Achieving Energy Efficiency in Datacenters by Virtual Machine Sizing, Replication, and Placement

Hadi Goudarzi[1] and Massoud Pedram[2]
[1] Qualcomm Incorporated, CA Email:*hgoudarz@usc.edu*
[2] University of Southern California, Department of Electrical Engineering, CA
Email:*pedram@usc.edu*

ABSTRACT – Monitoring and analyzing virtual machines (VMs) in a datacenter enables a cloud provider to place them on fewer physical machines with negligible performance penalty, a process which is known as the VM consolidation. Such a consolidation allows the cloud provider to take advantage of dissimilar workloads to reduce the number of ON servers, and thereby, reduce the datacenter energy consumption. Moreover, placing multiple copies of VMs on different servers and distributing the incoming requests among them can reduce the resource requirement for each copy and help the cloud provider do more aggressive consolidation. This chapter begins by a substantial review of various approaches for consolidation, resource management, and power control in datacenters. It continues by presenting a dynamic programming-based algorithm for creating multiple copies of a VM without degrading performance and doing VM consolidation for the purpose of datacenter energy minimization. A side benefit of the consolidation is to improve reliability of the services provided by the datacenter to its clients. Using the proposed algorithm, it is shown that more than 20% energy saving can be achieved compared to the previous work.

I. INTRODUCTION

Demand for computing power has been increasing due to the penetration of information technologies in our daily interactions with the world both at personal and public levels, encompassing business, commerce, education, manufacturing, and communication services. At personal level, the wide scale presence of online banking, e-commerce, SaaS (Software as a Service), social networking and so on produce workloads of great diversity and enormous scale. At the same time computing and information processing requirements of various public organizations and private corporations have also been increasing rapidly. Examples include digital services and functions required by the various industrial sectors, ranging from manufacturing to housing, from transportation to banking. Such a dramatic increase in the computing demand requires a scalable and dependable IT infrastructure comprising of servers, storage, network bandwidth, physical infrastructure, Electrical Grid, IT personnel and billions of dollars in capital expenditure and operational cost to name a few.

Datacenters are the backbone of today's IT infrastructure. The reach of datacenters spans a broad range of application areas from energy production and distribution, complex weather modeling and prediction, manufacturing, transportation, entertainment and even social networking. There is a critical need to continue to improve efficiency in all these sectors by accelerated use of computing technologies, which inevitably requires increasing the size and scope of datacenters. However, datacenters themselves are now faced with a major impediment of power consumption. Some reports such as [1] and [2] estimate the datacenter electricity demand in 2012 was around 31 GW globally which is equivalent to the electricity demand of around 23 million homes. These reports also predict fast growth rate for electrical energy consumption in datacenters. Power consumption of datacenters will soon match or exceed many other energy-intensive industries such as air transportation.

Apart from the total energy consumption, another critical component is the peak power; According to an EPA report [3], the peak load on the power grid from datacenters is estimated to be approximately 7 Gigawatts (GW) in 2006 in US, equivalent to the output of about 15 base-load power plants. This load is

increasing as shipments of high-end servers used in datacenters (e.g., blade servers) are increasing at a 20-30 percent CAGR.

System-wide power management is a huge challenge in datacenters. First, restrictions on availability of power and large power consumption of the IT equipment make the problem of datacenter power management a very difficult one to cope with. Second, the physical infrastructure (e.g., the power backup and distribution system and the computer room air conditioning, or CRAC for short, systems) tends to account for up to one third of total datacenter power and capital costs [4, 5, 6]. Third, the peak instantaneous power consumption must be controlled. The reason for capping power dissipation in the datacenters is the capacity limitation of the power delivery network (PDN) in the datacenter facility. Fourth, power budgets in datacenters exist in different granularities: datacenter, cluster, rack or even servers. A difficulty in the power capping is the distributed nature of power consumption in the datacenter. For example, if there is a power budget for a rack in the datacenter, the problem is how to allocate this budget to different servers and how to control this budget in a distributed fashion. Finally, another goal is to reduce the total power consumption. A big portion of the datacenter operational cost is the cost of electrical energy purchased from the utility companies. A trade-off exists between power consumption and performance of the system and the power manager should consider this trade-off carefully. For example, if the supply voltage level and clock frequency of a CPU are reduced, the average power consumption (and even energy needed to execute a given task) is reduced, but the total computation time is increased.

