

Robust Optimization for Energy-Efficient Virtual Machine Consolidation in Modern Datacenters

Robayet Nasim · Enrica Zola · Andreas J. Kassler

Received: date / Accepted: date

Abstract Energy efficient Virtual Machine (VM) consolidation in modern data centers is typically optimized using methods such as Mixed Integer Programming, which typically require precise input to the model. Unfortunately, many parameters are uncertain or very difficult to predict precisely in the real world. As a consequence, a once calculated solution may be highly infeasible in practice. In this paper, we use methods from robust optimization theory in order to quantify the impact of uncertainty in modern data centers. We study the impact of different parameter uncertainties on the energy efficiency and overbooking ratios such as e.g. VM resource demands, migration related overhead or the power consumption model of the servers used. We also show that setting aside additional resource to cope with uncertainty of workload influences the overbooking ration of the servers and the energy consumption. We show that, by using our model, Cloud operators can calculate a more robust migration schedule leading to higher total energy consumption. A more risky operator may well choose a more opportunistic schedule leading to lower energy consumption but also higher risk of SLA violation.

Keywords Virtual Machine Consolidation · Energy Efficiency · Optimization Model · Robust Optimization · Cloud Computing · Green Datacenter.

Robayet Nasim, Andreas J. Kassler
Department of Mathematics and Computer Science
Karlstad University, Sweden
E-mail: robayet.nasim@kau.se, andreas.kassler@kau.se

Enrica Zola
Department of Telematics Engineering
Universitat Politècnica de Catalunya (UPC), Spain
E-mail: enrica@entel.upc.edu

1 Introduction

Energy efficiency is an increasingly important concern for datacenter operators due to both cost and environmental issues. Clearly, reducing the energy consumption of a datacenter is an economic incentive for datacenter operators, and would also lead towards a more sustainable environment as it helps to reduce the global CO₂ footprint. As estimated by Amazon [8], the monetary costs of a datacenter related to energy consumption is approximately 42% which include both direct power consumption (approximately 19%) and the cooling infrastructure (approximately 23%). While operators of large datacenters such as Google or Facebook are constantly reducing their energy consumption by e.g. replacing old hardware by more energy efficient one or introducing more efficient cooling systems, still the total energy consumption is increasing due to the massive expansion of capacity in order to support increasing user demand. For example, the energy consumed by all datacenters of Facebook in 2012 was 678m kWh, an increase of almost 30% from the year before¹. But replacing datacenter hardware is difficult to do for small or medium datacenter operators due to the extra CAPEX involved. As a consequence, a more energy efficient operation of a datacenter is imperative.

Modern virtualization technology offers many benefits such as higher resource utilization, minimized operational cost, flexible server management, etc. Virtual Machines (VM) consolidation (i.e., the process to move around VMs among the datacenter Physical Machines (PMs) in order to reduce the total power consumption) is supported by the process of VM Live Migration, which transfers e.g. CPU, memory and disc states from one physical host to another with minimum service interruption. In order to save energy, it is imperative to reduce the total number of powered on servers

¹ <http://www.datacenterdynamics.com/news/facebook-datacenters-energy-use-up-in-2012/80642.fullarticle>

required in a datacenter by applying VM consolidation. As resource demands of applications are typically much lower during nights and weekends, a smaller set of servers would be sufficient during off-hours to host the given VMs. As a consequence, an important method to reduce the energy consumption is the consolidation of the VMs on the minimum number of physical rack servers that are required for the requested resources and the powering down of the unused ones. However, such consolidation has to be achieved without possibly violating the Service Level Agreements (SLA). Additionally, aggressive consolidation of VMs, due to e.g. overbooking the given host may lead to significant performance degradation if many of those VMs run at their peak demand concurrently. Therefore, finding an optimal allocation of VMs to support their resource demands on the given set of physical servers in order to minimize e.g. energy consumption is a very hard computational problem and has led to a number of interesting mathematical modeling approaches in recent years [29] [41] [19] [6] [32].

Common to all those models is the assumption that input data that drives those models is known precisely, which is very difficult to achieve in practice. For example, it is difficult to quantify exactly the required resources of each VM, as modern cloud applications often have highly variable workloads which can lead to dynamic resource usage patterns. Due to the complex architecture of modern servers, precise estimation of servers' power consumption is also becoming a hard task. Furthermore, the resource overhead due to VM migrations is complicated to predict as they often depend on their current workload and memory dirty patterns. Therefore, predicting all these design parameters beforehand in a precise way is a very difficult task in the real world. Unfortunately, the presence of uncertain data in an optimization problem may lead to solutions that are useless in practice [7] [10]. This is because small deviations in input data values may lead to situations where a found optimal solution is even not feasible any more. As a consequence, we need to develop models that allow to work with data uncertainty such as Stochastic Programming or Robust Optimization (RO). Specifically, we try to answer the following research questions: *1) How can we model the uncertainty in a modern datacenter using robust optimization techniques? 2) What is the impact of different parameter uncertainties (such as non-deterministic and non-stationary resource demands, migration related overhead, etc.) on the VM consolidation problem?*

In this paper, we build a mathematical model based on the theory of Robust Optimization (RO) for the problem of energy efficient VM consolidation in modern datacenters under the assumption that we do not know the input to the model precisely. Unlike [57], we consider uncertainty on several input parameters and also integrate the possibility of overbooking into our model. By using a so-called "row-wise

uncertainty" data model, datacenter operators are given a tool to specify their risk aversion. This leads to the so-called price of robustness [9] where the optimal value of the robust model counterpart may in general lead to higher energy consumption compared to the optimal value of the original deterministic problem. Therefore, the main contributions of this work can be highlighted as:

- We formulate a novel multi-objective robust optimization model that optimally solves the problem of VM consolidation under random and non-stationary resource demands and overhead due to VM migrations, given that we know the uncertain parameter bounds.
- Our model integrates the possibility of overbooking resources and demonstrates the relation between energy-efficiency and risk of SLA violations under the fluctuations of VMs resource demands which helps the datacenter operators to set proper overbooking ratios.
- We investigate real workload traces from a small datacenter that illustrate that CPU demands of different VMs vary over time but it is very unlikely that all fluctuates at the peak at the same time. Further, the workload traces indicate that the fluctuations of the resource demands are bounded by a range.
- We present extensive analysis on the impact of several uncertain parameters - such as power consumption of the PMs, resource demands of the VMs, migrations related resource overhead in order to study the trade-offs in terms of operational cost (energy cost) and SLA violations in modern datacenters.

Our model helps to improve the quality of VM placement decisions in scenarios where workloads are varying and non-stationary but bounded. In addition, our model allows to calculate an upper bound for the probability that constraints are violated for a given protection level and the price of the robustness of such protection in terms of additional energy needed to power on more servers to cope with demand uncertainty.

The remainder of this paper is structured as follows. Section 2 reviews related work and point out novelties of our work. In Section 3, we describe different sources of possible uncertainties in modern datacenters and the problem of VM consolidation under uncertainty. Section 4 provides the detailed background on robust optimization theory from Bertsimas et al. used in this work. Our problem formulation is detailed in Section 5. An extensive numerical evaluation with different scenarios is described in Section 6, which is useful to understand the trade-offs that our model allows to calculate. The impact of the uncertainty on the power consumed by a server, on the amount of CPU needed to run a VM and on the amount of resource overhead due to migrations is analyzed; also, the impact of more aggressive vs.

more relaxed overbooking strategies is studied. Finally, Section 7 concludes the paper.

2 Related Work

There are several works addressing the problem of VM consolidation in datacenters. The research within the area of VM consolidation is categorized into different subareas based on the focus of the research such as resource utilization, energy efficiency, multi-tier application deployment, load balancing, etc. Some works such as [53, 27, 19] use the technique of live VM migration in order to optimize power consumption of a datacenter. Prior research focusing on optimal resource provision for reducing energy consumption in datacenters can be classified into two distinct categories: 1) assume deterministic or exact values as input to the model, and 2) assume non-deterministic or range of data as input.

2.1 Energy-Efficient Resource Provision

Xiao et al. [53] present a system which uses virtualization technology in order to support dynamic application demands and to minimize the number of active physical machines (PMs) for minimizing the power consumption. The paper also introduces a concept called skewness to measure the utilization of multidimensional resources of the PMs. By minimizing skewness, the authors try to combine different types of workload efficiently so that the number of PMs can be minimized. Xu et al. [54] presents a multi-objective VM consolidation problem in order to minimize power consumption, resource wastage and cost of thermal dissipation simultaneously. Their work maps workloads to VMs and VMs to PMs using a two-level control system, where they apply combinatorial optimization and multi-objective optimization.

Li et al. [27] present an energy-efficient VM placement algorithm while focusing on minimizing the energy consumption. The authors consider physical resources as multi-dimensional and present a multi-dimensional space partition model to balance the utilization of physical resources in order to reduce the number of active PMs and minimize energy consumption. Ohta et al. [35] investigate the problem of optimal VM placement and propose an exact MILP formulation but also develop heuristic solution algorithms. The authors argue that, in order to minimize the power consumption under limited resource capacity, one should consider dynamic workload and also avoid frequent migrations. The authors formulate the problem as a multi-objective optimization problem. Zhao et al. [56] present a heuristic for live VM migration policy which combines the particle swarm optimization (PSO) with the idea of simulated annealing (SA). The main idea is to improve the accuracy of the global

optimum solution provided by the PSO by introducing SA. The authors aim at searching an appropriate destination host for migrating a VM with the aim of reducing total incremental energy consumption for a long period of time.

Beloglazov et al. [6] investigate the energy-efficient resource allocation policies and scheduling algorithms while considering negotiated Quality of Service (QoS) and power usage characteristics. The authors divide the problem into two subproblems (e.g., VM selection and VM placement) and propose different heuristics for dynamic adaptation of VM allocation without depending on the particular workload type. The proposed algorithms are evaluated in an OpenStack environment [5]. In [4], Beloglazov et al. focus on the host overload detection for the VM consolidation problem and try to maximize the mean inter-migration time under a QoS goal based on the model of the markov chain. While they have an exact algorithm assuming known stationary workload, they deal with unknown non-stationary workloads by a heuristic that tries to predict the workload. Murtazaer et al. [32] investigate the problem of server consolidation and propose an algorithm called ‘Sercon’ in order to minimize both the number of migrations and the number of active nodes. The authors compare their algorithm with first fit decreasing (FFD) for the bin packing problem and found that they can reduce the number of migrations by three to five times, at the expenses of a slightly higher power consumption. Marotta et al. [29] also tackle the problem of server consolidation and propose an algorithm based on SA which tries to minimize both the number of migrations and the number of active nodes. The termination criteria is based on the probability to accept a new solution rather than a value for the final temperature, and instead of exploring a large number of solutions in each iterations, the algorithm searches for a unique solution. The authors show that their algorithm outperforms both FFD and Sercon [32].

Wolke et al. [51] present a VM allocation strategy based on the dynamic server allocation problem (DSAP), which is used to calculate a dynamic allocation plan. DSAP can increase the efficiency of the VM consolidation but degrade the application performance due to a potential high number of migrations. Therefore, the reallocation frequency needs to be adjusted so that it does not affect the service quality negatively. Ghribi et al. [19] present two exact algorithms for VM allocation and VM migration with the aim of reducing excessive energy consumption. The authors solve the optimal allocation problem as a bin packing problem with the aim of minimizing power consumption, and solve the VM migration problem with the aim of minimizing both the number of migrations and power consumption with a set of valid constraints. These two approaches are merged together in order to put maximum number of servers in sleep mode. Wu et al. [52] considers dynamic VM consolidation while considering the migration cost and achieving energy efficiency. The

authors propose a combination of greedy heuristic, Best Fit and swapping operation, and introduce an improved grouping genetic algorithm (IGGA) with a goal of higher energy efficiency and lower migration cost. The authors propose to use a consolidation score in order to evaluate the quality of the solutions.

The authors in [47] explore the trade-off between energy savings and service quality for dynamic resource configuration and bursty traffic in a datacenter. The authors introduce a queuing model with a control-level service rate which tracks the changes in the rate of arriving traffic and switching frictions for transitions between energy states with the aim of reducing power consumption in a datacenter. In [46], they present a loss queuing model for planning capacity of the VMs which uses physical resources from the cloud under varying friction cost. The paper shows the optimal resource management control using dynamic programming. Li et al. [26] study the problem of dynamic resource allocation and tries to optimize the objectives for end users, Infrastructure as a Service (IaaS) providers and Software as a Service (SaaS) providers in the area of cloud computing. They combine different layers of service provision within a cloud environment such as IaaS, SaaS, Platform as a Service (PaaS), etc. and provide a joint iterative optimization algorithm for efficient resource allocation. Hieu et al. [34] presents a VM consolidation algorithm with multiple usage prediction (MUP) to estimate long term future utilization and aims to improve the energy efficiency for the datacenters. The authors try to use both current and future predicted workload in order to ensure reliable characterization of overload and underload PMs and hence, avoid the possibility of SLA violations. Li et al. [28] also consider the impact of dynamic workload on the utilization of PMs and predict the migration probability of VMs. Based on that, the authors develop a Bayesian network-based estimation model (BNEM) for live VM migration to estimate workload patterns, predict overload probability and adaptively adjust the overload threshold.

Comments on above works: All the approaches presented above as well as some recent works such as [36] [15] [50] [22] share the common assumption of having an exact knowledge or accurate prediction on the input parameters, such as VM resource demands or migration related overhead; based on this assumption, the authors then calculate an optimal solution or propose a heuristic algorithm. However, once uncertainty in some parameters is present or some parameters cannot be estimated precisely, it is likely that their optimal solution is no longer optimal or the proposed solution may lose its quality; it may even be a totally infeasible one, thus leading to the violation of the constraints. Such constraint violation may manifest itself as e.g. violation of SLAs and unhappy customers. In contrast, our model explicitly takes into account uncertainty and assume that we

only know the bounds on parameter values. This allows us to trade-off different robustness levels during the solution finding process.

2.2 Non-Deterministic Resource Provision

In relation to resource provisioning techniques that deal with uncertainty of data, [43] presents a robust optimization model for proactive capacity planning using robustness on both number of VMs and their CPU demands. The authors claim that RO is a better choice over the stochastic optimization because of its less computational complexity when the uncertain distribution of data is very complex [55]; they also conclude that RO is more realistic to be used in a datacenter in order to achieve energy saving. Zola et al. [57] propose a robust MILP model for energy-efficient VM consolidation under uncertain resource demand of the VMs. The authors present a numerical study which shows the trade-off between two important aspects - operational cost and SLA violations for the cloud operators. Takouna et al. [44] propose a robust consolidation approach to achieve a balance between service quality and power consumption. The proposal consists of three algorithms - over utilized host detection, VM selection and VM placement. All three algorithms consider a robust statistical analysis of historical data of CPU demands of a set of VMs.

Poolal et al. [38] identify the problem of scientific workflow in the context of cloud computing and propose a robust scheduling algorithm to schedule a task in a heterogeneous cloud environment in order to minimize both execution time and cost. The proposed algorithm is robust against uncertainties such as performance variation and scheduling time and offers multi-objective resource allocation policies to add a task based on deadline and budget constraints. Chaisiri et al. [13] present a robust cloud resource provisioning (RCRP) approach while considering the fact that consumers' resource demand and cloud providers resource price both fluctuate. A RO model is formulated and solved while considering various types of uncertainties such as customer demand, provider prices, provider resource availability, etc. The solution is able to reduce both oversubscribed and on demand cost while meeting the model robustness.

Hwang et al. [23] consider the VM workloads as random variables with a known average and standard deviation which may be correlated with each other. The VM consolidation problem is modelled as multi-capacity stochastic bin packing problem. The work proposes a heuristic with hierarchical structure and compared with four other methods - algorithms based on SA, random VM allocation, FFD and PM-to-cluster algorithm. The hierarchical heuristic is comprised of two phases, where in the first phase, VMs are migrated from over-utilized PMs to under-utilized PMs and in

the second phase, if there is no over-utilized PMs, then PMs are consolidated in order to save energy consumption.

