Chapter 1
Extending Energetic Potentials of Data Centers by Resource Optimization to Improve Carbon Footprint

Alexander Borgerding and Gunnar Schomaker

Abstract The electric power is one of the major operating expenses in data centers. Rising and varying energy costs induce the need of further solutions to use energy efficiently. The first steps to improve efficiency have already been accomplished by applying virtualization technologies. However, a practical approach for data center power control mechanisms is still missing.

In this paper, we address the problem of energy efficiency in data centers. Efficient and scalable power usage for data centers is needed. We present different approaches to improve efficiency and carbon footprint as background information. We propose an in-progress idea to extend the possibilities of power control in data centers and to improve efficiency. Our approach is based on virtualization technologies and live-migration to improve resource utilization by comparing different effects on virtual machine permutation on physical servers. It delivers an efficiency-aware VM placement by assessing different virtual machine permutation. In our approach, the applications are untouched and the technology is non-invasive regarding the applications. This is a crucial requirement in the context of Infrastructure-as-a-Service (IaaS) environments.

Keywords Data center • VM placement • Energy efficiency • Power-aware • Resource management • Server virtualization

1 Introduction

The IP traffic increases year by year worldwide. New Information and Communication Technology (ICT) services are coming up and existing services are migrating to IP technology, for example, VoIP, TV, radio and video streaming. Following

A. Borgerding (✉)
University of Oldenburg, 26111 Oldenburg, Germany
e-mail: alexander.borgerding@uni-oldenburg.de

G. Schomaker
Software Innovation Campus Paderborn, Zukunftsmeile 1, 33102 Paderborn, Germany
e-mail: schomaker@sicp.de

J. Marx Gómez et al. (eds.), *Advances and New Trends in Environmental and Energy Informatics*, Progress in IS, DOI 10.1007/978-3-319-23455-7_1

these trends, the power consumption of ICT obtains a more and more significant value. In the same way, data centers are growing in number and size in order to comply with the increasing demand. As a result, their share of electric power consumption increases too, e.g. it has doubled in the period 2000–2006 [16]. In addition, energy costs rise continuously and the data center operators are faced with customer questions about sustainability and carbon footprint while economical operation is an all-over goal. The electric power consumption has become one of the major expenses in data centers.

A high performance server in idle-state consumes up to 60 % of its peak power [11]. To reduce the quantity of servers in idle-state, virtualization technologies are used. Virtualization technologies allow several virtual machines (VMs) to be operated on one physical server or machine (PM). In this way the number of servers in idle-state can be reduced to save energy [6]. However, the rising energy costs lead to a rising cost pressure and further solutions are needed as they will be proposed in the following.

This paper extends our contribution to EnviroInfo 2014 – 28th International Conference on Informatics for Environmental Protection [3] and is organized as follows: Sect. 2 motivates and defines the problem of energy efficiency and integrating renewable energy in data centers. Section 3 gives background on approaches relevant to energy efficiency, virtualization technology and improving the carbon footprint. In Sect. 4, we present the resource-efficient and energy-adaptive approach. The paper is concluded by comments on our progressing work in Sect. 5.

2 Problem Definition

The share of volatile renewable power sources is increasing. This leads to volatile energy availability and lastly to varying energy price models. To deal with the variable availability, we need an approach that ensures controllable power consumption beyond general energy efficiency. Thus, we need to improve the efficiency of the data center using an intelligent and efficient VM placement in order to adapt to volatile energy availability and improve carbon footprint while keeping the overall goal to use the invested energy as efficient as possible.

The increasing amount of IT services combined with steadily raising energy costs place great demands on data centers. These conditions induce the need to operate a maximum number of IT services with minimal employment of resources, since the aim is an economical service operation. Therefore, the effectiveness of the invested power should be at a maximum level. In this paper, we focus on the server's power consumption and define the efficiency of a server as the work done per energy unit [5].

In the related work part of this paper, we analyze different kinds of approaches in the context of energy consumption, energy efficiency and integrating renewable power. In this research approach, we want to explore which further options exist to use energy efficient and how we can take effect on the data center's power consumption and, finally, to adapt it to available volatile renewable energy.

To take advantage of current developments, power consumption should be increasable in times of low energy prices and reducible otherwise while we stick to a high efficiency level in both cases. In Service Level Agreements (SLAs) for instance, a specific application throughput within a time frame is defined. Due to these agreements, we can use periods of low energy prices to produce the throughput far before the time frame exceeds. In periods of high energy prices, a scheduled decrease of the previously built buffer can be used to save energy costs.