Low utilization of servers in a datacenter is one of the biggest factors in low power efficiency of the datacenter. The most important reason behind having the best energy efficiency at 100% load in servers is the energy non-proportional behavior of the servers [7]. This means that servers with idle status consume a big portion of their peak power consumption. The fact that most of the times, servers are utilized with between 10 to 50% of their peak load and discrete frequent idle times of servers [8] amplify this issue in the datacenters. This fact motivates the design of energy-proportional servers [4] to minimize the overall power consumption. However, due to the non-energy-proportional nature of the current servers, it is prudent from an energy efficiency viewpoint to have as few servers as possible turned on with each active server being highly utilized. In order to decrease the number of active servers, sharing a physical server between several applications is necessary. Virtualization technology creates this opportunity.

Virtualization technology creates an application-hosting environment that provides independence between applications that share a physical machine together [9]. Nowadays, computing systems rely heavily on this technology. Virtualization technology provides a new way to improve the power efficiency of the datacenters: consolidation. Consolidation means assigning more than one Virtual Machines (VM) to a physical server. As a result, some of the servers can be turned off and power consumption of the computing system decreases. Again the technique involves performance-power tradeoff. More precisely, if workloads are consolidated on servers, performance of the consolidated VMs (virtual machines) may decrease because of physical resource contention (CPU, memory, I/O bandwidth) but the power efficiency will improve because fewer servers will be used to service the VMs.

In order to determine the amount of the resources that needs to be allocated to each VM, some performance target needs to be defined for each VM. The IT infrastructure provided by the datacenter owners/operators must meet various Service Level Agreements (SLAs) established with the clients. The SLAs may be resource related (e.g., amount of computing power, memory/storage space, network bandwidth), performance related (e.g., service time or throughput), or even quality of service related (24-7 availability, data security, percentage of dropped requests). SLA constraints can be used to determine the limit (minimum and maximum) on the resource requirement of each VM to be able to satisfy the required performance target. On the other hand, in order to minimize the operational cost of the datacenter, energy cost also needs to be considered to decide about optimal resource assignment to VMs.

The scale of the resource management problem in datacenters is very big because a datacenter comprises of thousands to tens of thousands of server machines, working in tandem to provide services to hundreds of thousands clients at the same time, see for example reference [10] and [11]. In such a large computing system, energy efficiency can be maximized through system-wide resource allocation and VM consolidation. This is in spite of non-energy-proportional characteristics of current server machines [7].

Resource management solution affects the operational cost and admission control policy in the cloud computing system. Resource management in datacenter is usually handled by three types of resource

manager: resource arbiter, power manager and thermal managers. Resource arbiter or VM manager, decides about VM to server assignment and migration and resource allocation. Power manager controls the average and peak power in a distributed or centralized fashion in datacenters and thermal manager keeps the hardware temperature below certain critical point and minimizes the power consumption of the cooling system. In this chapter, a review of the most important work in the area of the resource arbiter and power manager is presented. Moreover, a novel approach to minimize the energy cost of datacenter by increasing the VM consolidation opportunity using VM replication is proposed.

Generating multiple copies of a VM and placing them on different servers is one of the basic ways to increase the service reliability. In this approach, only the original copy of the VM handles the requests and the other copies are idle. In this chapter, we propose to exploit all of these copies for servicing the requests. In this scenario, resource provided for each copy of the VM should satisfy SLA requirements and the set of distributed VMs should be able to service all of the incoming requests. For this reason, memory Band Width (bandwidth) provided for each copy of the VM should be the same as that of the original VM whereas the total CPU cycles provided for these VMs should be greater or equal to the provided CPU cycles for the original VM. Using this approach and an effective VM placement algorithm, which determines the number of VMs and place them on physical machines, the energy cost of the system can be reduced by 20%.

The proposed VM replication and placement algorithm is based on the dynamic programming and local search methods. The dynamic programming method determines the number of copies for each VM and places them on servers and the local search tries to minimize the energy cost by turning off the under-utilized servers.

The rest of this chapter is organized as follows. Related work is presented in the next section. The system model and problem formulation are presented in section III and IV. The proposed algorithm is presented in section V. The simulation results are presented in the section VI and the conclusions and future work directions are presented in the last section.