2.3 Difference and Benefits of our Modeling Technique

In contrast to the related works presented above, our work, is to our knowledge, the first one to jointly model uncertainty on different input parameters such as VM resource demands, power model of servers, and migration cost. Also, we have integrated the possibility for overbooking the resources of servers into the model. Assuming we know the uncertainty bounds of the parameters, we can calculate optimal solutions for a given protection level. The protection level allows us to specify how much uncertainty our solutions should protect against in terms of simultaneous maximum deviations of number of input parameters. With our model, we are able to study the impact of different parameter uncertainties on the VM placement decisions. The model also allows us to evaluate the impact of overbooking on both energy efficiency and possible performance penalty under demand uncertainty. Additionally, we are also able to quantify the cost of uncertainty in terms of additional energy needed in order to reduce the probability that SLAs are violated. Further, the model helps to analyse the trade-off between the total number of migrations and the total energy savings. Finally, we want to stress that our intention is to calculate exact solutions for the VM migration problem under uncertainty, as the model intends to calculate an optimal VM allocation plan rather than a sub-optimal heuristic. Consequently, our model can serve as a benchmark against which any robust heuristic can be compared.

3 Uncertainty Inside Cloud Datacenters

Due to dynamic nature of modern datacenters, along with diverse applications and rapidly-changing hardware, datacenter operators may observe significant uncertain behaviours which may require proper controls in order to ensure efficient management. Several possible uncertain behaviours in modern datacenters are briefly described below.

3.1 Uncertainty in the Power Model for PMs

A large number of recent works such as [29], present that CPU is the most influencing factor on power consumption for a PM. This can also be seen from Figure 1, which illustrates the breakdown of the total power consumption for different components in Google data centers in 2012 [3]. However, building an accurate power model to estimate the power consumption as a function of the CPU load of a PM is a difficult task. An accurate power model usually requires a

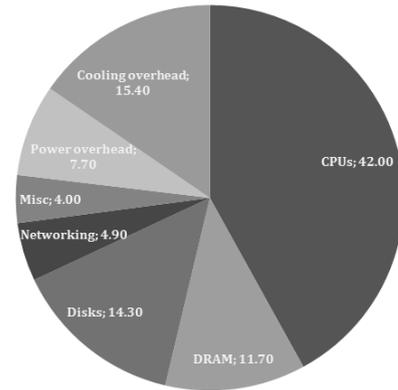


Fig. 1: Power breakdown (%) by components in a Google data center [3]

large number of input data which can cause additional overhead [25]. For example, [30] shows that the power consumption of a PM with a single core can be modeled with 97% accuracy where the accuracy lies within 94% to 98% for a PM with multiple cores. Consequently, a more simple linear model for the power consumption can reduce the accuracy of the model up to 88-90% on average or even higher due to clock throttling, dynamic voltage and frequency scaling, caching effects, etc. Therefore, it is rational to assume that the estimation of power consumption is imprecise to some extent when it follows a linear model.

3.2 Uncertainty in the Resource Demands for VMs

Efficient management of heterogeneous resources in a datacenter is a big challenge due to the dynamic and uncertain nature of resource demands. Hence, given the dynamic nature of user and application demands, it is very difficult to estimate resource demands accurately for most of the cases. Additionally, a small uncertainty in such demand estimates may render an optimal VM allocation a highly infeasible one, may result in SLA violations and thus loss in revenue.

In order to demonstrate such demand fluctuations, we have collected traces from 6 VMs that run in our Computer Centre based on VMWare at Karlstad University (KAU) over 1 week in the begin of June 2015. We have measured CPU utilization in terms of required frequency to run the given VM workload². Table 1 gives an overview on the workloads that the 6 VMs run. Figure 2 shows a CDF of the CPU demands of the different VMs as observed over one week. Each measurement sample represents the average CPU usage, as measured in megahertz, during 30 minute intervals.

² We used the parameter *usagemhz* extracted from vSphere cluster, see https://www.vmware.com/support/developer/vc-sdk/visdk41pubs/ApiReference/cpu_counters.html

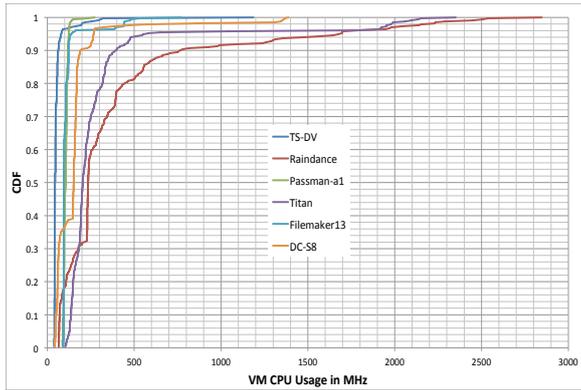


Fig. 2: CDF of CPU usage for 6 VMs over one week at KAU

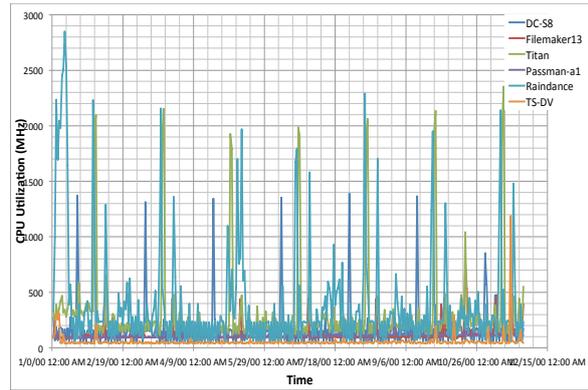


Fig. 3: Actual CPU usage for 6 VMs over one week at KAU

Table 1: Virtual Machine types

DC-S8	Active Directory server for Student domain
Raindance	Runs internal economy system
Filemaker13	Database server, which serves two systems
Titan	one of our Student mail servers
TS-DV	terminal server for CS domain
Passman-a1	Part of our authentication infrastructure

As can be seen, different VMs have different CPU demands, that also vary differently. For example the VM that runs part of the economic system (Raindance) has heavier demands and varies more (between 65 MHz and 2847 MHz), compared to e.g. Passman-a1, which varies between 87 MHz and 273 MHz or Filemaker 13 which varies between 91 MHz and 770 MHz. Interestingly, one can observe for Filemaker that 95% of the values are bounded between 91 and 143 MHz. For Raindance, 95% of the samples are bounded between 65 and 1696 MHz, they have a significantly larger span. Similarly, the CPU demands for the mail server Titan lie between 102 and 544 MHz. This clearly shows that CPU demands vary over time within a given interval most of the time.

Another observation is that all the VMs are not running at their peak at the same time. This is illustrated from our workload traces from the VMSphere centre at our university, where Figure 3 presents the CPU demands of the given 6 VMs over one week. While we see that some VMs peak in the evening (due to backup jobs), not all have their peak demand at the same time. While we acknowledge that our workload traces may be not generalizable to any given VM load, we can assume nevertheless that for large datacenter the probability that all VMs run at their peak loads is small.

3.3 Uncertainty in the Migration-related Overhead

VM migration is one of the most elementary tool for DC operators in order to manage dynamic re-allocations of VMs into PMs. However, each VM migration creates an additional resource overhead during the time of migration. VM migration is an I/O-intensive application, which involves a significant amount of data to be transferred over the network. In addition, VM migration process needs to mirror block devices, maintain device drivers, configuring IP address, etc. which can cause overhead on other resources [51]. Additionally, performance of both source and destination host involved in a migration process, is affected. For example, in [41], the authors present experimental results using different workload scenarios based on real data sets and show that CPU overhead due to migrations may exceed up to 30% to 40% of a VM's actual CPU load. This VM migration related overhead may lead a PM to run into an unexpected overloaded state which can significantly degrade the performance of the VMs which are running inside the PM.

The VM migration depends on several factors such as size of the memory of the VM, the workload characteristics (e.g. memory dirty rate), migration algorithm, network transmission rate, etc. For example, [1] estimates the migration overhead and tries to predict VMs' workload. Consequently, the overhead due to migration may be very difficult to predict precisely, it will rather vary dynamically due to the high diversity and large variation of VMs' workload characteristics over time. As we have seen before, the resource demand of the VMs may vary over time, hence, the resource overhead of these VMs' during migrations can vary as well.

3.4 Uncertain Resource Demands and Overbooking

The objective of overbooking is to both improve the expected profit of the cloud providers but also to ensure higher

utilization of the cloud resources. For example, an analysis of the Google traces [39] shows that CPU utilization is only 40% and memory utilization is only 53% of the available capacity in a typical datacenter. Another study using 5000 Google servers [2] shows that average CPU utilization for most of the hosts within six months duration is 10% to 50%. Similarly, for PlanetLab [37] the average utilization is approximately 22%. The elastic nature of the cloud applications may lead to large fluctuations of resource utilization [45] within a datacenter over time. One reason behind low utilization of the resources is that users put over-provisioned requests for the resources in order to be on the safe side, which utilize only a fraction of the allocated resources.

As a consequence, most of the popular hypervisors such as Xen, KVM, and VMware provide an option for resource overbooking ratio. In addition, open source cloud platforms such as OpenStack, OpenNebula or Eucalyptus allow to enable resource overbooking. For example, OpenStack [18] has by default enabled 16:1 CPU over-commit ratio (one physical core can be overbooked by up to 16 virtual cores) and 1.5:1 over-commit ratio for memory. For memory overbooking, hypervisors use different technique such as memory ballooning [49] which steals memory from underloaded VMs and provides it to overloaded VMs. However, it is very difficult for a cloud provider to decide about the proper capacity management because, on one side, the energy consumption decreases by packing more VMs on a given PM as less PMs need to be powered on; on the other hand, there is a higher risk of potential SLA violations due to performance degradation if all the VMs are running on their peak. Therefore, identifying proper overbooking ratio [12] is one of the most important aspects in datacenter management.

3.5 VM Consolidation Problem under Uncertainty

The primary goal of VM consolidation is to re-allocate a set of VMs onto the fewest possible PMs and powering down the unused ones, with the help of VM migrations. However, VM migrations should not impact the services of all running VMs on the affected hosts. An efficient design for VM consolidation needs to ensure an efficient trade-off between total energy consumption of the PMs and the number of migrations [29]. The VM consolidation problem can be modeled as a multi-dimensional bin packing problem [31] where each VM is considered as an item and the dimensions are capacities of the resources such as CPU, memory, etc. The goal is to minimize the number of active PMs, power down the unused ones to conserve energy but placing VMs in such a way that the constraints in terms of e.g. resource demands are not violated.

This problem is NP hard and usually is solved through the help of mixed integer linear programming (MILP) or

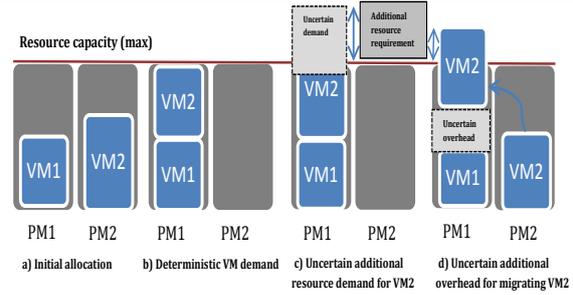


Fig. 4: Effect of uncertainty on VM consolidation problem

heuristic algorithms. Most of the previous approaches assume perfect knowledge on e.g. resource demands of the VMs, migration-related resource overhead, migration time, etc., which is either to be known beforehand or estimated deterministically through the use of prediction algorithms. However, in practice these assumptions are likely to be not accurate that may turn a proposed optimal solution to a highly infeasible or sub-optimal one [9]. Moreover, multiple identical resources may show different performance for the same workload [17]. Therefore, it is very difficult to quantify the design parameters for such an optimization problem exactly. In order to give deep understandings, we describe the impacts of uncertainty on VM allocation decisions in the next two sections.

3.5.1 An Illustrative Example

A small illustrative example using 2 VMs and 2 PMs helps to demonstrate how a VM allocation approach may lead to potential SLA violations when uncertainty is involved. At the beginning, 2 VMs are allocated to 2 different PMs so that the VMs get their demanded share of resources from the assigned PMs. Assume then that the workloads of the 2 VMs are reduced so that it is possible to migrate them together into a single PM and power down the unused one. Figure 4 illustrates the effect of inaccurate knowledge of resource demands and migration overhead for VM2 and shows how deviation on the demand values can turn an optimal allocation into an allocation that may degrade the VMs' performance significantly. For the estimated demand, if it is quantified precisely, placing both of them on the same PM is the optimal solution to reduce energy cost (see b), but uncertainty or erroneous estimation of demands (see c) or migration overhead (see d) may lead to a situation where the PM does not have enough capacity to fulfil both VMs' resource demands, which may lead to potential contention for resources and performance penalty for the affected VMs.

3.5.2 Problem Analysis and Decision Problem

In order to analyse the VM consolidation problem in more detail, we have considered the bin packing problem again.

Now, a set of items are presented as independent random variables without following any known distributions, $R = \{r_1, r_2, \dots, r_n\}$. We assume the probability for exceeding bins' capacity is $0 \leq \delta \leq 1$, the dimension of the items is 1, and the objective is to pack the set of R into the smallest possible number of bins, $S = \{s_1, s_2, \dots, s_j\}$. If $i = \{1, 2, \dots, n\}$ where n is the number of items, $j = \{1, 2, \dots, k\}$ where k is the number of bins and X_j is the subset of items which are packed into the j^{th} bin, then the problem needs to satisfy the following probabilistic constraint:

$$Pr \left[\sum_{i \in X_j} r_i \geq s_j \right] \leq \delta, \quad \forall i \forall j \quad (1)$$

If we relate this with the VM consolidation problem, constraint (1) can be translated into a probabilistic SLA, i.e. the probability that the VMs suffer for contention due to resource shortfall is at most δ . The goal is to tightly allocate VMs into PMs in order to increase energy-efficiency; however in reality, there are many sources of uncertainty that can increase the risk of exceeding δ , thus creating unexpected performance degradation for the VMs due to resource shortfall. Hence, it is very important for the datacenter operators to know how a specific VM placement plan will behave under the presence of uncertainty e.g. fluctuations on VMs' workload or migrations overhead and how the uncertainty can alter the plan. Therefore, in order to propose a robust VM allocation scheme, the datacenter operator should be able to deal with the following questions:

- What is the chance that the proposed migration plan becomes infeasible? What is the probability of exceeding PM's capacity due to uncertain behaviour (e.g. workload fluctuations) of allocated VMs'?
- What is the magnitude of infeasibility? How to control that the magnitude of infeasibility is below a certain target in order to achieve the expected performance of the applications?

The ability of estimating the risk taken by a given migration plan will provide the datacenter operators with a deeper insight into the performance of the proposed plan. Knowing the probability of infeasibility can increase the confidence of the datacenter operator in choosing a plan. The expected shortage on resource allocation gives the datacenter operator a better understanding on how bad things can get and what is the cost of possible actions. If we look back to our previous example (Figure 4), it is hard for the datacenter operator to decide about the number of PMs to allocate for the given VMs. For instance, if one PM is selected then there is a high possibility of revenue loss due to service degradation; on the other hand, if both PMs are selected, then there is a high possibility of increasing operational cost (due to increased power consumption).

Additionally, when multiple VMs are packed into the same PM, the allocation scheme needs to consider both the individual workload of the VMs and the aggregate workload of all the allocated VMs for each host. We also should consider that in reality it is very rare that all allocated VM's for a given PM may simultaneously deviate to their highest demands at the same time. For instance, some of the VMs workload may decrease, which may release resources, while other VMs workload may increase, which may require more resources. Therefore, there is a high chance that aggregate resource demands of the VMs allocated to a PM is still below the capacity. Therefore, in order to make an efficient VM allocation plan, the datacenter operator needs a budget for handling uncertainty that can be acted as a knob to adjust their risk attitude i.e the level of risk they are willing to take in terms of SLA violations.