Some approaches [4, 10, 15] use geographically-distributed data centers to schedule the workload across data centers with high renewable energy availability. The methodology is only suitable in big, geographically-spread scenarios and the overall power consumption is not affected. Hence, we do not pursue these approaches. In general, many approaches are based on strategies with focus on CPU utilization because CPU utilization correlates with the server's power consumption directly [5]. The utilization of other server components does not have such an effect on the server's power consumption. However, the application's performance depends not only on CPU usage, but all required resources are needed for optimal application performance. Hence, the performance relies on other components too and we also want to focus on these other components such as Network Interface Card (NIC), Random Access Memory (RAM) and Hard Disk Drive (HDD) to improve the efficiency, especially if their utilization does not have an adverse effect on the server's power consumption. Our assumption is that the optimized utilization of these resources is not increasing the power consumption, but it can be used to improve the efficiency and application performance.

There are different types of applications; some applications work stand-alone while others rely on several components running on different VMs. Components of the latter communicate via network and the network utilization takes effect on such distributed applications. In our approach, we want to include these communication topology topics. However, the applications' requirements are changing during operation, sometimes in large scale and in short intervals. Therefore, we need an online algorithm that acts at runtime to respond to changing values. We need to keep obstacles at a low level by acting agnostic to the applications. The capable approach should be applicable without the need to change the operating applications. This is a crucial requirement in the context of Infrastructure-as-a-Service (IaaS) environments.

Being agnostic to applications means to influence their performance without they become aware of our methodology. For example, if an application intends to write a file on the hard disk, it has to wait until it gets access to the hard disk. This is a usual situation an application can handle. In the wait state, the application cannot distinguish whether the wait was caused by another application writing on the hard disk or by our methodology.

The problem of determining an efficient VM placement can be formulated as an extended bin-packing problem, where VMs (objects) must be allocated to the PMs (bins). In the bin-packing problem, objects of different volumes must be fitted into a finite number of bins, each of the same volume, in a way that minimizes the number of bins used. The bin-packing problem has an NP-hard complexity. Compared to the VM allocating problem, we have a multidimensional bin packing problem. Instead of the object size, we have to deal with several resource requirements of VMs.

In a data center with k PMs and n VMs operated on the PMs, the number of configuration possibilities is described by partioning a set of n elements into k partitions while the k sets are disjoint and nonempty. This is described by the Stirling numbers of the second kind:

$$S_{n,k} = \frac{1}{k!} \sum_{j=0}^{k} (-1)^{k-j} \binom{k}{j} j^n$$

In case of a data center with 10 VMs and 3 PMs, we have $S_{10,3} = 9330$ different and possible VM allocations to the PMs that are named as configurations in this paper.

Hence, a global bin-packing solver will not be able to deliver a VM placement for a fast acting online approach.

The formal description of the VM placing problem relating to the bin-packing problem is as follows: A set of virtual machines $\mathrm{V} = \{\mathrm{VM}_1, \ldots, \mathrm{VM}_n\}$ and a set of physical machines $\mathrm{P} = \{\mathrm{PM}_1, \ldots, \mathrm{PM}_k\}$ is given. The VMs are represented by their resource demand vectors d_i. The PMs are represented by their resource capacity vectors c_s. The resource capacity vector of a PM describes the available resources that can be requested by VMs. The goal is to find a configuration so that for all PMs in P:

$$\sum_{i=1}^{j} d_i \le c_s$$

while j is the total number of VMs on the PM.

To measure the quality of an allocated configuration C, the efficiency E defined by:

$$E(C) = \frac{work\, done}{unit\, energy}$$

is a suitable metric [5]. The aggregated idle times of the PMs may also indicate the quality of the configuration.

To the best of our knowledge, this is the first approach that researches on agnostic methodologies, without scheduling components, to control the data centers power consumption with the aim of efficiency and the possibility to increase and decrease the power consumption as well.

3 Related Work

Power consumption and energy efficiency in data centers is a topic, on which a lot of work has already been done. In this section, we give an overview of different approaches.

The usage of low-power components seems to offer solutions for lower energy consumption. Meisner et al. [12] handled the question whether low power consumption correlates with energy efficiency in the data center context. They discovered that the usage of low power components is not the solution. They compared low power servers with high power servers and defined the energy efficiency of a system as the work done per energy unit. They achieved better efficiency with the high power servers and found that modern servers are only maximally efficient at 100 % utilization.

Another potential for improvement is to let IT requirements follow energy availability. There are some approaches [4, 10, 14] that use local energy conditions. They migrate the server workload to data center destinations with available renewable power. These ideas are finally only suitable for distributed and widespread data centers. Data center locations at close quarters typically have the same or not significantly different energy conditions. In the latter scenario, the consumption of renewable energy can be increased, but the efficient power usage is not taken into consideration.