II. Related Work in Datacenter Power and Resource Management

A datacenter resource management system is comprised of three main components: resource arbiter, power manager, and temperature manager. An exemplary architecture for the datacenter resource management system with emphasis on the resource arbiter is depicted in Figure 1.

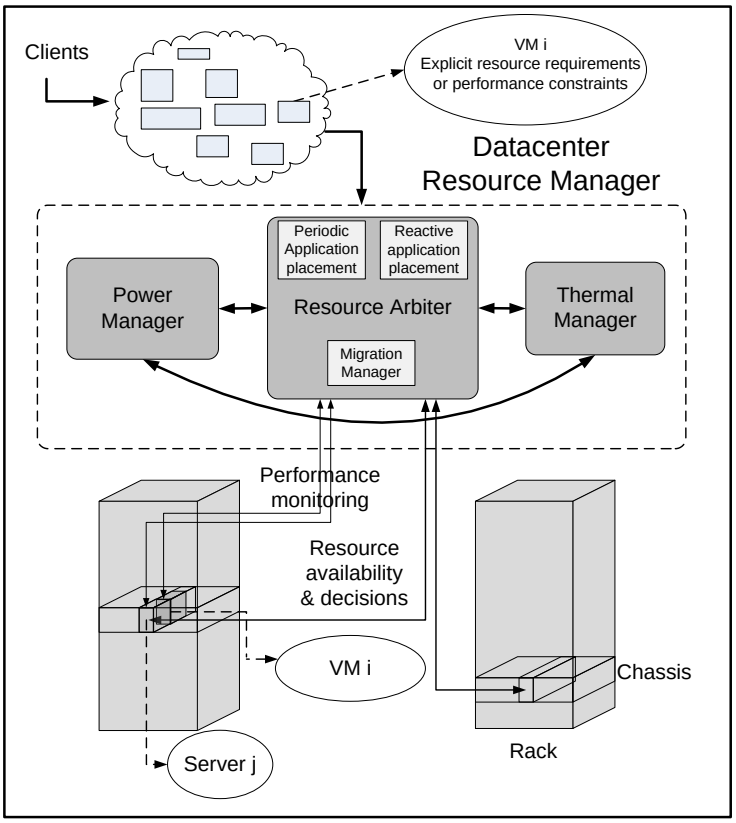


Figure 1. An example of resource management architecture in a datacenter.

In this architecture, the resource arbiter handles the application placement into resources and application migration. In this chapter, the term resource arbiter and resource manager are used interchangeably.

To assign applications or VMs to resources in a datacenter, one must monitor the resource availability and performance state of the physical servers in the datacenter. In addition, the resource arbiter must interact with the power and thermal managers. For example, if the power manager has limited the maximum power consumption of server, the resource arbiter should consider this limitation when assigning a new VM to the server. On the other hand, the power manager tries to minimize the average power consumption in each server considering the performance constraints of VMs assigned to servers as determined by the resource arbiter. Similarly, the resource arbiter must use the information provided by the thermal manager to decrease the workload of hot servers. At the same time, the thermal manager tries to control the temperature of active servers while accepting VM to server assignments made by the resource arbiter and meeting the per-VM performance constraints set forth the arbiter.

In this section, a review of the important approaches and techniques for design and optimization of resource arbiter and power managers in datacenters are presented. Thermal managers are out of the scope of this chapter. The review is by no means comprehensive, but aims to present some key approaches and results.

A. Resource Arbiter in Datacenters

Several versions of the resource management problem in datacenters have been investigated in the literature. Some of the prior work focuses on maximizing the number of served tasks in a datacenter (or total revenue for the datacenter operator) without considering the energy cost. Example references are [12] and [13], where the authors present heuristic solutions based on network flow optimization to find a revenue maximizing solution for a scenario in which the total resource requirement of tasks is more than the total resource capacity in the datacenter. The resource assignment problem for tasks with fixed memory, disc, and processing requirements is tackled in [14], where the authors describe an approximation algorithm for solving the problem of maximizing the number of tasks serviced in the datacenter.

Another version of the resource management problem is focused on minimizing the total electrical energy cost. Key considerations are to service all incoming tasks while satisfying specified performance guarantees for each task. A classic example of this approach is the work of Chase *et al.* in [15] who present a resource assignment solution in a hosting datacenter with the objective of minimizing the energy consumption while responding to power supply disruptions and/or thermal events. In this paper, economics-based approaches are used to manage the resource allocation in a system with shared resources in which clients bid for resources as a function of delivered performance.