4 Background on Robust Optimization

Robust Optimization [7, 10, 9] effectively deals with optimization problems where robustness is sought against uncertainty or deterministic variability in input parameters [43]. Unlike stochastic optimization (SO), RO assumes that probability distribution of uncertain data is not known beforehand, rather the uncertain data is assumed to reside in a so called uncertainty set. As a consequence, robust solutions are by construction deterministically immune to realizations of the uncertain parameters in certain sets. This is an interesting approach for problems where the distribution of uncertainty is not available during design time or not stochastic. In addition, many RO problems are tractable which makes RO an interesting approach for practical problems where decision makers are interested in probabilistic guarantees for the robust solution that can be computed a priori. Formally, an uncertain linear optimization problem can be specified as [10]:

$$\begin{aligned} & \text{minimize } c^T x \\ & \text{s.t. } Ax \leq b \end{aligned} \quad (2)$$

where $x \in \mathbb{R}^n$ is a vector of decision variables and the uncertain parameters take arbitrary values from the user specified uncertainty set $U \in \mathbb{R}^m \times n$. The goal is to find minimum cost solutions x^* among all feasible solutions for any realizations taken by the unknown coefficients. The constraints are considered individually so that they are satisfied for all U . It is also possible to design an optimization problem with multiple uncertain design parameters. For example, when looking at the constraint $Ax \leq b$, both A and x can be uncertain. It is also possible to cope with uncertainty in the objective function by introducing a new auxiliary variable $t \in \mathbb{R}$ and

then transform the original problem to minimize t subject to $c^T x - t \leq 0$.

The robust counterpart of (2) is an optimization problem with usually infinitely many constraints, depending on the uncertainty set U . The optimal solutions provided by the robust counterparts (called robust optimal solutions) are usually worse than the optimal solution provided by the original problem because it tries to mitigate the adverse effects of uncertainty on the solution quality. The quality of the solutions can typically be evaluated by the “price of robustness” [9], which indicates the effects on the solutions when they are protected against data deviations.

The decision makers often try to trade-off between robustness against historical realizations of the random variables and the size of the uncertainty set. Different possible uncertainty sets such as box, ellipsoidal, polyhedral, cardinality constrained, cone, etc. and their advantages and disadvantages are described in [21]. The cardinality constrained uncertainty set defined by Bertsimas and Sim is practically relevant as it defines a family of polyhedral uncertainty sets [9, 11] that presents a budget of uncertainty in terms of cardinality constraints; that is, they define the maximum number of parameters that are allowed to deviate from their nominal value, also known as “so-called row-wise uncertainty”. The key benefits of this type of uncertainty set are: 1) a protection level (known as *budget of uncertainty*) against deviations of the coefficients specified by the adopted uncertainty; and 2) the possibility to calculate probabilistic bounds for constraint violation for a given protection level. This type of uncertainty is practical for large number of real world problems as it is not too conservative but still provides a reasonable protection.

Let us consider again the bin packing problem described in Section 3.5.2. Same as before, items are presented as a random variable $R = \{r_1, r_2, \dots, r_n\}$; however, they are now treated as an uncertain variable with a known nominal value \bar{r}_i and a possible symmetric maximum deviation, $\hat{r}_i \geq 0$, thus lying in the interval $[\bar{r}_i - \hat{r}_i, \bar{r}_i + \hat{r}_i]$. We assume that each item is independent and that the deviations of item sizes only affect the item independently. When applying the budget of uncertainty, Γ_i , the RO with cardinality constraint uncertainty set advocates that at most Γ_i items may deviate from their nominal value; thus, Γ_i denotes the budget of uncertainty for constraint i . The robust uncertainty set can be defined as all the items where the sum of the relative deviation from their nominal values is at most Γ_i . More formally, we can define a scaled variation ϕ_i of parameter r_i from its nominal value as:

$$\phi_i = \frac{r_i - \bar{r}_i}{\hat{r}_i} \quad (3)$$

and require

$$|\phi_i| \leq \Gamma_i, \quad \forall i, \quad |\phi_i| \leq 1, \quad \forall i \quad (4)$$

The bin packing problem can be reformulated as a single convex programming problem [11] for any convex uncertainty set, $U = \{(r_i) \mid r_i = \bar{r}_i + \hat{r}_i \phi_i \quad \forall i, \quad \phi \in \Phi\}$. After relaxing and considering the dual of this inner maximization problem, it is possible to represent the bin packing problem as a linear optimization problem which is tractable (for detailed steps, we refer the reader to [11]). Further, the upper bound on the probability of constraint violation for a given uncertainty budget Γ can be calculated as in [9]. In the next section, we describe in detail how this concept of Γ -robustness can be used to build a robust VM consolidation model.

5 Robust Model for the VM Consolidation Problem

In this section, the VM consolidation problem is formally defined as a robust optimization model by starting with the robust model presented in [57]. However, in this paper we extend that model by introducing uncertainty on several aspects such as power model of servers, VM resource demands or VM migration related overhead. In addition, we extend the model to take into account a possible overbooking of physical servers. Table 2 presents all the design parameters for our model and Table 3 describes the robust optimization model in detail.

5.1 Model Formulation

The objective (5) of the proposed model is to both minimize the normalized power consumption and the number of migrations. $0 \leq \alpha \leq 1$ is a weighting factor that allows cloud operators to put more emphasis to conserve the power or minimize the number of migrations. If α is small, we try to reduce the number of migrations while still conserving energy, while a larger α will lead to a more aggressive reduction in energy consumption.

Modeling power consumption with uncertainty for PMs:

As in [29], we adopt a linear power model for the PMs, where the power consumption for a given PM is linearly increasing with the CPU utilization between the idle power (PM is powered on but no VMs are hosted on it) and the maximum power (the PM is running at full load). In our model, the power consumption of a PM consists of three different factors: 1) power consumption of the PM during idle load; 2) linearly dependent power consumption based on the resource utilization of the VMs allocated to the PM

Table 2: Model parameters

Input parameters:	
m	Total number of VMs
n	Total number of Servers
x_{jk}^O	Is 1 if VM k is allocated to server j before consolidation, and 0 otherwise
P_{mi_j}	Initial power consumption for server j
$P_{idle,j}$	Idle power consumption for server j
$P_{max,j}$	Maximum power consumption of server j
r_{ik}	Amount of resource i needed to allocate VM k
$rov_{h_{ik}}$	Overhead for resource i to migrate VM k
s_{ij}	Amount of resource i available at server j
t_{down_k}	Downtime for the migration of VM k
SLA_k	SLAs for the applications running on VM k
η_{ij}	Overbooking of resource i at server j
Γ	Protection level over the uncertain variables
M	A large number
Decision variables:	
y_j	Is 1 if server j is active after consolidation, 0 otherwise
x_{jk}^N	Is 1 if VM k is allocated to server j after consolidation, and 0 otherwise
$allocR_{ij}$	Resource i allocated to server j after consolidation
w_{ij}	Is 1 if resource i on server j is overbooked after consolidation
P_j	Power consumption at server j after consolidation
$z_{jk}^>$	Is 1 if VM k migrates from server j
$z_{jk}^<$	Is 1 if VM k migrates to server j
$uncP_j$	Uncertain power at server j
$uncR_{ik}$	Uncertain demand of resource i for VM k
$uncROV_{ik}$	Uncertain overhead for resource i to migrate VM k

(only CPU utilization is considered in this model and is denoted as u_{ij} for the resource i at server j); 3) an uncertain parameter, which is presented as a random variable $uncP_j$ and symmetrically distributed between $[-\Delta P_j, +\Delta P_j]$ with mean 0.0. The power consumption of a PM is calculated as (6) (Table 3). Constraint (7) ensures that the power consumption lies within the minimum and maximum operational points. Finally, a PM is powered down if no VM is mapped onto that PM. Note, that in our model we support heterogeneous PMs which may have different idle and maximum power consumption.

Modeling uncertainty for resource demand: We adopt data uncertainty on resource requirements of the VMs. We use a random variable to model the uncertainty on resource requirements of a given VM k as $uncR_{ik}$, which is symmetrically distributed between $[-\Delta R_{ik}, +\Delta R_{ik}] \cdot r_{ik}$ and with mean 0.0. r_{ik} represents the fixed amount of resource required by the VM k and we can calculate the utilization of PM j for resource i according to (9), which is used to calculate the power consumed as mentioned above.

Modeling uncertainty for migration-related overhead: We take the migration-related resource overhead ($rov_{h_{ik}}$) into account for the budget constraint of a physical server. Uncertainty on resource overhead due to migration of VM k is modeled through a random variable $uncROV_{ik}$, which is symmetrically distributed between $[-\Delta ROV_{ik}, +\Delta ROV_{ik}] \cdot r_{ik}$ and with mean 0.0. Hence, the total amount of migration-related overhead is calculated as the sum of a fixed overhead $rov_{h_{ik}}$ and the uncertain variable. The budget constraint (11) says that, for each server j and resource i , the amount of resources held by the old assignment and the one produced by the migrations should not exceed the maximum amount of available resources (for overbooking multiplied by overbooking parameter). This includes the uncertain demands of both currently allocated VMs as well as the VMs subject to migration as well as the uncertain overhead they produce due to migrating in or out.

Modeling overbooking of resources for PMs: Let us illustrate first how we model in this paper an overbooked scenario using a simple example where we have given a single PM having a single CPU with 8 cores (pCPU) and we assume that we want to run a set of VMs that require 2 cores (vCPU) each, each one utilized to 100%. Let us assume that we use an overbooking factor of 50%, idle server power consumption of 80W, and maximum power consumption of 200W. If the total demand of the VM is smaller than the capacity, each VM will get allocated the requested resources. For example, if we put 2 VMs on the given host, we would need 4 vCPU out of the total 8 physical ones and our server would not run at full load. The power consumption would be 140W in this case (i.e., 30W increase for each VM running on the server). When we put 4 VMs, the server will run at its full CPU utilization (and thus would consume the maximum power of 200W) and without overbooking, we could not allocate more VMs on that host. However, enabling 50% overbooking on the available CPU, we can run 6 VMs concurrently on the same server while still running at 100% utilization (and thus having maximum power consumption of 200W). Consequently, the VMs will share time slices of the available CPUs and some VMs will get penalized leading to a potential performance penalty for the applications running inside the VMs. The overall power consumption in the datacenter may be reduced when we enable overbooking because, in this case, only one PM needs to be powered on while without overbooking we would need to power on another server. Then, the total power consumption would be 200W for server one and 140W for server two, leading to a total energy consumption of 340W, which is 70% higher compared to the case when we allow overbooking.

In our model, we introduce an asymmetric overbooking ratio, η_{ij} which considers the possible overbooking ratio

Table 3: Robust VM consolidation model with uncertainty

MILP Model:

$$\min f = \alpha \cdot \frac{\sum_{j=1}^n P_j}{\sum_{j=1}^n P_{mi_j}} + (1 - \alpha) \cdot \frac{\sum_{k,j} (\bar{z}_{jk}^> + \bar{z}_{jk}^<)}{m} \quad (5)$$

subject to

$$P_j = P_{idle,j} \cdot y_j + (P_{max,j} - P_{idle,j}) \cdot u_{ij} + uncP_j \cdot y_j, \quad i = CPU \quad (6)$$

$$P_{idle,j} \cdot y_j \leq P_j \leq P_{max,j} \cdot y_j \quad \forall j \quad (7)$$

$$\sum_{k=1}^m (r_{ik} + uncR_{ik}) \cdot x_{jk}^N - s_{ij} \leq M \cdot w_{ij}, \quad (8)$$

$$s_{ij} - \sum_{k=1}^m (r_{ik} + uncR_{ik}) \cdot x_{jk}^N \leq M \cdot (1 - w_{ij}),$$

$$allocR_{ij} \geq \sum_{k=1}^m (r_{ik} + uncR_{ik}) \cdot x_{jk}^N - (M \cdot w_{ij}),$$

$$allocR_{ij} \geq s_{ij} - (M \cdot (1 - w_{ij})), \quad \forall i, \forall j.$$

$$u_{ij} = \frac{allocR_{ij}}{s_{ij}}, \quad \forall i, \forall j \quad (9)$$

$$\sum_{j=1}^n \left| \frac{uncP_j}{\Delta P_j} \right| \leq \Gamma, \quad \left| \frac{uncP_j}{\Delta P_j} \right| \leq 1, \quad \forall j, \quad \Gamma \in \{0, \dots, n\},$$

$$\sum_{k=1}^m \left| \frac{uncR_{ik}}{\Delta R_{ik}} \right| \leq \Gamma, \quad \left| \frac{uncR_{ik}}{\Delta R_{ik}} \right| \leq 1, \quad \forall i, \forall k, \quad \Gamma \in \{0, \dots, m\},$$

$$\sum_{k=1}^m \left| \frac{uncROV_{ik}}{\Delta ROV_{ik}} \right| \leq \Gamma, \quad \left| \frac{uncROV_{ik}}{\Delta ROV_{ik}} \right| \leq 1, \quad \forall i, \forall k, \quad \Gamma \in \{0, \dots, m\} \quad (10)$$

$$\sum_{k=1}^m (r_{ik} \cdot x_{jk}^O + (r_{ik} + uncR_{ik} + rovh_{ik} + uncROV_{ik}) \cdot z_{jk}^< - (r_{ik} + uncR_{ik} + rovh_{ik} + uncROV_{ik}) \cdot z_{jk}^>) \leq \eta_{ij} \cdot (s_{ij} \cdot y_j) \quad \forall j, \forall i \quad (11)$$

$$x_{jk}^O + x_{jk}^N + z_{jk}^> + z_{jk}^< \leq 2,$$

$$x_{jk}^O - (x_{jk}^N + z_{jk}^>) \leq 0, \quad x_{jk}^O + x_{jk}^N \geq b_{jk}, \quad (12)$$

$$x_{jk}^N - (x_{jk}^O + z_{jk}^<) \leq 0, \quad z_{jk}^> + z_{jk}^< \leq b_{jk},$$

$$x_{jk}^N \leq y_i \leq \sum_{j=1}^n x_{jk}^N, \quad \sum_{j=1}^n x_{jk}^N = 1, \quad \forall j, \forall k.$$

$$tdown_k \cdot z_{jk}^> \leq SLA_k, \quad (13)$$

$$tdown_k \cdot z_{jk}^< \leq SLA_k, \quad \forall j, \forall k.$$

of each resource for every PM (i.e., $\eta_{ij} = 1.5$ means 50% maximum allowed overbooking for the resource i on PM j) and a new variable that holds the amount of allocated resources for each given VM. The maximum resource allocation for a given PM is thus smaller or equal to $\eta_{ij} * s_{ij}$. Constraint (8) ensures the accurate value for the amount of allocated resources i at the server j after consolidation ($allocR_{ij}$). The allocated resources of all VMs on a given PM need to be the minimum of the total requested ones (total resource demands of all hosted VMs for the given host),

and the capacity of the host. This is because if the requested resources are less than the host capacity, we can allocate them fully. If the requests are larger than the physical capacity, due to overbooking we can allocate just the proportional resource demands up to the maximum physical capacity leading to less allocated resources than requested. This is written by the following equations in the model, $allocR_{ij} = \text{minimum}(\sum_{k=1}^m (r_{ik} + uncR_{ik}), s_{ij})$. w_{ij} is a binary variable which indicates if the resource i on PM j is overbooked or not and M is a big number known as ‘‘Big-M’’ [40] (also

known as “Big-D” or “Big-R”) which is used to capture the logical constructs that if a PM is overbooked (i.e., more demands than the available capacity) then $allocR_{ij} = s_{ij}$, while $allocR_{ij} = \sum_{k=1}^m (r_{ik} + uncR_{ik})$ otherwise.