A different idea is mentioned by Krioukov et al. [9]. In this work, a scheduler has access to a task list, where the task with the earliest deadline is at the top. This is an earliest deadline first (EDF) schedule. If renewable energy is available, the EDF scheduler starts tasks from the top of the task list to use the renewable energy. If less energy is available, tasks will be terminated. In such approaches, we have to deal with application-specific topics. To build a graded list of tasks to schedule, we determine the duration a task needs to be processed and we need a deadline for each task to be processed. Terminated tasks lead to application-specific issues that need to be resolved afterwards.

The approach of Hoyer [8] bases on prediction models to calculate the needed server capacity in advance to reduce unused server capacity. Optimistic, pessimistic and dynamic resource strategies were presented. This approach offers methodologies to improve efficiency, but controlling the data centers power consumption is not focused.

Tang et al. [17] propose a thermal-aware task scheduling. The ambition is to minimize cooling requirements and to improve the data center efficiency in this way. They set up a central database with server information, especially server heat information. An EDF scheduler is placing tasks with the earliest deadline on the coldest server. Thus, they avoid hot spots and cooling requirement can be decreased to improve efficiency. The usage of a graded task list comes with the same disadvantages as described before. To avoid dealing with application-specific topics, the virtual machine is a useful container to place IT loads instead of explicit application tasks. In many approaches, for example Corradi et al. [6], power consumption is reduced by concentrating VMs on a fewer number of servers and powering down unused ones to save energy. Chen et al. [5] describe the power consumption of a server as the sum of its static power consumption and its dynamic power consumption. The static power consumption is the consumption of the server in power-on state without workload. This amount of power can be saved with this approach. The dynamic part of server's power consumption correlates with its CPU utilization, as described by Pelley et al. [13]. Thus, most methodologies are only focused on CPU utilization.

Dalvanadi et al. [7] and Vu et al. [18] pointed out that network communication can also influence the overall performance of an IT service and network-aware VM placement is also an important and challenging issue. Hence, they embrace network traffic to minimize power consumption.

As described, many approaches [1, 15, 19] use virtualization technologies to concentrate VMs on a small number of PMs. While migrating VMs onto a PM, the size of the RAM is a limiting factor. If the RAM-size of the PM is exhausted, further VMs cannot be migrated onto this PM. This can be an adverse effect, especially if resources such as CPUs are still underutilized or completely idling. The memory sharing technology offers the possibility to condense redundant memory pages on a PM to one page. Unneeded physical memory can be freed to improve the VMs memory footprint. The VMs run on top of a hypervisor, which is responsible for allocating the physical resources to individual VMs. The hypervisor identifies identical memory pages on the different VMs on a PM and shares them among the VMs with pointers. This frees up memory for new pages. If a VM's information on that shared page changes, the hypervisor writes the memory to a new page and re-addresses a pointer. The capacity of the PM can be increased to concentrate further VMs on the PM and to achieve higher server utilization. Wood et al. [19] present a memory sharing-aware placement approach for virtual machines that includes a memory fingerprinting system to determine the sharing potential among a set of VMs. In addition, it makes use of live migration to optimize the VM placement.

In summary, the state of the art approaches deliver several solutions in the context of energy efficiency, but an efficiency-aware approach with combined data center power control mechanisms is still missing.

4 Resource-Efficient and Energy-Adaptive VM Placement Approach

In this section the in-progress idea for resource-efficient and energy-adaptive VM placement in data centers is proposed. To optimize the server utilization, many data center operators already use server virtualization technologies and operate several virtual machines on one physical server. This technology is the base for our further optimizations. In our approach, we are at the point that the first steps of optimizations have already been done. Hence, we are running a set of VMs concentrated on a small number of potential servers. Unused servers are already switched off. As further input, we get a target power consumption value.

It is generally accepted that applications operate ideally if they have access to all required server resources. With the aim of improving the data center's efficiency, resource-competing VMs should not be operated on the same physical server together. Our approach is to create a VM allocation that concentrates VMs with suitable resource requirements on the same physical server for ideal application performance and efficiency. In this constellation, each application has access to the required server resources and operates ideally. Finally, the overall server resources are more utilized than before and the efficiency rises. Beside the increased efficiency, this situation also leads to a higher power consumption and application performance. This scenario is suitable for times of high energy availability. Following the idea of green energy usage, this technology is also capable of reducing the data center's power consumption in situations of less green power availability. Therefore, the methodology can be used to explicitly reduce resource utilization by combining resource-competing applications, leading to lower power consumption but also to a potentially reduced application performance.

In data centers, applications induce specific power consumptions by their evoked server load. This required amount of power is so far understood as a fixed and restricted value. Our concept is to let this amount of power become a controllable value by applying a corresponding VM allocation.

The power consumption PC_{dc} of a data center breaks down as follows:

$$PC_{dc} = PC_{Support} + PC_{Servers}$$

The total power consumption is the sum of the power consumption of all data center components. Beside the power consumption of all PMs $PC_{Servers}$, we have the power consumption of the support infrastructure $PC_{Support}$ i.e. network components, cooling components, UPS, lights, etc.