Yet another version of the resource management problem considers the server and cooling power consumptions during the resource assignment problem. A representative of approaches to solving this problem is reference [16], in which Pakbaznia *et al.* present a solution for concurrent task assignment and VM consolidation in regular period called epochs. More precisely, workload prediction is used to determine the resource requirements (and hence the number of ON servers) for all incoming tasks for the epoch. Next considering the current datacenter temperature map and using an analytical model for predicting the future temperature map as a function of the server power dissipations, locations of the ON servers for the next epoch are determined and tasks are assigned to the ON servers so that the total datacenter power consumption is minimized.

Considering the effect of consolidation on the performance of servers is the key to reducing the total power consumption in a datacenter without creating unacceptable performance degradations. For example, Srikantaiah *et al.* [17] present an energy-aware resource assignment technique based on an experimental study of the performance, energy usage, and resource utilization of the servers while employing VM consolidation. In particular, two dimensions for server resources are considered in this paper: disk and CPU. Effects of the consolidation on performance degradation and energy consumption per transaction are quantified. The authors recommend applying consolidation so as not to over-commit servers in any resource dimension. The problem of application placement into a minimum number of ON servers, which is equivalent to the well-known bin-packing problem, is discussed, and a greedy algorithm for solving it is described.

Correlation of resource utilization patterns among VMs is an important factor when VM consolidation decision is being made [18]. Assigning highly (positive) correlated VMs in terms of resource usage, will increase the chance of VM migrations that is needed to avoid SLA violation. On the other hand, consolidating VMs that have less correlation in terms of their resource usage pattern results in more packed servers and lower power consumption [19]. Practical experiments of this theory are presented at reference [20] which suggests that the interferences between consolidated VMs in terms of (CPU/memory/networking) resource usage can cause the resource utilization to be lower or higher than the summation of the resource usage for VMs assigned to the server and needs to be considered to avoid performance degradation and SLA violations.

Resource usage of a VM in server can interfere with other VMs placed on that server. Moreover, VMs can have uneven resource utilization along different dimensions. These issues need to be considered in VM consolidation decisions. For example, the effect of uneven resource utilization along different dimensions (e.g., CPU, memory, and I/O) by different VMs and the question of how to improve datacenter energy efficiency by increasing the resource utilization in different resource dimensions are investigated by Xiao *et al.* in [21]. An approach to resolve the interference between VMs placed on the same physical machine is presented at [22].

A technique to maximize the utilization of the active server while creating more idle servers that can subsequently be turned off is to migrate VMs from a server with a low utilization factor to another server. A good example of considering server power consumption and VM migration cost in the resource assignment problem is reference [23], which presents power and migration cost-aware application placement in a virtualized datacenter. For this problem, each VM has fixed and known resource requirements based on the specified service level agreement (SLA). An elaborate architecture called pMapper and an effective VM placement algorithm to solve the assignment problem are key components of the proposed solution. More precisely, various actions in pMapper algorithm are classified as: (i) soft actions like VM re-sizing, (ii) hard actions such as Dynamic Voltage Frequency Scaling (DVFS), and (iii) VM consolidation actions. These actions are implemented by different parts of the implemented middleware. There is a resource arbiter, which has a global view of the applications and their SLAs and issues soft action commands. A power manager issues hard action commands whereas a migration

manager triggers consolidation decisions in coordination with a virtualization manager. These managers communicate with an arbitrator as the global decision maker to set the VM sizes and find a good application placement based on the inputs of different managers. Any revenue losses due to performance degradation caused by VM migration are calculated considering the given SLAs and used to set the migration costs of VMs. To optimally place VMs onto servers, the authors rely on a power efficiency metric to statically rank the servers independent of the applications running on them. This is because creating a dynamic ranking model for all mixes of all applications on all servers is infeasible. A heuristic based on the first-fit decreasing bin-packing algorithm [24] is presented to place the applications on servers starting with the most power-efficient server. Different versions of the first fit decreasing solution are proposed in a number of previous work including [25] and [26] to decide about VM assignment and consolidation.