Modeling additional constraints and role of ‘ Γ ’: Additional constraints (12) are needed in order to avoid unacceptable combinations of the migration and allocation variables (e.g., x_{jk} old and new both 1 for the same server and VM, or z_{jk} to and from both 1 for the same server and VM). b_{jk} is a binary variable, see also [29] for details. Migrating VMs may break the SLA requirements for the applications running on the VMs because of potential high downtime. Constraint (13) prevents the migrations of those VMs which have strict SLA requirements but the downtime due to migrations may be not acceptable for them.

We define Γ as the protection level against uncertainty. Constraint (10) impose that the sum of the deviations of the uncertain coefficients should be smaller than Γ , as defined by Bertsimas et al. in [10]. A natural interpretation is given for integer values of Γ , where it can be interpreted as the maximum number of parameters that may deviate from their nominal values. For example, if $\Gamma = 0$, there is no protection against uncertainty at all and all robust parameters are assumed to take on their nominal values. This would for example mean that we can precisely quantify the resource demands for all VMs, etc. If Γ is at the maximum, each constraint is protected against maximum uncertainty, leading to a very conservative solution which has the highest energy consumption. For all intermediate values of Γ , the datacenter operator can thus tune against how much of this variability on the whole system he wants to protect from. For example, when $\Gamma = 2$ and we only consider uncertainty on the resource demands of VMs, we protect the solution against any possible maximum deviation of resource demands of at most two VMs out of the total set of all VMs.

In summary, cloud operators can use Γ to reflect their hosted applications characteristics and connect them with operational cost and associated performance for the applications. Since cloud applications may experience time-varying diversity of workload, therefore, cloud operators can utilize Γ to reduce the overall capacity needed to support the aggregated demand of all applications through optimal allocations. Further discussion on the trade-offs between cost and SLA that we can make with our model by varying Γ is presented in Section 6.

5.2 Modeling Assumptions and Simplifications

The proposed model aims to calculate optimal allocation of the VMs in a datacenter where many parameters e.g. workload of the VMs, the migrations related resource overhead, or power consumption of the PMs are uncertain. It is worth

to mention that the model is based on a few fundamental assumptions. First of all, it is assumed that all the uncertain parameters are independent. In other words, we assume that the probability that all the VMs in a datacenter run at their peak load at the same time is small. This is certainly the case for large data centers where different VMs are allocated to different customers implementing different services.

Another assumption is that for all uncertain input parameters of the model, we know the maximum deviation from their nominal values. For the uncertain parameters on VM workload, such statistics can be provided by cloud monitoring solutions that observe workload history over time and thus would allow to provide estimates about probability of certain workload bounds. Otherwise, it would be rather easy to implement extensions to hypervisors that allocate not more than a certain number of time slices to those VMs and the VMs specify the maximum number of time slices they want to use. For the power model, measurements have been done that allow to quantify the maximum deviation of the actual power consumption from the linear power model as mentioned above. Finally, for the migration allowed uncertainty, one could implement extensions to the hypervisor that would limit the overhead due to migrations. Such limit may however lead to longer downtime or time to migrate.

We also assume a rather simple model for the energy aware VM consolidation and we explicitly acknowledge the fact that our simple model may not perfectly capture real data centers. We had to keep our model simple in order to make it tractable because we want to study the effects of parameter uncertainties on the exact solutions. For example, we adopt a linear power model for the PMs which is only based on CPU utilization, and we don’t consider any other parameters such as size of the cache, number of cache accesses, the number of the last level cache misses, etc. Secondly, our model for overbooking only considers over-subscription of the available resources and does not take into account any advanced feature such as sharing memory segments between VMs, sharing a single cache or having separate Level 2 caches for each core, etc. Finally, our model is temporal-oblivious that is, it is not intended to optimize VM-to-PM allocations over time.

With the clear understanding of the model assumptions and simplifications, and a careful assessment of the applicability of our model to a specific data center, an application of the model can nevertheless bring substantial benefits for the cloud operators in terms of balancing performance and power cost as we will see in the next section.

6 Numerical Results and Uncertainty Analysis

We implemented our robust optimization model with ROME (Robust Optimization Made Easy) [20], which is an algebraic modeling toolkit for Matlab. ROME operates as an

intermediate layer between the modeler and the optimization solver and translates the modeling code into a solver specific input format. For all the experiments, we used the IBM ILOG CPLEX 12.6.0 [24] as the optimization solver. One parameters $epgap$ is tuned for the experiments in order to set a relative tolerance for the gap between the best relaxed and best integer solution. In the following, we solve the model for the optimum migration schedule depending on different input parameters and uncertainty settings. We understand that the time to solve for large model instances is prohibitive for online optimization and fast heuristics are desired. However, this is not the goal of this paper. We rather are interested to calculate an optimal solution under different uncertainty settings which can be used to benchmark any heuristic later on.

6.1 Evaluation Scenarios and Parameter Settings

Regarding the number of PMs, in Section 6.2, where we investigate the impact of uncertainty on the power model, we have used a set of 50 PMs (which is approximately the common size of a rack in a datacenter [3]) and assume that each PM is equipped with different amounts of physical CPU and memory. The parameters related to PMs' power consumption such as maximum power consumption (P_{max}), idle power consumption (P_{idle}) are taken from [16], where two types of physical servers (Lynx Calleo 1240, and IBM x3550 M4) are used for an OpenStack testbed setup. However, unlike them, we take a range of values for the parameter P_{max} and P_{idle} in order to present PMs which are invariably heterogeneous in terms of their power consumption. The values are presented in Table 4. Note that the idle power is presented as a percentage of the maximum power consumption. In order to set uncertainty on power, we have chosen two different maximum deviations for ΔP_j which is $\pm 5\%$ and $\pm 10\%$. This values are taken from [30] which shows that a linear model for power consumption of a multi-core PM has an accuracy ranging from 94% to 98%.

Regarding the number of VMs in Section 6.2, we have considered 150 VMs and initially, they are allocated to PMs randomly while making sure that each VM is allocated to at most one of the PMs that have enough resource capacity to allocate that VM. After initial allocation, there are 44 active PMs where 150 VMs are allocated. For all the other experiments (Section 6.3-6.6), we consider a set of 100 VMs and 14 PMs; after running the random initialization algorithm, 12 of the PMs are powered on. For our experiments, the number of PMs and VMs are selected in such a way that it can represent a small to medium size data center while keeping the solution time in a reasonable boundary. As our model is too complex to solve for a whole data center having 100,000s of VMs, one either needs to develop fast solution

Table 4: Settings for the PMs and VMs

Resources	Values (PMs)	Values (VMs)
CPU Requirements (r_{1k})	1.0 - 8.0	0.10 - 1.00
Memory Requirements (r_{2k})	1.0 - 8.0	0.10 - 1.00
Power Consumption (P_{max})	160, 180, 190, 220, 240, 260, 270, 280, 290	-
Idle Power (%)	20 30 40	-
SLA (Sec)(SLA_k)	-	10, 20
Downtime (Sec) (t_{downk})	-	10, 20

heuristics or we can apply our model for each rack separately (leading to sub-optimal solutions).

The average resource demand (r_{ik}) for the VMs are taken from [29] and the uncertain range is taken from [48]. The 11,776 24-hour long real-world workload traces from more than a thousand VMs that are hosted on PMs which are located on more than 500 places around the world, are analyzed [48] to obtain statistical indicators such as maximum, minimum CPU usage, standard deviation, etc. The authors mention that the standard deviation range varies from 26.34 to 43.5875. Hence, we have selected the uncertain range ($uncR_{1k}$) of CPU demand from 30% to 50%. Regarding migration overhead, the uncertainty range ($uncROV_{ik}$) is selected to be either 10%, 20% or 30% which has also been confirmed by several experiments with real data sets in [41]. In our case study, the migration downtime (t_{downk}) is taken from [1].

For all the experiments in Sections 6.2-6.4, α is set to 0.9 as [29] and η is set to 1. In Section 6.5, we explore the impact of overbooking factors by increasing η up to 2. In our experiments, we investigate and try to distinguish the SLA violations due to uncertainty and due to allowed overbooking. Therefore, η is selected as 1 to represent non-overbooking scenario and additionally, the other values for η (e.g. 1.25, 1.50, or 2.0) are selected to define maximum allowed overbooking on PMs' capacity. The overbooking factors are chosen in relation with the VMs' workload fluctuations as we are interested in exploring the relationship among VMs' workload fluctuations ($uncR_{ik}$), allowed overbooking (η) and the level of protection (Γ). The impact of weighting factors in the objective function is explored in Section 6.6 by setting the values for α to 0.1, 0.5 or 0.9. The values for all the parameters in our experimental study are summarized in Table 4 and all the input data are available online³ to reproduce results.

³ https://1drv.ms/f/s!AiNfwK2_wAzqjMdeDfdlObemsju2g

6.2 Impact of Uncertainty in PMs' Power Consumption

In this section, we evaluate the impact of adding uncertainty on the power model parameters for the PMs. Under power model uncertainty, the power consumption of a PM depends not only on its CPU utilization but also on a symmetrically distributed uncertain factor, as explained in Section 5.1. Figure 5 and Figure 6 show numerical results for different protection levels (varying Γ from 0 to 50) under different maximum uncertainties (5% and 10%) on power consumption of servers. Figure 5 also illustrates the effect of the protection level Γ on the total energy consumption (left axis: relative power consumption after consolidation when compared to the original power consumption before consolidation) and the upper bound for the probability of constraint violation (right axis). Note, that when $\Gamma = 0$, there is no protection against uncertainty assumed and we end up with the deterministic solution. However, $\Gamma > 0$ calculate solutions to protect against deviations of Γ uncertain variables, that is the PM power consumption in this case. For example, if we set $\Gamma = 10$ we protect the solution against a maximum deviation of 10 servers from the linear power function, no matter which servers there are. The upper bound for the probability of constraint violation indicates the maximum probability that a constraint will be violated, if total power deviations of the PMs is more than Γ . For instance, for $\Gamma = 1$, the probability of constraint violation assumes the maximum probability that the constraint (7) may be violated for some of the PMs, if the sum of the absolute normalized power deviations of the PMs is greater than 1. We also observe the time to solve the robust problems to optimality (with 0.3% tolerance) using CPLEX on a Intel(R) Core(TM) i7-4770 CPU @ 3.40GHz, which ranges from 952 sec to 1280 sec for a given Γ .

The risk adjusted and expected power relative to the initial power consumption are shown in Figure 5. The expected power is calculated by taking the solution of the robust model and assuming that all parameters are at the nominal values while the risk adjusted power denotes the value of the objective function for the robust model. For all the cases, both the risk adjusted and expected power consumption increases with the level of protection, while at the same time the maximum probability of constraint violation is decreasing. However, the increase rate is higher when the maximum allowed deviation is higher. It is interesting to note that after some point, an increase in the protection level does not affect the gap between risk adjusted and expected power significantly. For instance, when 5% maximum deviation is allowed for the deviation, the risk adjusted power consumption is not increasing any more for $\Gamma \geq 10$, where the same trend is observed for $\Gamma \geq 20$ when maximum allowed deviation is equal to 10%. This is directly reflected in the solution when analyzing the number of active nodes after consolidation.

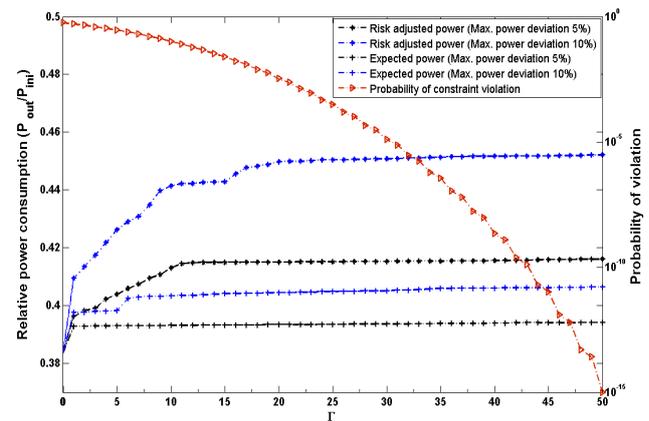


Fig. 5: Relative power consumption and upper bound for the probability of constraint violation for different protection level, Γ for different maximum power uncertainty.

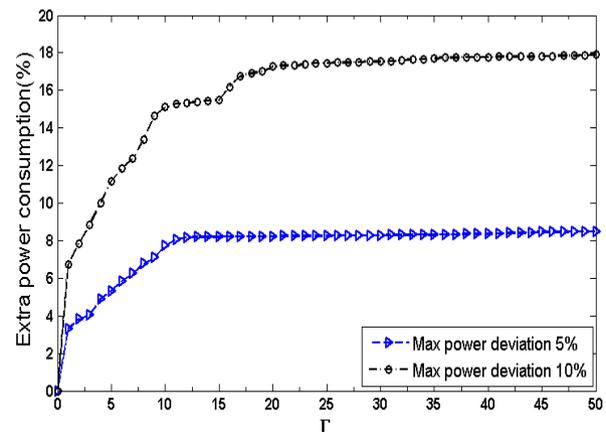


Fig. 6: Extra power consumption (%) with respect to increasing protection level, Γ for different maximum power uncertainty.

For example, when considering 5% maximum deviation in uncertainty in the power model, the number of active nodes increases from 11 to 12 for increasing Γ while it increases from 11 to 13 nodes for 10% maximum deviation. After that, no more server is needed to cope with the uncertainty.

Figure 6 depicts how much additional power is needed (in %) for different protection level Γ for both 5% and 10% maximum allowed deviation on power consumption. For example, for $\Gamma = 50$ and 10% maximum allowed deviation, we can observe that we need to spend approximately 18% more power for protecting against the uncertainty. This is closely related to the risk measures which indicate the extra power requirements in order to avoid constraint violation for a given uncertainty budget. This gives the cloud operators a nice tool to trade-off risk versus additional power required in

Table 5: The trade-off between risk and power consumption for the robust solutions with 5% power uncertainty

Γ	Risk Adjusted Power (W)	Power Increase(%)	Prob. Violation
0.0	2570.97	0.00	0.558996459042231
1.0	2656.01	3.31	0.502577500687455
2.0	2669.19	3.82	0.446158542332680
13.0	2781.28	8.18	0.046218279890685
18.0	2782.13	8.21	0.007721104945138
25.0	2783.07	8.25	0.000312812309172
31.0	2784.17	8.29	0.000007436916207
40.0	2785.82	8.36	0.000000002141733
45.0	2788.51	8.46	0.000000000010132
50.0	2789.16	8.49	0.000000000000001

order to protect against uncertainty in model parameters. For example, as shown in Table 5, if we take the risk of 0.77% probability of violation at maximum, we need to assume $\Gamma = 18$ which leads to around 8.21% more power consumption compared to no protection at all ($\Gamma = 0$). Interestingly, for lower probability of constraint violation (i.e., more risk-aversion), the amount of additional power required is very small. However, in the absence of protection against data uncertainty ($\Gamma = 0$), the probability that a constraint will be violated is 55.89%.

6.3 Impact of Uncertainty in VMs' Resource Demands

In this section, we evaluate two different aspects. First, we study the impact of uncertain resource demands on total energy consumption inside a datacenter while in the second part we study the potential adverse effect of uncertain demands on VMs' performance.

6.3.1 Impact on Energy Consumption

Under CPU demand uncertainty for the VMs, we introduce an uncertain range centered around the average demand which is either 30%, 40% or 50% of the CPU demand (denoted as r_{1k} in our model). So, instead of knowing or predicting that the demand of a given VM is precisely 20% of a physical CPU, we rather have an estimate of the demand which is bounded between 14% and 26% (for the 30% case) and may fluctuate at any time within this bound.