Chen et al. [5] describe the power consumption of a server as the sum of its static (idle, without workload) power consumption $PC_{Servers\,idle}$ and its dynamic power consumption $PC_{servers\,dyn.}$:

$$PC_{Servers} = PC_{Servers\,idle} + PC_{servers\,dyn.}$$

$PC_{servers\,dyn.}$ is the amount of power we directly take influence on. It reflects the amount of power consumption deviance between 100 % server utilization and idle mode. Idle servers still consume 60 % of their peak power draw [11].

Hence, a sustainable part (up to 40 %) of the server's power consumption is controllable; it can be increased in times of high energy availability and decreased otherwise. Our approach is based on virtualization technology and the possibility to live-migrate VMs. The methodology is agnostic to the operating applications. This is an advantage compared to other task scheduling-based algorithms, since these have to deal with task execution times and other application-specific topics. In our approach, the applications are untouched and the technology is non-invasive regarding the applications; it only takes effect on the availability of server resources. The variable availability of server resources is a usual setting that applications are confronted with.

As described in the related work part of this paper, the PM's RAM can be a limiting factor while migrating further VMs to the PM. In addition, we make use of the technology to share RAM across the VMs to increase the number of VMs operated on a PM.

The following diagrams illustrate the practice, how the methodology's strategy migrates VMs between physical servers.

In Fig. 1.1, the initial, non-optimized situation is displayed showing a set of VMs operated on three physical servers. The resource utilization is highlighted (lighter colors meaning low, darker colors high utilizations). On PM2, for example the performance is affected by high network utilization.

Our methodology achieves an equilibrium allocation regarding the resource utilization, as shown in Fig. 1.2. VMs to migrate are chosen depending on their RAM size and their fraction of scarce resource utilization. The subsequent VM permutation leads to an average utilization of all involved resources. Hence, the approach increases efficiency and power consumption by resource usage optimization.

The configuration is suitable for times of high energy availability and low energy prices. In periods of less available renewable energy or high energy prices, we need to reduce the power consumption while keeping a high efficiency level.

The situation, as shown in Fig. 1.3, is the result with reduced power consumption objectives. The CPU utilization is reduced to likewise reduce the power consumption as well while the utilization of other resources is balanced. The result is the most effective constellation at reduced power conditions. The Dynamic Voltage

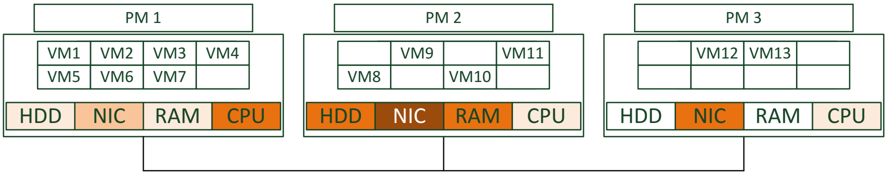


Fig. 1.1 Schematic VM on physical server diagram: initial situation

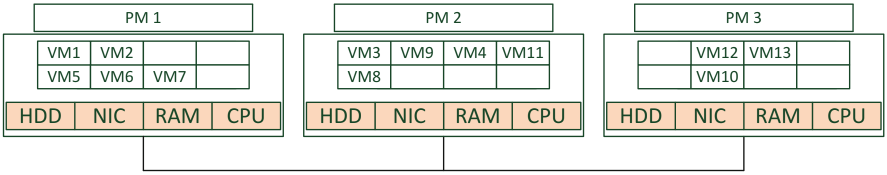


Fig. 1.2 Schematic VM on physical server diagram: optimized situation

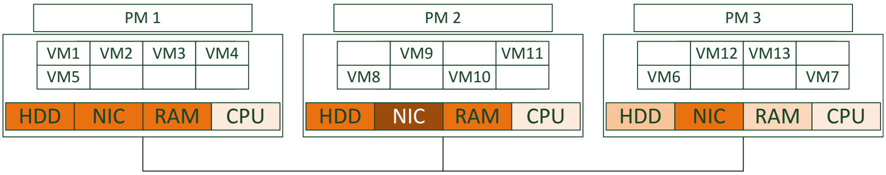


Fig. 1.3 Schematic VM on physical server diagram: aim of reduced power consumption