The problem of resource allocation is more challenging in case of having clients with SLA contracts for a datacenter owner who wants to maximize its profit by reducing the SLA violations and decrease the operational cost [27]. Many researchers in different fields have addressed the problem of SLA-driven resource assignment. Some of the previous work has considered probabilistic SLA constraints with violation penalty, e.g., references [28] and [29]. Other work has relied on utility function-based SLA [30, 31, 26, 32]. In reference [33], a SLA with soft constraint on average response time is considered for multi-tier applications to solve the resource assignment problem. To determine and adjust the amount of resource allocated to VMs to satisfy SLA constraints, approaches based on reinforcement learning [34] and look-ahead control theory [35] have also been proposed. SLA contracts with guarantee on response time and/or penalties paid for violating the stipulated response time constraint is considered in references [36]. In this paper, a resource management system is presented that determines the amount of resource that needs be allocated to VMs based on SLA contracts and energy cost and subsequently assigns VMs to servers so as to reduce the operational cost of datacenter.

Due to big number of VMs and servers in a datacenter, an important factor in designing VM management solution is to make it as scalable as possible. Different works in the literature tackles this problem. Feller et al. [37] present a fully decentralized VM control solution to make the VM consolidation decisions. The proposed solution is based on peer-to-peer communication between physical servers to decide about assignment of new VM to a server and migration of VMs from an overloaded server. A decentralized VM assignment and migration is presented in [38] that targets to make the resource management solution scalable. The decision regarding accepting new VMs is decided by servers (based on a probabilistic approach) inside datacenter based on their current utilization. Hierarchical resource management solution is another way of decreasing the complexity of the resource management solution. A hierarchical resource allocation solution to minimize the server energy consumption and maximize a SLA-based utility function for datacenters is presented in [31]. The proposed hierarchical solution breaks the problem of resource scheduling to multiple smaller problems (smaller server and application sets) to reduce the complexity of the problem and increase the parallelism.

Modeling the performance and energy cost is vital for solving the resource assignment problem. Good examples of theoretical performance modeling are [39] and [40]. Benani et al. [39] present an analytical performance model based on queuing theory to calculate the response time of the clients based on CPU and I/O service times. Urgaonkar et al. [40] present an analytical model for multi-tier internet applications based on the mean-value analysis. An example of experimental modeling of power and performance in servers is presented in [41].

In order to satisfy SLA and be able to keep the guaranteed-level of performance for clients, datacenter resource manager needs to continuously monitor the performance of VMs and utilization level of the active servers in order to perform VM migration to avoid possible SLA violation [42] and [43]. Different approaches are suggested in the literature to decide about the maximum utilization point at which VM migration needs to happen. For example, authors in [44] and [45] suggest to use the statistical CPU utilization behavior of the consolidated VMs on a server in order to come up with a workload behavior-adaptive utilization limit that triggers the VM migration to avoid SLA violation. This adaptive limit makes the decision regarding VM migration more accurate compared to a fixed maximum utilization limit.

Statistical analysis of the resource utilization is also used in order to decide about the VM consolidation in datacenter resource managers. For example, network bandwidth of VMs are dynamic and

cannot be predicted perfectly. This fact motivated Wang et al. [46] to develop a solution to decide about VM consolidation based on the statistical data gathered from network bandwidth utilization of the VMs that can over-perform the VM consolidation solution based on the assumption of fixed communication bandwidth for each VM.

In order to be able to use the VM consolidation in its full extent, we need very fast VM migration solutions to avoid SLA violation in case of workload change. For example, Hirofuchi et al. [47] suggest using a fast solution called postcopy VM migration. In this approach, instead of migrating the whole memory before starting the VM operation in the destination host, VM operation starts right after the migration and before memory copy is finished. In this case, if VM needs to access a point in its memory before all the memory is copied to the destination host, VM operation is stalled for a short amount of time before copy of that point of memory is finished. This solution significantly reduces the VM stall time during the live migration.

B. Power Management in Datacenters

Power management is one of the key challenges in datacenters. The power issue is one of the most important considerations for almost every decision making process in a datacenter. In this context, the power issue refers to power distribution and delivery challenges in a datacenter, electrical energy cost due to average power consumption in the IT equipment and the room air conditioning, and power dissipation constraints due to thermal power budgets for VLSI chips.

