluster, Cloud, and Grid Computing (CCGrid 2010); and *Software: Practice and Experience* (SPE 2009). He is a frequent reviewer for research conferences and journals, such as European Conference on Information Systems (ECIS), IEEE International Workshop on Internet and Distributed Computing Systems (IDCS), International Conference on High Performance Computing (HiPC), International Conference on Software, Telecommunications and Computer Networks (SoftCOM), and Journal of Network and Computer Applications (JNCA). For further information please visit the web-site: <http://beloglazov.info/>.



**Jemal Abawajy** is an associate professor, Deakin University, Australia. Dr. Abawajy is the director of the “Pervasive Computing & Networks” research groups at Deakin University. The research group includes 15 Ph.D. students, several masters and honors students and other staff members. Dr. Abawajy is actively involved in funded research in robust, secure and reliable resource management for pervasive computing (mobile, clusters, enterprise/data grids, web services) and networks (wireless and sensors) and has published more than 200 research articles in refereed international conferences and journals as well as a number of technical reports. Dr. Abawajy has given keynote/invited talks at many conferences. Dr. Abawajy has guest-edited several international journals and served as an associate editor of international conference proceedings. In addition, he is on the editorial board of several international journals. Dr. Abawajy has been a member of the organizing committee for over 100 international conferences serving in various capacity including chair, general co-chair, vice-chair, best paper award chair, publication chair, session chair and program committee. He is also a frequent reviewer

for international research journals (e.g., FGCS, TPDS and JPDC), research grant agencies, and Ph.D. examinations.



**Rajkumar Buyya** is Professor of Computer Science and Software Engineering; and Director of the Cloud Computing and Distributed Systems (CLOUDS) Laboratory at the University of Melbourne, Australia. He is also serving as the founding CEO of Manjrasoft Pty Ltd., a spin-off company of the University, commercializing its innovations in Grid and Cloud Computing. He has authored and published over 300 research papers and four text books. The books on emerging topics that Dr. Buyya edited include, *High Performance Cluster Computing* (Prentice Hall, USA, 1999), *Content Delivery Networks* (Springer, Germany, 2008) and

*Market-Oriented Grid and Utility Computing* (Wiley, USA, 2009). He is one of the highly cited authors in computer science and software engineering worldwide ( $h$ -index = 52,  $g$ -index = 112, 14 500 + citations). Software technologies for Grid and Cloud computing developed under Dr. Buyya's leadership have gained rapid acceptance and are in use at several academic institutions and commercial enterprises in 40 countries around the world. Dr. Buyya has led the establishment and development of key community activities, including serving as foundation Chair of the IEEE Technical Committee on Scalable Computing and four IEEE conferences (CCGrid, Cluster, Grid, and e-Science). He has presented over 200 invited talks on his vision on IT Futures and advanced computing technologies at international conferences and institutions in Asia, Australia, Europe, North America, and South America. These contributions and international research leadership of Dr. Buyya are recognized through the award of "2009 IEEE Medal for Excellence in Scalable Computing" from the IEEE Computer Society, USA. For further information on Dr. Buyya, please visit his cyberhome: [www.buyya.com](http://www.buyya.com).