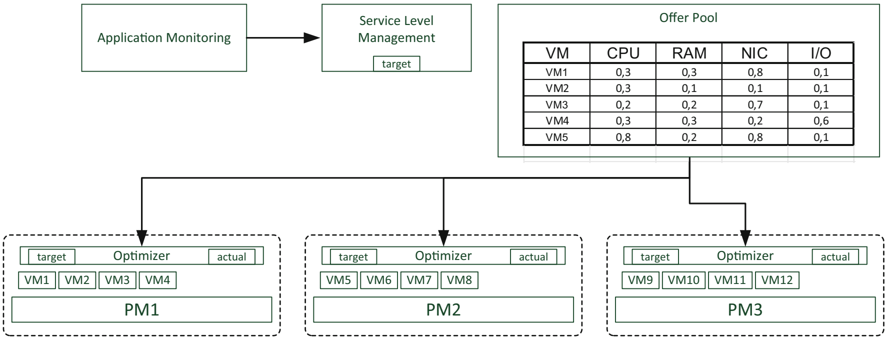
and Frequency Scaling (DVFS) technique is used to adapt the power consumption to the actual CPU utilization. DVFS allows a dynamical adaption of CPU voltage and CPU frequency according to the current resource demand.

4.1 System Model

As described, we have an NP-hard complexity if we stick to methodologies that involve all possible configurations to find the best suitable configuration for our actual requirements. To reduce the complexity and the long computation time, we change from an all-embracing global solution to local solving strategies. On the one hand, this is required for online acting approaches and on the other hand, we assume that we will not get significantly better overall solutions if we include all VMs to find a suitable configuration.

In Fig. 1.4, a component model of the entire system is shown. We have an application-monitoring component that delivers information about the applications and servers to the service level management (SLM). The SLM component contains all service level agreements (SLAs) and calculates new power target values for the data center to observe the SLAs. These values are propagated to all optimizers, working on every physical server. The optimizer compares the new incoming target values with its own actual value. If the difference is in range of a predefined hysteresis, the optimizer does not take any action. Otherwise it starts optimization. If the target is not in the predefined range and the actual value is lower than the target, the optimizer resolves resource competing constellations and hosts additional VMs from the offer pool. In the offer pool, all distributed optimizers can announce VMs, for example, if they do not fit to their actual placement strategy. The VMs in the offer pool are represented with their resource requirements that are the base for later VM placement swaps. If the actual value is higher than the target, the optimizer arranges a resource competing allocation to reduce the power consumption.

The energy availability, energy prices and service level values are independent and global values to aggregate to a target power consumption value. This is a task for the central service level management (SLM) component of our system. Here we do not have any local issues to attend, so we can calculate these values globally. As an additional effect of the globally defined target power consumption value, we



VM	CPU	RAM	NIC	I/O
VM1	0,3	0,3	0,8	0,1
VM2	0,3	0,1	0,1	0,1
VM3	0,2	0,2	0,7	0,1
VM4	0,3	0,3	0,2	0,6
VM5	0,8	0,2	0,8	0,1

Fig. 1.4 Schematic system model

have evenly distributed server utilization. This reduces the occurrence of hot spots, similar to the approach mentioned by Tang et al. [17].

We use a local optimizer component working on a single PM that focuses on a solution for its own PM. This component has to find a solution for just one PM and the set of possible VMs is reduced to the actual operated ones and to a subset of those in the offer pool. As input, the optimizer receives a defined target power consumption, which has to be reached with best possible efficiency.

4.2 Algorithm

CPU utilization is the most effective value regarding power consumption as mentioned before. In other words, the overall CPU utilization is the value to increase or decrease to take effect on the data center's power consumption. Our approach uses competing resource allocations to slow down applications and in series the CPU utilization. Consolidating VMs on a PM that utilize the same resources except the CPU can accomplish this. Consequently, the CPU utilization and power consumption decreases. This practice affects the application's performance and we need a feedback that is sent from the application-monitoring component to the SLM component to ensure the SLAs. With the information about the SLAs and actual application performance, the SLM component is able to calculate power consumption target values that achieve the economic data center objectives.

The target power consumption is broadcasted to all PMs. The PM has got an optimizer component that receives the target and compares it with its actual value. If the target is similar to the actual value, the optimizer does not interfere. Otherwise it starts optimizing. While doing this, the focus is kept on balanced resource utilization and efficiency. Hence, the overall CPU utilization is reduced or increased but all other resources are used as efficiently as possible. Balanced resource utilization is always the goal except for CPU utilization and resources that are used to build the competing resource situation. Merely the attainable CPU utilization is a variable and implicit value that corresponds to the power consumption target value.

Every PM's optimizer strives to reach the target value by optimizing its own situation. We have an offer pool of VMs, which can be accessed by every PM's optimizer. The optimizer is able to read the offered VMs from other PMs or even to offer VMs. If the target value is greater than the actual value, the optimizer removes suitable VMs from the pool to host until the target value is reached. If the target is lower than the actual value, the optimizer offers VMs to the pool to reduce the own value. Furthermore, additional VMs can be hosted from the pool to create competing resource situations to reduce the CPU utilization and to reach the target value. Developing a reduced power consumption VM allocation can be done in three ways:

(i) Migrate VMs to other PMs. This reduces the CPU utilization and the power consumption by DVFS technology.
(ii) The optimizer arranges a resource competing allocation, which reduces the CPU utilization and-as a result-decreases the power consumption by DVFS technology.
(iii) The optimizer arranges CPU overprovisioning. CPU utilization is already at 100 % and further VMs will be hosted. The additional VMs do not increase the PM's power consumption but reduce the power consumption of the PM they came from. Hence, the overall power consumption is decreasing.