Figure 7 shows the risk adjusted power consumption relative to the initial power consumption (given by the initial allocation before consolidation) on the Z-axis, with respect to different maximum CPU uncertainties (on the Y-axis) and the protection level Γ (on the X-axis). For $\Gamma = 0$, the risk adjusted power consumption is the same for any maximum allowed CPU deviation because no protection is assumed and all values are at their nominal values. As can be seen,

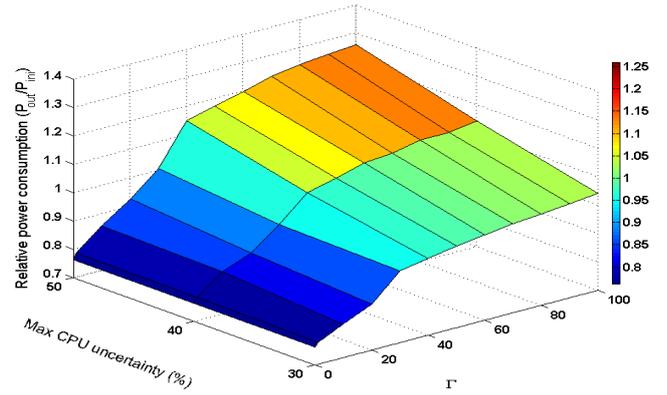


Fig. 7: Relative risk adjusted power consumption for different maximum CPU uncertainties.

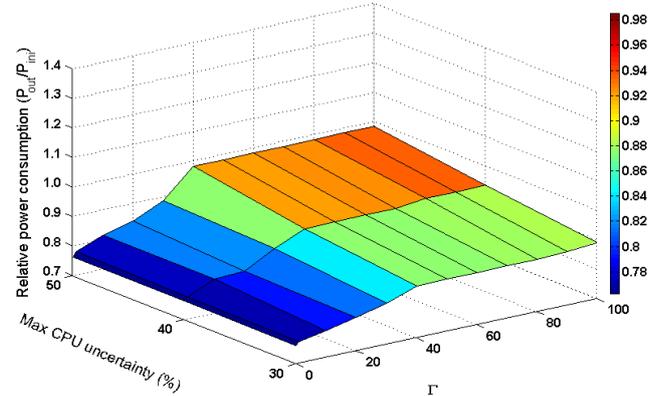


Fig. 8: Relative expected power consumption for different maximum CPU uncertainties.

the power consumption is increasing with the level of protection Γ because the new VM to PM assignment provides more room for accommodating the uncertain demand thus powering on more and more servers. As a consequence, the higher the relative CPU uncertainty (increasing from 30% to 40% and 50%), the higher the power consumption. For instance, for 30% maximum CPU uncertainty, the relative risk adjusted power increases 1.79% (from 0.762 to 0.776) for $\Gamma = 0$ to $\Gamma = 1$ whereas for 40% maximum CPU uncertainty the power consumption increases 2.37% (from 0.762 to 0.780) for the same value of Γ , while for 50% maximum allowed deviation the increase in power consumption is the highest (2.80%, from 0.762 to 0.785). Note, that for each Γ , our solution may result in a different migration schedule in order to protect against different uncertain demands, leading to different power consumption and different VM placement after the consolidation.

Table 6: The trade-off between risk and power consumption for the robust solutions with different CPU uncertainty

Γ	Power Increase (%)			Prob. Violation
	30%	40%	50%	
0.0	0.00	0.00	0.00	0.5411630030000
1.0	1.80	2.38	2.79	0.5012687750000
10.0	6.93	11.22	12.17	0.1845760000000
20.0	13.05	16.29	21.19	0.0285000000000
30.0	24.78	26.23	32.02	0.0017000000000
40.0	27.20	37.51	50.48	0.0000300000000
50.0	29.45	40.57	53.99	0.0000020000000
60.0	31.28	44.25	57.12	0.0000000050000
70.0	32.96	45.34	60.61	0.0000000000002
80.0	33.92	48.03	62.75	10^{-17}
90.0	35.38	48.21	63.99	10^{-23}
100.0	36.48	49.93	65.22	10^{-30}

Moreover, with increased level of protection for different CPU uncertainties, the power consumption is even surpassing the initial power consumption. For example, for $\Gamma \geq 70$ with maximum 30% CPU uncertainties, for $\Gamma \geq 50$ with maximum 40% CPU uncertainties, $\Gamma \geq 40$ with maximum 50% CPU uncertainties the risk adjusted power consumption is higher than the initial power consumption. Although the risk adjusted power is increasing with the level of protection, the growth rate is showing a different trend for different ranges of Γ . As we can see, the risk adjusted power consumption increases rapidly for the range zero to medium level of protection ($\Gamma = 0$ to 50). However, for larger level of protection ($\Gamma = 100$), the power consumption is increasing very slowly. For instance, when maximum allowed deviation for CPU is 30%, the relative risk adjusted power consumption increases 29.45% for $\Gamma = 0$ to 50 but after that it increases only 7.03% for the highest level of protection. This is because after some point, there are enough servers powered on to provide free resources to cope with the uncertain demand and the additional nodes to be powered on is very small to cope with the increased protection level. Figure 8 shows the relative expected power, which is slightly increasing with Γ and with the maximum allowed uncertainty for CPU. For instance, for maximum 50% CPU uncertainty, the expected power increases only 0.29% for $\Gamma = 70$ to 100.

The upper bound for the probability of constraint violation is presented in Table 6. A good trade-off between the amount of risk taken and the robustness of the solution is given for example for Γ between 20 and 30 for which the upper bound for constraint violation is between 2.8% and 0.1%, but the energy consumption increases only 13-24% (for 30% maximum CPU uncertainty), 16-26% (for 40% maximum CPU uncertainty) and 21-32% (for 50% maximum CPU uncertainty) compared to having no protection against CPU demand variation at all.

Figure 9 illustrates how much extra power is required in order to limit the maximum constraint violation for differ-

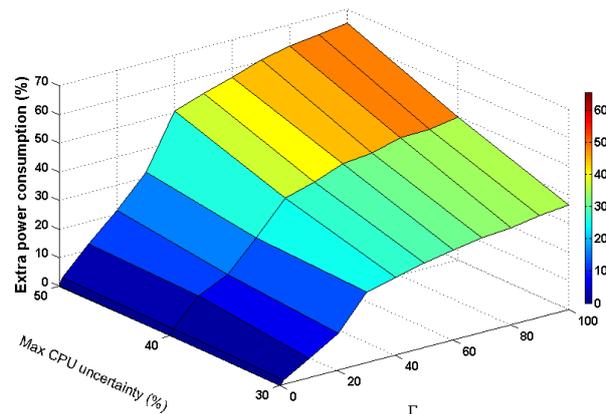


Fig. 9: Extra power consumption (%) for different maximum CPU uncertainties.

ent (30%, 40% and 50%) maximum allowed deviation on CPU demand. This risk measure provides a deeper insight on the power consumption for the proposed VM consolidation plan. As we can see, with higher level of protection there is an increase in power consumption. For example, for the case of $\Gamma = 100$ and 50% maximum allowed CPU deviation, the new allocation is expected to consume 65.23% additional power compared to no protection ($\Gamma = 0$). As expected, the budget for additional power needs for a given protection level is comparatively lower for the case of 30% and 40% maximum deviation on CPU demand. Together with the upper bound for the probability of constraint violation, the cloud operators can now trade-off additional power consumption versus risk.

Quality of the robust solutions: In order to validate the results from ROME and compare with the deterministic problem in [29] where the problem is implemented and solved using CPLEX, we solved in total 1000 deterministic problems (100 for each Γ) varying CPU demands of the VMs randomly but making sure that they are bounded by $\pm 30\%$ of the VMs' CPU requirements (r_{ik}). For example, to set up the deterministic problem when $\Gamma = 10$, we randomly pick 10 VMs out of 100 and for these 10 VMs we assume they have random CPU demands but inside the range as given by the uncertainty set. Figure 10 depicts the corresponding results for 30% maximum CPU uncertainty when varying Γ . The expected power consumption is calculated as before by taking the solution of the robust model and then assuming that CPU demands for all the VM's are at their nominal values. The objective value of the robust model indicates the risk adjusted value which also indicates the possible worst value (maximum power consumption) for the robust optimization problem for a given level Γ of protection. The minimum power consumption is calculated by taking the solution of the robust model while considering that VM

demands are at their minimum value (lower bound). As we have noticed before, with increasing protection level the expected and the maximum power consumption is increasing. However, the minimum power consumption is decreasing. As for the minimum power consumption, with increasing protection level, the number of VMs which are running on their minimum CPU demand are also increasing, therefore the minimum power consumption is showing a decreasing trend.

The calculated power consumptions for the 1000 deterministic problems are presented with coloured points in Figure 10 (e.g. green for $\Gamma = 10$). The results of the deterministic problems indicate the trade-off between risk and power consumption very nicely. For instance, when $\Gamma = 1$, the robust solution protects against uncertainty of demands for 1 VM. However, a large number of the 100 deterministic solutions for $\Gamma = 1$ resulted in a power consumption that was larger than the expected power consumption of the robust model. This corresponds very nicely to a high probability of constraint violation. For the 100 deterministic problems for this case (randomly picked 1 VM and for this VM picked a random VM demand within the interval given and solve that deterministic problem), the average power consumption was 1706.20 W while the maximum power consumption was 1716.04 W and the standard deviation was 3.95 W. But the tendency of consuming a higher power than the expected power provided by the robust model is getting lower with higher values of Γ . For instance, for $\Gamma = 10$, the average power consumption of the 100 deterministic problems was 1721.83 W at a standard deviation of 32.37 W and 15 out of 100 deterministic problems showed higher power consumption than the expected power of the robust solution. However, for $\Gamma \geq 30$, the power consumption of all deterministic problems are always lower than the expected power generated by the robust model. For instance, for $\Gamma = 50$ the average power consumption of the 100 deterministic problems was 1743.11 W at a standard deviation of 47.28 W whereas the expected power for the robust model was 1959.11 W.

Impact of different maximum deviation in VMs' demands: In all scenarios so far, we introduced the same percentage deviation of CPU demands for all VMs. But this may not be the case in the real world where different VMs may have different demand fluctuations for CPU, which clearly depends on the type of workload. This is also motivated by the traces provided in Figure 3. In order to investigate the effects of skewed demand deviations on the results, we now assume a scenario where different VMs have random demand fluctuations while having the same total deviation for all the VMs. We compare the previous scenario of 50% maximum CPU deviation for all the VMs (*scen-50*) with a scenario where each VM has a random deviation, while the average allowed deviation for all the VMs is 50% (*scen-avg50*). The results presented in Figure 11 indicate strong

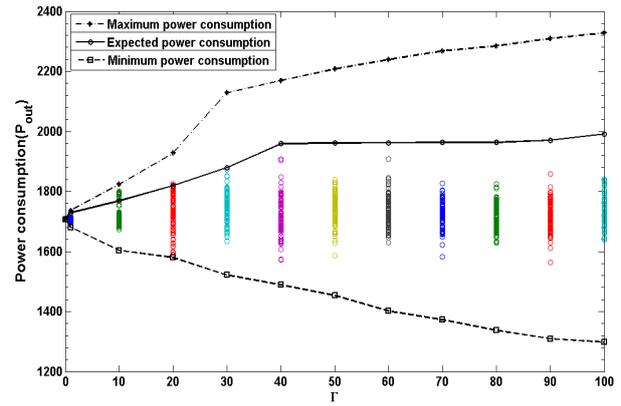


Fig. 10: Empirical results on power consumption for 30% maximum CPU uncertainties.

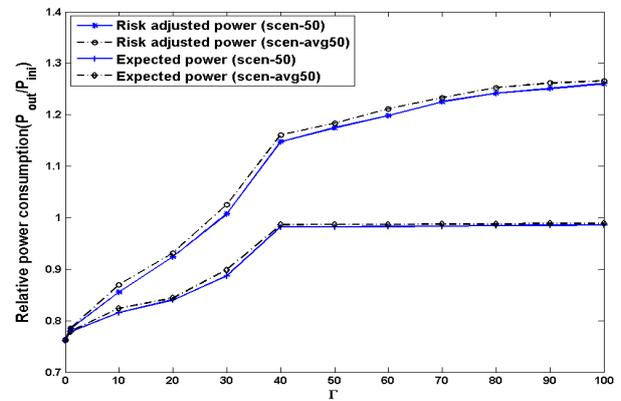


Fig. 11: Relative risk adjusted and expected power consumption for 50% CPU uncertainty with same and different deviation of all VMs.

similarity in risk adjusted and expected power consumption between *scen-50* and *scen-avg50* for different CPU demands. However, when the demand deviations are not distributed equally, the power consumption is in general higher compared to the case where the demand deviations are assumed to be the same percentage for each VM.

6.3.2 Impact on VMs' Performance

In Figure 4, we have presented how uncertain resource demand of VMs' can create potential contention for resources and penalize the performance of the applications running inside the VMs. In the previous section we have discussed how much extra power is consumed with increasing level of protection and we could relate that to the upper bound for probability of constraint violation, which in the case for uncertainty on VM resource demand is interpreted as the potential that a capacity constraint of a physical server is violated.

Table 7: Penalty metrics and power consumption for the robust solutions with 50% CPU uncertainty

Γ	Power		SLAV(%)		Conflict	
	Mean	Std	Mean	Std	Mean	Std
0.0	1749.77	5.32	46.52	2.18	1.92	0.11
1.0	1777.84	6.57	33.12	2.26	1.43	0.10
10.0	1846.24	7.47	22.22	1.00	1.12	0.04
20.0	1943.92	7.62	21.25	0.15	0.80	0.03
30.0	2067.96	7.56	20.00	0.60	0.41	0.03
40.0	2202.56	8.26	0.0	0.0	0.0	0.0
50.0	2202.93	8.42	0.0	0.0	0.0	0.0
60.0	2203.14	8.37	0.0	0.0	0.0	0.0
70.0	2205.27	7.83	0.0	0.0	0.0	0.0
80.0	2207.30	8.68	0.0	0.0	0.0	0.0
90.0	2211.47	8.71	0.0	0.0	0.0	0.0
100.0	2318.98	8.38	0.0	0.0	0.0	0.0

This may lead to situations where VMs are packed tightly on PMs in order to conserve energy but once they deviate in terms of resource demands to the maximum values, contention for resources may arise. The more we increase the protection level, the more energy we will spend to power on new machines that enable us to protect the solution against such VM demand deviations. In this section we try to understand the impact of such potential resource contention due to VM demand uncertainty for different protection levels.

In order to quantify the potential penalty due to uncertainty on VM performance, we are using two different metrics inspired by the ‘‘SLAV (Service Level Agreement Violation)’’ proposed in [6]. The idea is to divide the time into timeslots and calculate the percentage of time the PMs run into a potential contention situation because the total VM demand of all VMs allocated to the given PM is higher than the PM capacity. The second metric tries to guess how severe such potential contention is, inspired by the degree of conflict [42]. In order to evaluate those metrics, we use the following procedure. First, we calculate the optimal solution of the robust optimization problem for each Γ , which gives us a new VM placement after consolidation. For this new placement, we know which VM is running on what host and what host is powered down. We divide time into 15 slots and for each VM and for each slot assume random CPU demands that vary within the specified bounds (e.g. within $\pm 50\%$ of their requested resource demands). Suppose that a PM hosts k VMs after consolidation: VM_1, VM_2, \dots, VM_k . Let $\{r_{i,t1}, r_{i,t2}, \dots, r_{i,t15}\}$ denote the CPU demands of the VM_i from time $t1$ to $t15$. Then, the total CPU demands for all the VMs mapped to a PM is $\{\sum_{s=1}^{15} \sum_{i=1}^k r_{i,t_s}\}$. Therefore, the SLAV for all active nodes (y_j in the model) can be derived as $\frac{\sum_{j=1}^{AN} \sum_{s=1}^{15} \sum_{i=1}^k r_{i,t_s} \times 100}{\sum_{s=1}^{15} AN_s}$, where AN denotes the total number of active nodes that are powered on after consolidation.