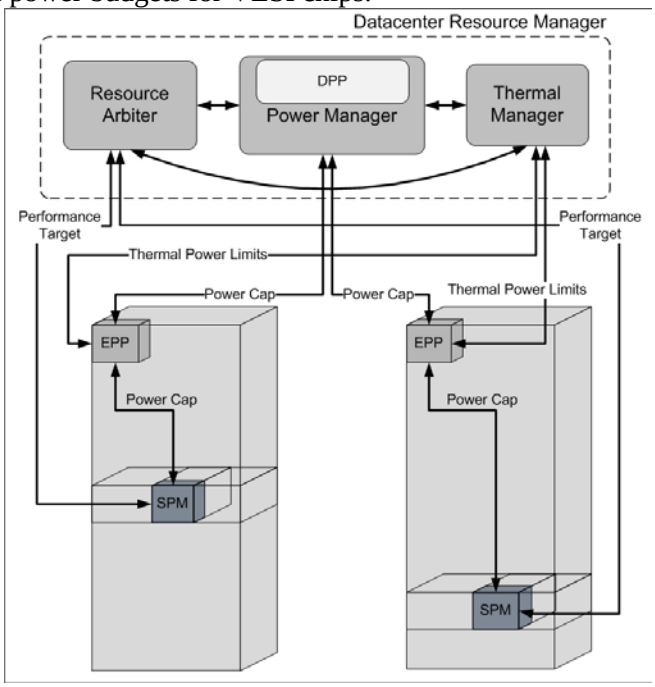


Figure 2. An example power management architecture and its relation to resource arbiter and thermal manager

Figure 2 depicts a distributed power management architecture composed of server-level power managers, plus blade enclosure and rack-level and datacenter-level power provisioners, denoted as SPMs, EPPs, and DPP, respectively. There is one SPM per server, one EPP per blade enclosure, and a single DPP for the whole datacenter. This architecture is similar to the four-layer architecture proposed in [48]. The only difference with the architecture proposed in [48] is that instead of using one server power manager for each server that minimizes the average power consumption and avoids power budget violation, two power manager is proposed to do these jobs.

A number of dynamic power provisioning policies have been presented in the literature, including [48], [49] and [50], where the authors propose using dynamic (as opposed to static) power provisioning to increase the performance in datacenter and decrease power consumption. Notice that the power provisioning problem can be formulated as deciding how many computing resources can be made active with a given total power budget for the datacenter.

Fan *et al.* [49] present the aggregate power usage characteristics of different units (servers, racks, clusters, and datacenter) in a datacenter for different applications over a long period of time. This data is analyzed in order to maximize the use of the deployed power capacity in the datacenter while reducing the risk of any power budget violations. In particular, this reference shows that there is a large difference between theoretical peak and actual peak power consumptions for different units. This difference grows as the unit size grows. This shows that the opportunity of minimizing the power budget under performance constraints (or maximizing the number of servers that are turned ON under a fixed power budget) increases as one goes higher in the datacenter hierarchy (e.g. from individual servers to datacenter as a whole.) For example, it is reported that in a real Google datacenter, the ratio of the theoretical peak power consumption to actual maximum power consumption is 1.05, 1.28 and 1.39 for rack, Power Distribution Unit (PDU) and cluster, respectively. The authors consider two approaches usually used for power and energy saving in datacenters, i.e., DVFS and reducing the idle power consumption in servers and enclosures (for example, by power gating logic and memory). Reported results suggest that employing the DVFS technique can result in 18% peak power reduction and 23% total energy reduction in a model datacenter. Moreover, decreasing the idle power consumption of the servers to 10% of their peak power can result in 30% peak power and 50% energy reduction. Based on these analyses and actual measurements, the authors present a dynamic power provisioning policy for datacenters to increase the possibility of better utilization of the available power while protecting the power distribution hierarchy against overdraws.

Exploring the best way of distributing a total power budget among different servers in a server farm in order to reach the highest performance level is studied in reference [51]. Moreover, an approach to reduce the peak power consumption of servers by dynamic power allocation using workload and performance feedbacks is presented in reference [52].

Design of an effective server-level power management is perhaps the most researched power management problem in the literature. Various Dynamic Power Management (DPM) techniques that solve versions of this problem have been presented by researchers. These DPM approaches can be broadly classified into three categories: ad hoc [53], stochastic [54], and learning based methods [55].