The strategy to reduce the power consumption starts with (i) and is cascading down to the methodology of (iii). At first, the target is strived with (i), if this is not leading to the required results, we go on with (ii) and lastly with (iii). Using the methodology of (i) means, we have no further risks of SLA-violation because the application's performance is not influenced. In (ii) und (iii) we potentially slow down the applications, probably increasing the risk of SLA violations. Hence, the methodology always starts in step (i).

The formal description of the efficiency and power consumption problem is a follows: A set of virtual machines $\mathrm{V} = \{\mathrm{VM}_1, \ldots, \mathrm{VM}_n\}$ and a set of physical machines $\mathrm{P} = \{\mathrm{PM}_1, \ldots, \mathrm{PM}_k\}$ is given. The VMs are represented by their resource demand vectors d_i. The PMs are represented by their resource capacity vectors c_i. The goal is to find a configuration C so that for all PMs in P:

$$\sum_{i=1}^{j} d_i \leq c_s + x_s$$

where the vector x_s is an offset to control under- and overprovisioning of the server resources on PM_s while j is the total number of VMs on the PM_s. We use x_s to control the resource utilization on the PMs to induce the intended server utilization and thereby their power consumption.

To measure the quality of an allocated configuration C, we have now two different metrics. On the one hand, we have the efficiency E:

$$E(C) = \frac{work\,done}{unit\,energy}$$

On the other hand, we have the difference Δ between the PMs power consumption PC_{server} and the target power consumption PC_{target}:

$$\Delta_{(target,\,C)} = \left|PC_{target} - PC_{server}(C)\right|$$

The Δ represents the deviance (positive) from the target power consumption. In case of lower target power consumptions, a lasting deviance is the indicator to go on with the next step (ii) or (iii).

The process of reaching a suitable VM placement and the behaviour of the locally executed optimizer is demonstrated by the following pseudo code:

```
Inputs: $t$ target power consumption for local PM, $p$ actual PM's power
    consumption, resource utilization
Output: VM placement for local PM that evokes target power consumption
```

```
1. receive new  target $t$ given by SLM component
2. if $t > p$ and the PM's CPU utilization is 100 %, offer VMs to other PMs via
   offer pool
3. if $t > p$ and the PM's CPU utilization is lower than 100 % and all other
   resources are underutilized, the PM invites VMs to shelter from other PMs
   with high CPU utilization
4. if $t > p$ and the PM's CPU utilization is lower than 100 % and other
   resources are strong utilized, offer VMs to other PMs to solve the competing
   resource situation
5. if $t < p$ and the PM's CPU utilization is lower than 100 % and other
   resources are strong utilized, invite VMs to shelter from other PMs with
   high CPU utilization
6. if $t < p$ and the PM's CPU utilization is 100 %, invite VMs to shelter from
   other PMs to create resource competing situation
7. if $t = p$ do nothing
```

In addition to the event of changing power consumption targets, we have further events to deal with. During the operation a host can become over- or underloaded. A PM's overload might lead to SLA violations and an underload means that the efficiency is not at optimum level. Depending on the actual power consumption targets, we have to act in different ways. If the power consumption target is higher than the PM's actual value and an underload is detected, the condition of:

$$\sum_{i=1}^{j} d_i \leq a_s + x_s$$

implies to resolve resource competing allocations or to host further VMs.

If an overload is detected, the condition might be fulfilled. If x_s is used to reduce the power consumption and to strive the power consumption target, we have a pseudo-overloaded PM. The SLA component decides weather to risk SLA violations and stick to the values of x_s, or to consume expensive energy, for example and to adapt x_s for using additional CPU resources.

Another point is to find candidates for VM migrations. There are some key properties, which suitable candidates should fulfill. In general, the target PM must provide the required RAM space to avoid page swapping. The migration costs for a new VM allocation are a substantial topic we have to look at. Instead of choosing randomized candidates, we sort the set of candidates by their RAM size at first to build an ordered list. The RAM size indicates the duration of the migration because copying the RAM pages to the target PM is a major part in the migration process. In a second step, the resource requirements will be taken into account, reducing the set of potential candidates and leading to a graded list of suitable VMs.

Finally, the local optimizer initializes the migration of the best fitting VM.