The degree of conflict for each active PM is calculated by $\frac{\sum_{s=1}^{15} \sum_{i=1}^k r_{i,t_s}}{\sum_{s=1}^{15} C_s} - 1$, when a PM is overbooked. We aggregate over all active PMs in order to calculate the total degree of conflict for all the time slots. Here ‘C’ denotes the resource capacity of the given PM. We repeated such calculation 100 times by varying CPU demands randomly over each of the 15 time slots for each VM. Table 7 presents the average and standard deviation for the power consumption, SLAV (%) and total conflict degree for the 100 repetitions. We can clearly notice that both SLAV and conflict degree is decreasing with increasing level of protection. When Γ is large, the protection level is high against potential deviations from the mean demand for a large number of VMs’. Consequently, there is less probability that the PMs cannot fulfill the compute demands. On the other hand, for a low protection level, the energy consumption is low, but the risk for SLAV is high as is the conflict degree.

In addition, we also calculate the actual energy consumption averaged over all repetitions based on demand fluctuation given by the random demands within the timeslots. For instance, for $\Gamma = 0$ when there is no protection against uncertainty, the actual power consumption is 1749.77 W which is 2.58% more than the risk adjusted power consumption given by the robust model. Moreover, the SLAV and total conflict degree both are at the peak in this case (SLAV is 46.52% and conflict degree is 1.92 because there is no protection from demand uncertainty but in our simulations we assume that demands actually fluctuate randomly within the given bounds for all VMs. For $\Gamma = 1$ the actual power consumption is 1.35% higher than the risk adjusted power consumption and the penalty metrics are also lower than the previous case (SLAV is 33.12% and conflict degree is 1.43). It is interesting to note that for $\Gamma = 40$ both the penalty metrics are 0 and the actual power consumption is lower than the risk adjusted power consumption. This very nicely illustrates the benefits of our robust optimization approach. When one assumes a perfect prediction of VM demands and does not apply robust optimization, there will be no protection against demand fluctuations. If one would use the optimum VM allocation as given by the non-robust solution, one would have low energy consumption at the expense of a very high potential SLA violation and a large conflict degree. On the other hand, using robust optimization and protecting against potential deviations using e.g. $\Gamma = 40$, the probability to have potential resource contention is very low at the expense of higher energy consumption.⁴

⁴ As a final note, we explicitly acknowledge the fact that our model for potential contention and conflict degree is a simple one and more sophisticated models can be used in order to analyse a potential penalty for applications running inside the VM for a given placement. However, this depends on many factors such as the sensitivity of applications inside the VM to CPU, memory, disc or network contention. While some apps may be penalized severely under e.g. CPU con-

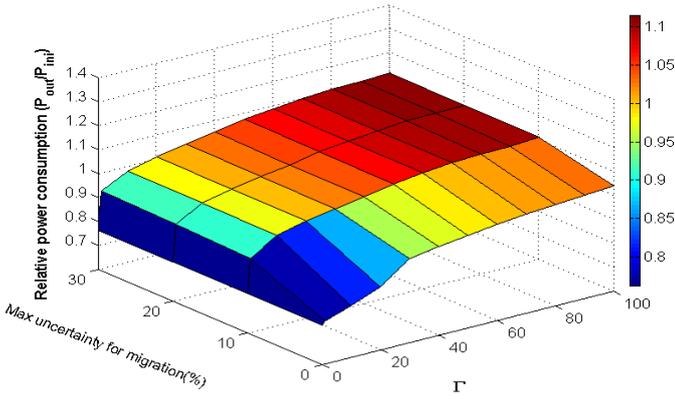


Fig. 12: Relative risk adjusted power when varying migration related overhead uncertainty for a given 30% CPU uncertainty.

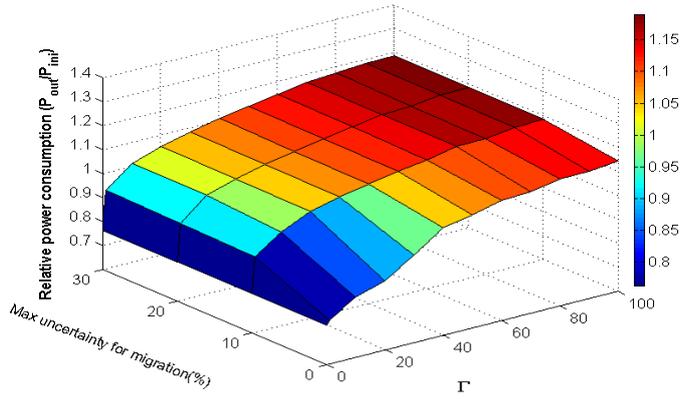


Fig. 13: Relative risk adjusted power when varying migration related overhead uncertainty for a given 40% CPU uncertainty.

6.4 Impact of Uncertainty in both VMs' Resource Demands and Migration-related Resource Overhead

Due to the high stress of VM migration on the PMs [51, 41], VM migrations must be exercised in a careful way in order to avoid unwanted overload of the PMs. In particular, migrations should only be triggered if there are enough resources available on the PM during the time of migrations to cope with the additional migration related overhead such as copying the memory pages and streaming them over the network to the migration target host. In this section, we investigate a set of scenarios where both the amount of CPU demands of the VMs and the amount of resource overhead (both CPU and memory) are not known precisely. For these scenarios, the maximum uncertainty for VM' CPU demands is varied between 30% ($cpu30ovh0$), 40% ($cpu40ovh0$), and 50% ($cpu50ovh0$, called *scen-50* in the previous case) while the maximum variability on the migration-related resource overhead varies between 10% ($cpu30ovh10$, $cpu40ovh10$, $cpu50ovh10$), 20% ($cpu30ovh20$, $cpu40ovh20$, $cpu50ovh20$) and 30% ($cpu30ovh30$, $cpu40ovh30$, $cpu50ovh30$) of VMs' resource demands. The experimental results are presented in Figures 12 to 15.

Figures 12 to 14 show the risk adjusted power consumption relative to the initial power consumption before consolidation for different Γ and different maximum uncertainty bounds on the migration related overhead. Results for a specific ΔR_k is shown in each figure (i.e., 30% in Figure 12, 40% in Figure 13, and 50% in Figure 14). The results show a positive correlation between migration related overhead and power consumption. This is because when the resource overhead due to migration decreases, the cost of a migra-

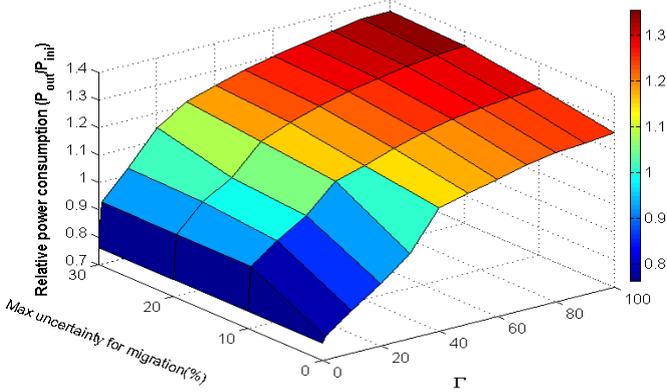


Fig. 14: Relative risk adjusted power when varying migration related overhead uncertainty for a given 50% CPU uncertainty.

tion gets cheaper. For example, an overhead of 0% means that VMs can be migrated without any restriction in order to evacuate some PMs; on the other hand, a large overhead (e.g., 30% on top of VMs' resource demands) would prevent a large number of migrations, which may increase the overall power consumption of the datacenter. Additional uncertainty on the migration related overhead increases power consumption further. For example, in Figure 12, we can see that an increase in the maximum uncertainty bound for migration related overhead from 0% to 10% for a given fixed protection level (e.g. $\Gamma = 1$) leads to an increase in the risk adjusted power consumption of almost 16.46% (from 0.776 to 0.904).

Also, increasing the maximum uncertainty for migration related resource overhead to 30% may increase the risk adjusted power consumption up to 19.11% (from 0.776 to

tion, others might be not affected as much and we are working in the future to take also into account such effects.

0.928). As can be seen, the power consumption is the highest for the case when both of the parameters have maximum uncertainty (in our case 50% for CPU demand and 30% for migration overhead). For this case, even with small values of Γ , the relative risk adjusted power is higher than 1. For instance, for $\Gamma = 10$ the relative risk adjusted power is 1.025 and for $\Gamma = 100$ the risk adjusted power is almost 36% more than the initial power consumption. The relative expected power for the multi uncertainty case is more than 1 when both the uncertain parameters have maximum uncertainty. For instance, for $\Gamma > 30$, the expected power consumption is higher than the initial power consumption (by 4%). This shows that even by calculating the optimal migration schedule, we need to plan for a higher energy consumption to cope with the increased uncertainty.

6.4.1 Required PMs to Deal with Uncertainty

Determining optimal PM capacity configurations while considering uncertainty on different parameters is a challenging problem that has direct impact on the capacity planning for cloud infrastructures. Figure 15 presents the required number of PMs for the four different cases mentioned above (*cpu30ovh0*, *cpu50ovh0*, *cpu30ovh20*, *cpu50ovh30*). When the workload of the VMs is deterministic, the capacity planning is straight forward, as the PMs only need to satisfy the VMs' resource demand. For instance, 8 PMs are enough to allocate all the 100 VMs. However, uncertainty brings additional challenges regarding optimization of cost and capacity in virtualized infrastructure. Key observations from figure 15 are the following:

1. uncertainty in VMs' resource demands leads to allocations that leave enough room on the PM to cope with the uncertainty, again leading to a higher number of PMs that are left powered on. For instance, when VMs' CPU demands deviate up to maximum 30%, the number of required PMs is 10, when the maximum deviation increases to 50% the number of PMs increases to 12.
2. When an additional uncertain parameter is considered for migration related resource overhead, a number of VMs cannot migrate because of potential risk of not coping with the additional stress due to migrations. Furthermore, due to additional resource requirements, the number of required PMs increases again. For instance, for additional 30% uncertain overhead due to migrations, the number of PMs increases to 14. Moreover, the level of protection plays an important role when planning the number of required PMs, as with different protection levels the number of PMs is changing depending on the impact of uncertain parameters. For example, for $\Gamma=10$, the number of PMs is 8 for the case *cpu30ovh0* whereas the number is 9 for the case *cpu50ovh0*, and 12 for the cases *cpu30ovh20* and *cpu30ovh30*.

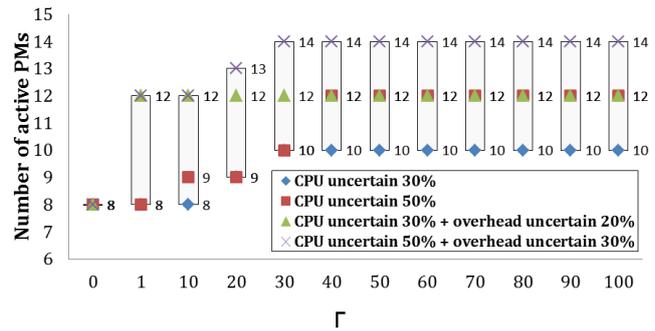


Fig. 15: Number of active nodes for different uncertainties

Our numerical results clearly highlight the limitations of prior works on modeling the infrastructures' capacity planning problem, which assumed complete and exact knowledge on input data (such as resource demands or migration-related resource overhead). Those models clearly may lead to a more energy efficient strategy, but at the risk of violating the SLAs or even increased power consumption, if actual VMs' workload or resource overhead for VM migrations varies from the assumed/predicted ones.

6.5 Impact of Overbooking

In order to highlight the benefits of resource overbooking in relation to energy savings, three different overbooking ratios (25%, 50% and 100% for CPU and memory) are compared with the default non-overbooking technique. We use the same input parameters while the maximum allowed deviation for CPU demand uncertainty is set to 50%. We vary the overbooking factor $\eta_j = 1.25$ or 1.5 or 2.0 . The risk adjusted power consumption relative to the initial power consumption for 50% maximum CPU uncertainty is presented in Figure 16. As expected, the energy consumption is decreasing with increasing level of overbooking as we pack more VMs on a given PM and we can power down more servers. For instance, for $\Gamma = 30$ the relative risk adjusted power for the case of no overbooking is 1.007 (highest) and it decreases gradually when using 25% overbooking to 0.745 and when using 50% overbooking to 0.591. For the case of 100% overbooking, the relative risk adjusted power consumption is the lowest (the value is 0.444). Therefore, for the same scenario, 100% overbooking can save approximately 55.87%, 50% overbooking can save approximately 41.28%, and 25% overbooking can save up to 26% power consumption compared to the case of default non-overbooking. A similar behavior is observed when we consider the relative expected power consumption. Here for all the cases of overbooking, the relative expected power consumption is even

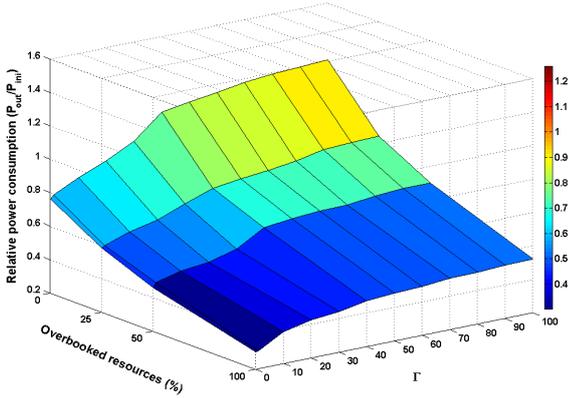


Fig. 16: Relative risk adjusted power consumption with different overbooking ratios and 50% CPU uncertainty.

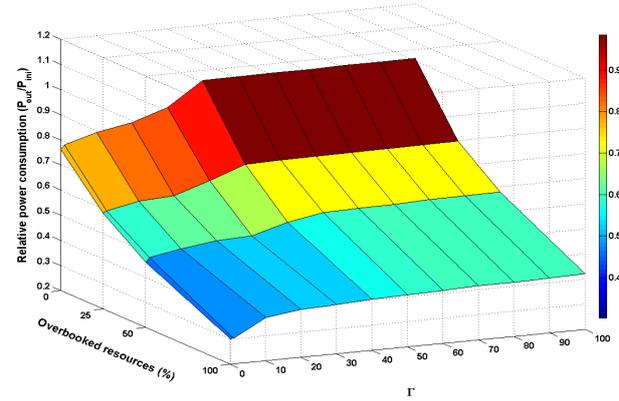


Fig. 17: Relative expected power consumption with different overbooking ratios and 50% CPU uncertainty.

less than 0.8. For example, in the worst case with 25% overbooking the relative power consumption is 0.729 for the maximum protection level ($\Gamma = 100$).

The energy savings obtained due to the overbooking is significant even with large variation on the resource demands of the VMs, but the overbooking scheme may cause significant resource contention which may lead to unpredictable performance for the applications running inside the VMs. The main assumption behind overbooking is that not all the VMs will claim 100% of their resources at the same time. However, the SLA between cloud providers and customers may be violated due to lack of ensuring negotiated level of applications performance, when the VMs are changing their resource demands frequently which can lead a PM to an unwanted overbooking state. In order to quantify the risk of overbooking we compare the penalty metrics SLAV (%) and conflict degree (presented in 6.3.2) with the same set of data as before using 100 repetitions based on the optimal calculated robust VM allocation using the timeslot based evaluation approach. The corresponding results are presented in Figures 18 - 20. It is interesting to note that with increasing level of overbooking, although we can save a higher amount of energy, there is a higher probability of penalizing VMs' performance. We can see that 100% overbooking can save the highest amount of power however at the expense of higher SLA violation and conflict degree.

Figure 18 presents the actual relative power consumption for different overbooking ratios (evaluated using the real demands for each timeslot). We clearly see that the tendency of consuming higher power than the risk adjusted power is getting lower with the higher level of protection. For instance, for $\Gamma = 10$, 100% overbooking can save up to 54%, 50% overbooking can save up to 38.68% and 25% overbooking can save up to 24.58% power consumption compared to the case of non overbooking. But compared to the

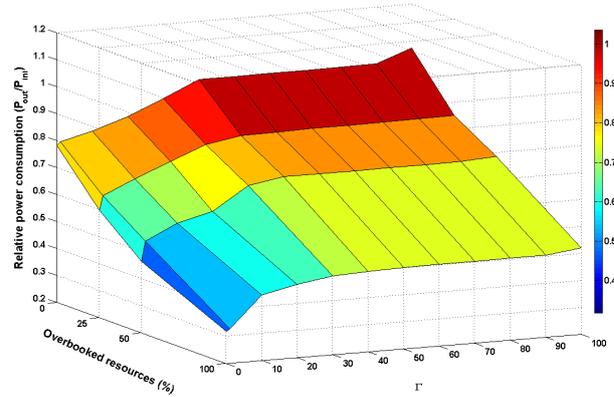


Fig. 18: Relative actual power consumption with different overbooking ratios and 50% CPU uncertainty.

risk adjusted power consumption, the actual power consumption are 9.07%, 12.08% and 13.42% higher for 25%, 50% and 100% overbooking, respectively. As a consequence, if the CPU demands for the VMs fluctuate over time, the more we increase the overbooking ratio, the higher the risk of consuming more power than expected.

Relation between overbooking factor and Γ : Figures 19 - 20 present the SLAV and total conflict degree for different overbooking ratios. Both of these penalty metrics are decreasing with an increasing level of protection for a given overbooking ratio. However, for a given protection level, both of these penalty metrics are increasing with increasing ratio of overbooking. For instance, for 100% overbooking the SLAV is decreasing from 99% to 75.16% and the conflict degree is decreasing from 5.67 to 1.43 for $\Gamma = 0$ to 100. But when $\Gamma = 10$, the SLAV is 22.22% for non overbooking case which is increasing to 63.20% for 25% over-

Table 8: Actual average overbooking for different allowed overbooking and 50% CPU uncertainty

Γ	Overbooking (%)		
	25%	50%	100%
0.0	23.94	46.97	85.57
1.0	22.95	42.81	82.79
10.0	20.07	32.49	69.71
20.0	17.87	31.81	55.05
30.0	13.38	27.72	54.81
40.0	11.94	15.45	52.23
50.0	10.89	15.41	47.23
60.0	6.30	15.34	43.01
70.0	5.38	15.22	41.36
80.0	5.16	14.96	38.92
90.0	5.04	14.68	35.77
100.0	4.84	13.85	29.19

booking, 82.61% for 50% overbooking and finally, 99% for 100% overbooking.

In general, the main challenge in overbooking is to decide about the allocation of excess capacity to minimize the risk of SLA violations when VMs' resource demands fluctuate over time. Our results nicely demonstrate this relation. For example, if all the VMs' CPU demand fluctuates within $\pm 50\%$ of their requested resources, it is reasonable to adapt the robust solution with 25% overbooking and $\Gamma = 60$ as it can save 29.43% power but the SLA violation is only 2.70% with 0.29 conflict degree. Obviously, when the VM's demand fluctuation rate is different, the balance between energy savings and penalty will come at different overbooking ratio and different protection level.

Furthermore, a single fixed overbooking ratio for a highly dynamic heterogeneous cloud environment may not be effective enough to calculate acceptable level of applications' performance with associated operational costs. Therefore, although the maximum allowed overbooking is fixed, the actual overbooking will be adjusted according to the protection level applied. This is because for higher protection level, the allocation of VMs to PMs will be less tight, leaving some spare capacity unused to cope with potential demand uncertainty at the expense of more servers used. Table 8 presents how the actual overbooking ratios are changing with increasing protection level, Γ , and for three different levels of maximum allowed overbooking. For instance, for 25% allowed overbooking, the actual overbooking is decreasing from 23.94% to 4.84% with increasing level of protection. In this case, overbooking is allowed up to 25%, but the actual average overbooking decreases substantially (approximately 19%) for $\Gamma = 100$, as the demand variability is high, the allocation is more conservative and overbooking ratio is comparatively low. The same behaviour is also evident in terms of SLA violation and conflict degree (figures 19 - 20).

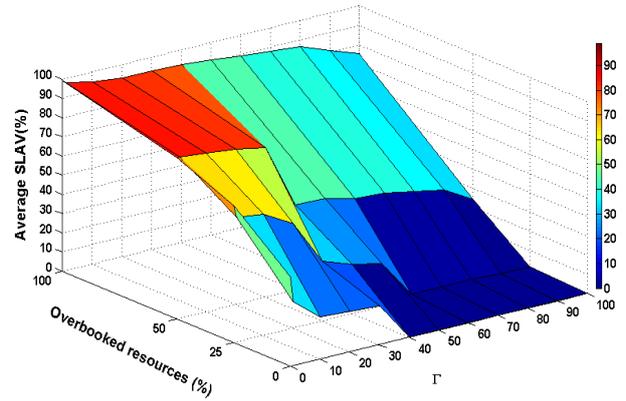


Fig. 19: Average SLAV(%) for different overbooking ratios and 50% CPU uncertainty.

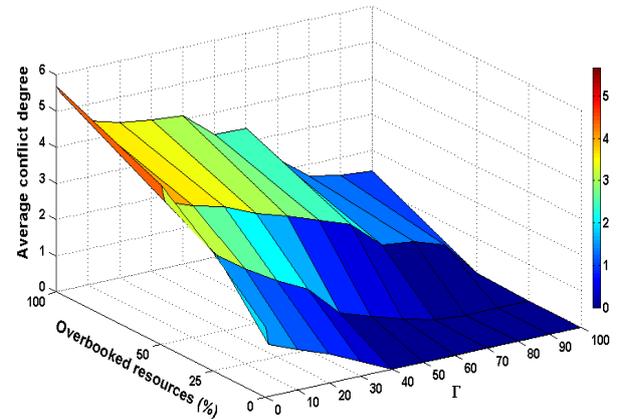


Fig. 20: Average conflict degree for different overbooking ratios and 50% CPU uncertainty.

6.6 Impact of Weights (α) on the Objective Function

In our model, we minimize both the power consumption as well as the number of migrations as migrations put stress on both PMs and the datacenter network. As a consequence, our objective function contains a weighted sum of individual normalized single objectives. By modifying the weighting coefficient α , we can study the trade-off between energy efficiency and the number of migrations. For example, if α is close to 1, we favour more energy efficient VM consolidation strategies while when setting α close to 0 we favour strategies that try to minimize the number of migrations. While all results so far have been computed for $\alpha = 0.9$ (we were rather interested in finding power efficient migration strategies), we analyze in this section the effect of modifying α on two scenarios - *cpu30ovh0* (30% CPU uncertainty and no migration overhead related uncertainty) and *cpu30ovh30*

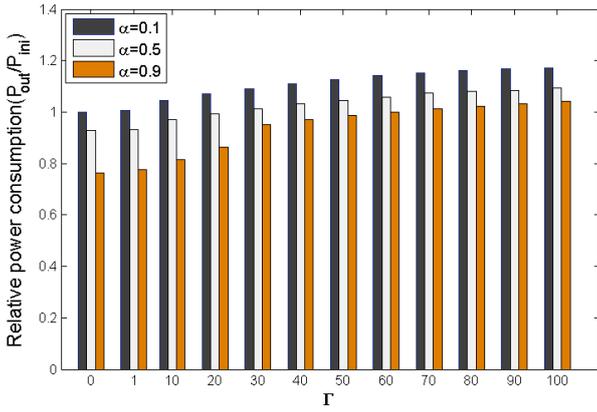


Fig. 21: Relative risk adjusted power consumption for 30% CPU uncertainty with different level of weight (α) on the objective

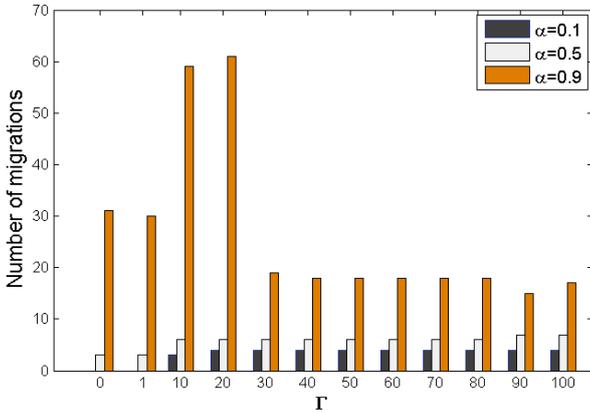


Fig. 22: Number of migrations for 30% CPU uncertainty with different level of weight (α) on the objective

(30% CPU uncertainty and 30% migration overhead related uncertainty) in Figures 21-24.

The relative power consumption for $\alpha = 0.1, 0.5, 0.9$ for the scenario *cpu30ovh0* is presented in Figure 21 and the corresponding number of migrations is depicted in Figure 22. As we can see, the energy consumption and the number of migrations are changing according to the value of α for a given protection level. As expected, the power consumption is higher for $\alpha = 0.1$ compared to the other cases when $\alpha = 0.5$ or 0.9 as for higher α we favour more energy efficient strategies. As also can be seen, the number of migrations are showing the opposite trend as we can observe the lowest number of migrations for $\alpha = 0.1$. For $\alpha = 0.5$, we can see a strategy that is not as power efficient compared to $\alpha = 0.9$ but at the same time not so aggressive in the number of migrations.

When we examine the results in more detail, for e.g. $\Gamma = 10$ the relative power consumption is 1.04 and the number of migrations is 3 for $\alpha = 0.1$ (highest power consumption while lowest migrations); the relative power consumption is 0.96 and the number of migrations is 6 for $\alpha = 0.5$; and the relative power consumption is 0.81 and the number of migrations is 59 for $\alpha = 0.9$ (smallest power consumption while highest number of migrations). It is interesting to note that, when we change the value of α from 0.5 to 0.9, the power consumption decreases around 15.83% but at a cost of 53 additional migrations. But when $\alpha = 0.1$, we cannot save power at all, as the power consumption is even higher than initial consumption due to the uncertainty. The situation is similar for the case with full protection ($\Gamma = 100$), where the power consumption decreases 5% when α changes from 0.5 to 0.9 but we require 10 additional migrations. When we add additional migration overhead related uncertainty, we observe a similar behaviour (*cpu30ovh30* and Figures 23-24) as 30% uncertainty on migration related overhead increases further the power consumption. For instance, under maximum protection ($\Gamma = 100$) when we change α from 0.5 to 0.9, the number of migrations increases from 6 to 13 for the sake of saving only 0.88% power consumption.

As we can see nicely from our results, by changing the weighting factor the cloud operators can either implement a more energy conserving VM consolidation strategy or favour a strategy that tries to minimize the number of migrations in order to reduce stress on the datacenter network. Figures 21 to 24 illustrate the trade-off between relative risk adjusted power consumption and the number of migrations for different level of protection. This helps the cloud operators to decide whether it is worth to improve the energy efficiency at the cost of increasing number of migrations and what extent of increment on the number of migrations is reasonable. When the weight changes (α from 0.9 to 0.1) we get different optimal solutions. Most importantly, if the temporary downtimes due to VM migrations are lower than the value guaranteed upon in the SLA, cloud operators can be benefited by choosing a higher value for α . Further, the applications running inside a datacenter have an important influence on selecting the proper value for α . For instance, if a datacenter is running mostly compute-intensive applications or web-applications, then the temporary service disruption due to VM migrations that the downtime causes is not really important, which can encourage cloud administrators to select higher value for α ; for example, it can be set as 0.9. On the other hand, if Network Function Virtualization (NFV) related services are hosted in the datacenter where different services such as firewalls, deep packet inspection, network address translation, etc. are running inside the VMs, the downtime due to migrations can degrade the applications performance significantly. In this situation, it is very justified to select a lower value for α ; for example,

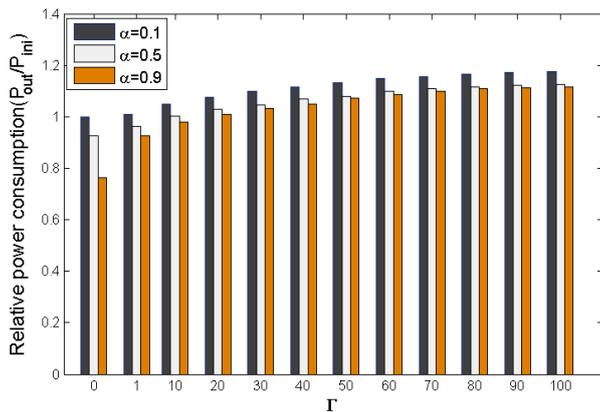


Fig. 23: Relative risk adjusted power consumption for the scenario *cpu30ovh30* with different level of weight (α) on the objective

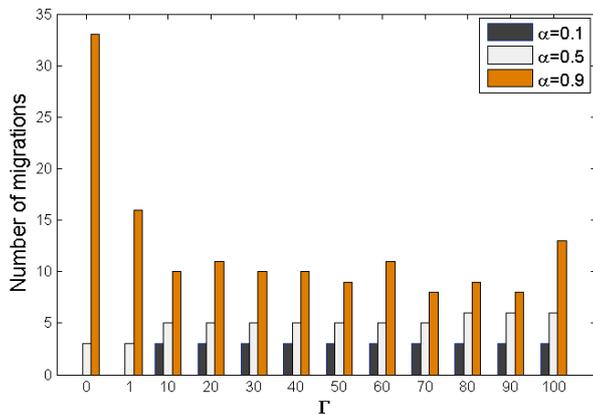


Fig. 24: Number of migrations for the scenario *cpu30ovh30* with different level of weight (α) on the objective

it may be set to 0.1. In that case, we focus less on energy efficiency but rather target at minimizing the migrations. Further, if a datacenter has SLA approaches like auction-based price negotiation in Amazon spot instances [14] and running both compute- and latency-sensitive applications, then the administrator may use a more balanced setting (e.g. $\alpha = 0.5$) and would limit the migration options by setting appropriate SLAs for the downtime for the given VM, which would then be excluded from the migration schedule.

Furthermore, depending on the migration strategies used (e.g. precopy or postcopy) and other network traffic during migrations (such as VM-to-VM communication traffic, management traffic etc.), the temporary downtime due to VM migration can vary [33] significantly. Therefore, these settings also play a critical role for selecting a proper value for α . For example, if a datacenter has implemented an efficient migration strategy [33] which tries to minimize the

impact of migration traffic on VM-to-VM communication or separate migration traffic in order to not disrupt ongoing VM-to-VM latency sensitive traffic, then the cloud operator can choose a more aggressive scheme using a large α in order to conserve power without caring too much on the impact of the downtime. If we compare the results from the three different weight factors (Figures from 21 to 24), we can draw similar conclusions. For example, if we plan to reduce the energy consumption by at least 20%, we need to migrate 30 out of 100 VMs and the value of α needs to be 0.9. However, if we want to reduce the number of migrations, say for example, only 5 migrations are allowed, we can reduce the energy consumption by 10% using $\alpha = 0.5$.

7 Conclusions

Energy is an important and limited resource. Therefore, it is crucial for Green Datacenters to reduce the energy consumption as much as possible while still obeying to SLAs of customers. Reducing energy consumption not only is important for reducing the operational expenses of a datacenter but also is imperative to cope with our limited resources on the planet. As we have seen, the energy can be reduced by leveraging VM live migration technology in order to consolidate the set of VMs on the smallest number of physical hosts and consequently power down the unused ones. While this problem can be modeled as a mixed integer program, it is very difficult to solve not only due to the intrinsic complexity but because it requires exact information as input to the model. Unfortunately, many parameters and coefficients that constitute this problem are not known precisely (such as CPU demands of VMs, the migration related resource overhead or the power model of the servers) which leads to the dilemma that a once calculated optimal solution may be highly infeasible in reality.