4.3 Future Work and Experiments

In experiments, we evaluated scenarios with different VMs to validate our approach, based on affecting server's power consumption and application performance by applying various VM allocations. The test-VMs are running benchmarks, simulating applications that rely on different server resources (VM1 is running RAM benchmarks, VM2 is running file benchmarks and VM3 is running CPU benchmarks). The results show that VM placement strategies can improve the performance up to 34 % and increases the power consumption up to 16 %. Vice versa, VM permutations can decrease the power consumption up to 16 % (up to 50 % if idle PMs are switched off) and even decrease the application performance up to 34 %.

The test results are shown in Table 1.1. The first columns illustrate the VMs allocated to the physical servers PM1 and PM2, followed by the corresponding power consumption of the PMs. The performance columns are containing information about the achieved performance per VM. We normalized the performances with the achieved performance operating the specific VM on the PM separately. VM2 reaching lightly more than 100 % performance is caused by measurement

Table 1.1 Test results

Case	PM1	PM2	PM1 [W]	PM2 [W]	Perf. VM1 [%]	Perf. VM2 [%]	Perf. VM3 [%]	Ø Perf. [%]	Σ PC [W]	Ø Perf./Σ PC
1	VM1, VM2, VM3	–	54.1	38.0	43.62	100.00	44.80	62.81	92.1	0.6840
2	VM1, VM2	VM3	55.9	53.4	89.22	97.56	100.00	95.59	109.3	0.8692
3	VM1, VM3	VM2	55.0	39.9	49.51	100.00	50.32	66.61	94.9	0.7060
4	VM2, VM3	VM1	54.6	55.0	100.00	101.88	86.39	96.09	109.6	0.8759

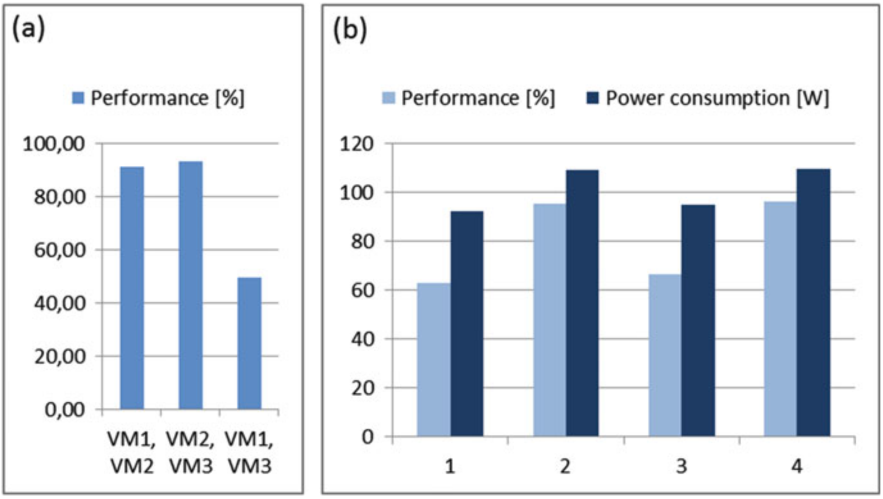


Fig. 1.5 Performance dependencies (**a**); Power consumption and performance per test-case (**b**)

uncertainty. The last columns summarize the average performance and power consumption (PC) results.

The last column shows the performance per power consumption as degree of efficiency. The fourth test-case attains the best efficiency. Figure 1.5b gives an overview of the achieved performance and power consumption. In Fig. 1.5a, the performance depending on the combined VMs is displayed. We got lightly different performance combining VM1, VM2 and VM2, VM3, but operating VM1 and VM3 together causes massive performance losses. The reached performance indicates variations of more than 40 % and shows high potential of our approach.

Our primary goal is to increase the data center's power efficiency. The essential research work is to analyze the different reachable effects by combining further methodologies, for example RAM-sharing, network aware approaches and integrating further resources (such as RAM, NIC and HDD) into the approach as described before.

In our ongoing work, we evaluate performance metrics to estimate the quality of VM allocations. The aggregated idle times of a PM just indicate the efficiency of the PM usage, as described. In order to be able to make sound statements on the quality of our VM allocations, we need further metrics, which allow us to involve the power consumption targets and to get a weighted estimation of the quality on efficiency and deviance to target power consumption.

To ensure comparability with other dynamic VM placing approaches, we implement our methodology into a standard framework such as OpenStack. It is planed to evaluate the methodology in an OpenStack testbed by applying it to different kinds of applications and different workloads. To generate reliable and comparable benchmark results, we execute benchmarks with standard applications and real world workload traces as mentioned in [2].

In further experiments, we will point out the major effects to reduce the complexity. We point out the insignificance of involving all VMs into the solution process according to our assumption that there is no significant potential lost by involving just a reduced number of VMs. Therefore, we compare the resulted configuration of our approach with an offline calculated configuration embracing all VMs.

Finally, we have to prove the additional efficiency and demonstrate the benefits of controllable power consumption. In this context, we evaluate available open source tools and testbeds to treat both aspects and show the advantage of our methodology in context of volatile power availability.

5 Conclusion

We pointed out the raising data center demand, the increasing energy costs and the requirement to handle volatile energy availability respectively. In Sect. 3, we presented different approaches related to energy efficiency, power consumption and usage of renewable power in data centers. We defined the problem of energy efficiency and proposed a resource-optimization approach that improves overall energy efficiency and also allows controlling actual data center power consumption without application-invasive measures. Our approach is an instrument to increase efficiency and to adapt to renewable power availability, both having a positive effect on the carbon footprint.

References

1. Beloglazov, A. and Buyya, R.: Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in Cloud data centers. Concurrency and Computation: Practice and Experience, 2012.
2. Beloglazov, A. and Buyya, R.: OpenStack Neat: a framework for dynamic and energy-efficient consolidation of virtual machines in OpenStack clouds, Concurrency and Computation: Practice and Experience, 2014.
3. Borgerding, A. and Schomaker, G.: Extending Energetic Potentials of Data Centers by Resource Optimization to Improve Carbon Footprint. EnviroInfo 2014 - 28th International Conference on Informatics for Environmental Protection, pages 661–668, 2014.
4. Chen, C., et al.: Green-aware workload scheduling in geographically distributed data centers. 4th IEEE International Conference on Cloud Computing Technology and Science Proceedings, pages 82–89, 2012.
5. Chen, H., Caramanis, M.C. and Coskun, A.K.: The data center as a grid load stabilizer, Design Automation Conference (ASP-DAC), 2014 19th Asia and South Pacific, pages. 105-112, 20–23, Jan. 2014.
6. Corradi, A., Fanelli, M. and Foschini, L.: VM consolidation: A real case based on OpenStack Cloud, Future Generation Comp. Syst., 32, pages 118–127, 2014.
7. Dalvandi, A., et al.: Time-Aware VM-Placement and Routing with Bandwidth Guarantees in Green Cloud Data Centers. In Proceedings of the 2013 I.E. International Conference on Cloud

Computing Technology and Science - Volume 01 (CLOUDCOM '13), Vol. 1. IEEE Computer Society, Washington, DC, USA, pages 212–217, 2013.

8. Hoyer, M.: Resource management in virtualized data centers regarding performance and energy aspects, Doctoral dissertation, Oldenburg, Univ., Diss., 2011.
9. Krioukov, A., et al.: Integrating renewable energy using data analytics systems: Challenges and opportunities. IEEE Data Eng. Bull., 34(1), pages 3–11, 2011.
10. Liu, Z., et al.: Greening geographical load balancing. In Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems, SIGMETRICS '11, pages 233–244, New York, NY, USA, 2011.
11. Meisner, D., et al.: PowerNap: eliminating server idle power. In Proceedings of the 14th international conference on Architectural support for programming languages and operating systems (ASPLOS XIV). ACM, New York, NY, USA, pages 205–216, 2009.
12. Meisner, D., et al.: Does low-power design imply energy efficiency for data centers? In Naehyuck Chang, Hiroshi Nakamura, Koji Inoue, Kenichi Osada and Massimo Poncino, editors, ISLPED, pages 109–114. IEEE/ACM, 2011.
13. Pelley, S., et al.: Understanding and abstracting total data center power. Proc. of the 2009 Workshop on Energy Efficient Design (WEED), Jun. 2009.
14. Qureshi, A., et al.: Cutting the electric bill for internet-scale systems. SIGCOMM Comput. Commun. Rev., 39(4), pages 123–134, August 2009.
15. Schröder, K., Schlitt, D., Hoyer, M. and Nebel, W.: Power and cost aware distributed load management. In Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking (e-Energy '10). ACM, New York, NY, USA, pages 123–126, 2010.
16. Smith, J. W. and Sommerville, I.: Workload Classification & Software Energy Measurement for Efficient Scheduling on Private Cloud Platforms, CoRR abs/1105.2584, 2011.
17. Tang, Q., et al.: Thermal-Aware Task Scheduling for Data centers through Minimizing Heat Recirculation, The IMPACT Laboratory School of Computing and Informatics Arizona State University Tempe, AZ 85287, 2008.
18. Vu, H. and Hwang, S.: A Traffic and Power-aware Algorithm for Virtual Machine Placement in Cloud Data Center. International Journal of Grid & Distributed Computing 7.1, 2014.
19. Wood, T., Tarasuk-Levin, G., Shenoy, P., Desnoyers, P., Cecchet, E. and Corner, M. D.: Memory buddies: exploiting page sharing for smart colocation in virtualized data centers. In Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments (VEE '09). ACM, New York, NY, USA, pages 31–40, 2009.