In this paper, we have applied the theory of robust optimization for the problem of energy aware VM consolidation under data and coefficient uncertainty. Based on the selected uncertainty model, we were able to calculate for a given protection level both the price of robustness in terms of higher energy consumption as well as the upper bound for the probability of constraint violation. Our numerical results demonstrate very nicely the trade-off a datacenter operator can deal with: by calculating more robust solutions, the resulting strategy leads to higher energy consumption. If the datacenter operator is more risky, our model can calculate more opportunistic solutions that save more energy at the expense of more likely SLA violations.

In addition, we have modeled a potential overbooking for resources and evaluated a potential penalty for services running inside VMs due to uncertain resource demands and overbooking. We have evaluated our model under several scenarios and uncertainty in different parameters of the model

such as CPU demands, migration-related CPU overhead, or power consumption. It is important to remark that the model is not intended for online optimization of a datacenter as it takes a rather long time to solve it for optimality. We were rather interested in solving the model to optimality as a kind of benchmark against which any fast heuristic can be compared.

As future work, we intend to extend our model to include the datacenter network related energy consumption of routers and switches as well as the latency requirements between communicating VMs. This will guide us towards a robust VM consolidation method taking into account network related SLAs between communicating VMs, which is important in order to model a set of VMs that implement NFV related services. Also, we intend to solve larger problem sizes and develop fast heuristics and integrate them into our local OpenStack testbed.

Acknowledgements This research has been funded by KKStiftelsen through the project READY and by the Spanish Government and ERDF through CICYT project TEC2013-48099-C2-1-P.

References

1. Akoush S, Sohan R, Rice A, Moore A, Hopper A (2010) Predicting the Performance of Virtual Machine Migration. In: IEEE International Symposium on Modeling, Analysis Simulation of Computer and Telecommunication Systems (MASCOTS), pp 37–46, DOI 10.1109/MASCOTS.2010.13
2. Barroso L, Holzle U (2007) The Case for Energy-Proportional Computing. *IEEE Computer* 40(12):33–37, DOI 10.1109/MC.2007.443
3. Barroso LA, Clidaras J, Hölzle U (2013) The Datacenter As a Computer: An Introduction to the Design of Warehouse-Scale Machines, 2nd edn. Synthesis Series on Computer Architecture, Morgan and Claypool Publishers, White paper
4. Beloglazov A, Buyya R (2013) Managing Overloaded Hosts for Dynamic Consolidation of Virtual Machines in Cloud Data Centers under Quality of Service Constraints. *IEEE Transactions on Parallel and Distributed Systems* 24(7):1366–1379, DOI 10.1109/TPDS.2012.240
5. Beloglazov A, Buyya R (2015) OpenStack Neat: a framework for dynamic and energy-efficient consolidation of virtual machines in OpenStack clouds. *Concurrency and Computation: Practice and Experience* 27(5):1310–1333, DOI 10.1002/cpe.3314
6. Beloglazov A, Abawajy J, Buyya R (2012) Energy-aware Resource Allocation Heuristics for Efficient Management of Datacenters for Cloud Computing. *Future Generation Computer Systems* 28(5):755 – 768, DOI <http://dx.doi.org/10.1016/j.future.2011.04.017>
7. Ben-Tal A, Ghaoui LE, Nemirovski A (2009) *Robust Optimization*. Princeton University Press, Princeton Series in Applied Mathematics, Princeton, USA
8. Berl A, Gelenbe E, Di Girolamo M, Giuliani G, De Meer H, Dang MQ, Pentikousis K (2010) Energy-Efficient Cloud Computing. *The Computer Journal* 53(7):1045–1051, DOI 10.1093/comjnl/bxp080
9. Bertsimas D, Sim M (2004) The price of robustness. *Operations Research* 52(1):35–53, DOI 10.1287/opre.1030.0065
10. Bertsimas D, Thiele A (2006) *Robust and Data-Driven Optimization: Modern Decision Making Under Uncertainty*, chap 4, pp 95–122. DOI 10.1287/educ.1063.0022
11. Bertsimas D, Brown DB, Caramanis C (2011) *Theory and Applications of Robust Optimization*. SIAM Review 53(3):464–501, DOI 10.1137/080734510
12. Caglar F, Gokhale A (2014) iOverbook: Intelligent Resource-Overbooking to Support Soft Real-Time Applications in the Cloud. In: IEEE 7th International Conference on Cloud Computing (CLOUD), pp 538–545, DOI 10.1109/CLOUD.2014.78
13. Chaisiri S, Lee BS, Niyato D (2010) Robust Cloud Resource Provisioning for Cloud Computing Environments. In: IEEE International Conference on Service-Oriented Computing and Applications (SOCA), pp 1–8, DOI 10.1109/SOCA.2010.5707147
14. Chen J, Wang C, Zhou BB, Sun L, Lee YC, Zomaya AY (2011) Tradeoffs Between Profit and Customer Satisfaction for Service Provisioning in the Cloud. In: ACM 20th International Symposium on High Performance Distributed Computing, pp 229–238, DOI 10.1145/1996130.1996161
15. Chen LY, Birke R, Smirni E (2017) Self-Aware Computing Systems, chap State of Practice of Non-self-aware Virtual Machine Management in Cloud Data Centers, pp 555–574. DOI 10.1007/978-3-319-47474-8_19
16. Cima V, Grazioli B, Murphy S, Bohnert TM (2015) Adding Energy Efficiency to Openstack. In: Sustainable Internet and ICT for Sustainability (SustainIT), pp 1–8, DOI 10.1109/SustainIT.2015.7101358
17. Dejun J, Pierre G, Chi CH (2010) EC2 Performance Analysis for Resource Provisioning of Service-Oriented Applications. In: Service-Oriented Computing Workshops (ICSOC/ServiceWave), Lecture Notes in Computer Science, vol 6275, Springer Berlin Heidelberg, pp 197–207, DOI 10.1007/978-3-642-16132-2_19
18. Fifield T, Fleming D, Gentle A, Hochstein L, Hyde A, Proulx J, Toews E, Topjian J (2015) *OpenStack Operations Guide*. O'Reilly

19. Ghribi C, Hadji M, Zeglache D (2013) Energy Efficient VM Scheduling for Cloud Data Centers: Exact Allocation and Migration Algorithms. In: IEEE/ACM 13th International Symposium on Cluster, Cloud and Grid Computing (CCGrid), pp 671–678, DOI 10.1109/CCGrid.2013.89
20. Goh J, Sim M (2011) Robust Optimization Made Easy with ROME. *Operations Research* 59(4):973–985, DOI 10.1287/opre.1110.0944
21. Gorissen BL, Yanikoglu I, den Hertog D (2015) A practical guide to robust optimization. *Omega* 53:124–137, DOI <http://dx.doi.org/10.1016/j.omega.2014.12.006>
22. Hallawi H, Mehnen J, He H (2017) Multi-Capacity Combinatorial Ordering Genetic Algorithm in Application to Cloud resources allocation and efficient virtual machines consolidation. *Future Generation Computer Systems* 69:1–10, DOI <http://dx.doi.org/10.1016/j.future.2016.10.025>
23. I Hwang and M Pedram (2016) Hierarchical, Portfolio Theory-Based Virtual Machine Consolidation in a Compute Cloud. *IEEE Transactions on Services Computing* PP(99):1–14, DOI 10.1109/TSC.2016.2531672
24. IBM ILOG (2006) CPLEX Optimization Studio. Url: https://www.ibm.com/support/knowledgecenter/S5SA5P_12.6.3/ilog.odms.studio.help/pdf/usrcplex.pdf
25. Kim M, Ju Y, Chae J, Park M (2014) A Simple Model for Estimating Power Consumption of a Multicore Server System. *International Journal of Multimedia and Ubiquitous Engineering* 9(2):153–160
26. Li C, Li L (2013) Efficient Resource Allocation for Optimizing Objectives of Cloud Users, IaaS Provider and SaaS Provider in Cloud Environment. *The Journal of Supercomputing* 65(2):866–885, DOI 10.1007/s11227-013-0869-z
27. Li X, Qian Z, Lu S, Wu J (2013) Energy efficient virtual machine placement algorithm with balanced and improved resource utilization in a data center. *Mathematical and Computer Modelling* 58(5-6):1222 – 1235, DOI <http://dx.doi.org/10.1016/j.mcm.2013.02.003>
28. Li Z, Yan C, Yu X, Yu N (2017) Bayesian Network-based Virtual Machines Consolidation Method. *Future Generation Computer Systems* PP(69):75 – 87, DOI <http://dx.doi.org/10.1016/j.future.2016.12.008>
29. Marotta A, Avallone S (2015) A Simulated Annealing based Approach for Power Efficient Virtual Machines Consolidation. In: IEEE 8th International Conference on Cloud Computing (CLOUD), pp 445–452, DOI 10.1109/CLOUD.2015.66
30. McCullough JC, Agarwal Y, Chandrashekar J, Kuppuswamy S, Snoeren AC, Gupta RK (2011) Evaluating the Effectiveness of Model-based Power Characterization. In: Proceedings of the USENIX Annual Technical Conference, pp 12–12, DOI 2002181.2002193
31. Mishra M, Sahoo A (2011) On Theory of VM Placement: Anomalies in Existing Methodologies and Their Mitigation Using a Novel Vector Based Approach. In: IEEE International Conference on Cloud Computing (CLOUD), pp 275–282, DOI 10.1109/CLOUD.2011.38
32. Murtazaev A, Oh S (2011) Sercon: Server Consolidation Algorithm using Live Migration of Virtual Machines for Green Computing. *IETE Technical Review* 28(3):212–231, DOI 10.4103/0256-4602.81230
33. Nasim R, Kassler AJ (2015) Network-centric Performance Improvement for Live VM Migration. In: IEEE 8th International Conference on Cloud Computing, pp 106–113, DOI 10.1109/CLOUD.2015.24
34. Nguyen TH, Francesco MD, Yla-Jaaski A (2017) Virtual Machine Consolidation with Multiple Usage Prediction for Energy-Efficient Cloud Data Centers. *IEEE Transactions on Services Computing* PP(99):1–14, DOI 10.1109/TSC.2017.2648791
35. Ohta S (2015) Strict and Heuristic Optimization of Virtual Machine Placement and Migration. In: 9th International Conference on Computer Engineering and Applications, DOI 978-1-61804-276-7
36. Panda SK, Jana PK (2017) An Efficient Request-Based Virtual Machine Placement Algorithm for Cloud Computing. In: Krishnan P, Radha Krishna P, Parida L (eds) 13th International Conference on Distributed Computing and Internet Technology, pp 129–143, DOI 10.1007/978-3-319-50472-8_11
37. Park K, Pai VS (2006) CoMon: A Mostly-scalable Monitoring System for PlanetLab. *ACM SIGOPS Operating Systems Review* 40(1):65–74, DOI 10.1145/1113361.1113374
38. Poola D, Garg S, Buyya R, Yang Y, Ramamohanarao K (2014) Robust Scheduling of Scientific Workflows with Deadline and Budget Constraints in Clouds. In: IEEE 28th International Conference on Advanced Information Networking and Applications (AINA), pp 858–865, DOI 10.1109/AINA.2014.105
39. Reiss C, Tumanov A, Ganger GR, Katz RH, Kozuch MA (2012) Heterogeneity and Dynamicity of Clouds at Scale: Google Trace Analysis. In: ACM 3rd Symposium on Cloud Computing, SoCC '12, pp 7:1–7:13, DOI 10.1145/2391229.2391236
40. Richards A, How J (2005) Mixed-integer Programming for Control. In: IEEE Proceedings of the American Control Conference, pp 2676–2683, DOI 10.1109/ACC.2005.1470372
41. Setzer T, Wolke A (2012) Virtual Machine Re-assignment Considering Migration Overhead. In: IEEE Network Operations and Management Symposium (NOMS), pp 631–634, DOI 10.1109/NOMS.2012.6211973

42. Shen Z, Subbiah S, Gu X, Wilkes J (2011) Cloud-Scale: Elastic Resource Scaling for Multi-tenant Cloud Systems. In: ACM 2nd Proceedings of the Symposium on Cloud Computing, pp 1–14, DOI 10.1145/2038916.2038921
43. Takouna I, Dawoud W, Sachs K, Meinel C (2013) A Robust Optimization for Proactive Energy Management in Virtualized Data Centers. In: ACM/SPEC 4th International Conference on Performance Engineering, ICPE '13, pp 323–326, DOI 10.1145/2479871.2479917
44. Takouna I, Alzaghoul E, Meinel C (2014) Robust Virtual Machine Consolidation for Efficient Energy and Performance in Virtualized Data Centers. In: IEEE International Conference on Green Computing and Communications (GreenCom), pp 470–477, DOI 10.1109/iThings.2014.84
45. Tomás L, Tordsson J (2013) Improving Cloud Infrastructure Utilization Through Overbooking. In: ACM Cloud and Autonomic Computing Conference, pp 5:1–5:10, DOI 10.1145/2494621.2494627
46. Valdez-Vivas M, Apostolopoulos J, Bambos N (2014) Resource Management in Cloud Computing with Frictions and Congestion Weather. In: IEEE Global Communications Conference (GLOBECOM), pp 2344–2350, DOI 10.1109/GLOCOM.2014.7037158
47. Valdez-Vivas M, Bambos N, Apostolopoulos J (2014) Dynamic Resource Management in Virtualized Datacenters with Bursty Traffic. In: IEEE International Conference on Communications (ICC), pp 4287–4293, DOI 10.1109/ICC.2014.6883994
48. Vigliotti ADF, Batista DM (2014) A Green Network-aware VMs Placement Mechanism. In: IEEE Global Communications Conference, pp 2530–2535, DOI 10.1109/GLOCOM.2014.7037188
49. Waldspurger CA (2002) Memory Resource Management in VMware ESX Server. 5th Symposium on Operating Systems Design and Implementation 36(SI):181–194, DOI 10.1145/844128.844146
50. Wang X, Chen X, Yuen C, Wu W, Zhang M, Zhan C (2017) Delay-Cost Tradeoff for Virtual Machine Migration in Cloud Data centers. *Journal of Network and Computer Applications* 78:62–72, DOI <http://dx.doi.org/10.1016/j.jnca.2016.11.003>
51. Wolke A, Ziegler L (2014) Evaluating Dynamic Resource Allocation Strategies in Virtualized Data Centers. In: IEEE 7th International Conference on Cloud Computing (CLOUD), pp 328–335, DOI 10.1109/CLOUD.2014.52
52. Wu Q, Ishikawa F, Zhu Q, Xia Y (2016) Energy and Migration Cost-aware Dynamic Virtual Machine Consolidation in Heterogeneous Cloud Datacenters. *IEEE Transactions on Services Computing* PP(99):1–13, DOI 10.1109/TSC.2016.2616868
53. Xiao Z, Song W, Chen Q (2013) Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment. *IEEE Transactions on Parallel and Distributed Systems* 24(6):1107–1117, DOI 10.1109/TPDS.2012.283
54. Xu J, Fortes J (2010) Multi-Objective Virtual Machine Placement in Virtualized Data Center Environments. In: IEEE/ACM International Conference on Cyber, Physical and Social Computing (CPSCom), pp 179–188, DOI 10.1109/GreenCom-CPSCom.2010.137
55. Yanbo W (2008) Empirical Comparison of Robust, Data Driven and Stochastic Optimization. Master's thesis, MIT
56. Zhao J, Hu L, Ding Y, Xu G, Hu M (2014) A Heuristic Placement Selection of Live Virtual Machine Migration for Energy-Saving in Cloud Computing Environment. *PLoS ONE* 9(9):e108,275, DOI 10.1371/journal.pone.0108275
57. Zola E, Kassler AJ (2015) Energy Efficient Virtual Machine Consolidation under Uncertain Input Parameters for Green Data Centers. In: IEEE 7th International Conference on Cloud Computing Technology and Science (CloudCom), pp 436–439, DOI 10.1109/CloudCom.2015.15