Server-level power manager can be quite effective in reducing the power consumption of datacenter. As an example, Elnozahy *et al.* [56] present independent as well as coordinated voltage and frequency scaling and turn on/off policies for servers in a datacenter and compare them against each other from a power savings perspective. Their results indicate that independent DVFS policies for individual servers results in 29% power reduction compared to a baseline system with no DVFS. In contrast, a policy that considers only turning on/off servers results in 42% lowering of the power consumption. The largest power saving of 60% is reported for a policy with coordinated DVFS and dynamic server ON/OFF decisions.

DPM techniques typically try to put the power consuming components to idle mode as often as possible to maximize the power saving. Studies on different datacenter workloads [7], [49] and [57] show frequent short idle times in workload. Because of the short widths of these idle times, components cannot be switched to their deep sleep modes (which consume approximately zero power) considering the expected performance penalty of frequent go-to-sleep and wakeup commands. At the same time, because of energy non-proportionality of current servers [7], idle server power modes give rise to relatively high power consumption compared to the sleep mode power consumption. As discussed at length before, VM consolidation is an answer to this problem. A new solution is however emerging. More precisely, a number of new architectures have been presented for hardware with very low (approximately zero) idle mode power consumption (energy-proportional servers) to be able to reduce the average power consumption in case of short idle times [49] and [4].

There are many examples of work that describe a combined solution for power and resource management solution. For example, Wang *et al.* [58] present a coordinated control solution that includes a cluster-level power control loop and a performance control loop for every VM. These control loops are configured to achieve desired power and performance objectives in the datacenter. Precisely, the cluster-level power controller monitors the power consumption of the servers and sets the DVFS state of the servers to reach the desired power consumption. In the same venue, the VM performance controller dynamically manages the VM performance by changing the resource (CPU) allocation policy. Finally, a cluster-level resource coordinator is introduced whose job is to migrate the VMs in case of performance

violation. As another example, Beloglazov and Buyya [59] propose a management architecture comprising of a VM dispatcher, as well as local and global managers. A local manager migrates a VM from one server to another in case of SLA violations, low server utilization, high server temperature, or high amount of communication with another VM in a different server. A global manager receives information from local managers and issues commands for turning on/off servers, applying DVFS or resizing VMs.

This chapter tackles the resource management problem in a cloud computing system. Key features of our formulation and proposed solution are that we consider heterogeneous servers in the system and use a two dimensional model of the resource usage accounting for both computational and memory bandwidth. We propose multiple copies of VMs to be active in each time in order to reduce the resource requirement for each copy of the VM and hence help increase the chances for VM consolidation. Finally an algorithm based on dynamic programming and local search is described. This algorithm determines the number of copies of each VM and the placement of these copies on servers so as to minimize some total system cost function.

III. System Model

In this section, detail of the assumptions and system configuration for the VM placement problem are presented. To improve the readability, Table I presents key symbols and definitions used in this chapter. Note that each client is identified by a unique id, denoted by index *i* whereas each server in the cloud computing system is identified by a unique id, denoted by index *j*.

Table I. Notation and Definitions

Symbol name	***Definition***
c_i^m and c_i^p	Required memory bandwidth and total processing demand of the i[th] client
L_i	Max. number of servers allowed to serve the i[th] client
s_k	Set of servers of type k
C_j^p and C_j^m	Total CPU cycle capacity and memory bandwidth of the j[th] server
P_j^0	Constant power consumption of the j[th] server in the active mode
P_j^{p}	Power of operating the j[th] server which is proportional to the utilization of processing resources
T_e	Duration of a epoch in seconds
x_j	A pseudo-Boolean variable to determine if the j[th] server is ON (1) or OFF (0)
y_{ij}	A pseudo-Boolean variable to determine if the i[th] VM is assigned to the j[th] server (1) or not (0)
ϕ_{ij}^p, ϕ_{ij}^m	Portion of the processing and memory bandwidth resources of the j[th] server that is allocated to the i[th] client
ϕ_j^p, ϕ_j^m	Portion of the processing and memory bandwidth resources of the j[th] server that is allocated to any clients
α	Processing size ratio of the VM copy (between $1/L_i$ and 1) that determines the portion of the original VM CPU cycle provided by the VM copy
$f(\alpha)$	Function of processing size ratio that is used in calculating ϕ_{ij}^p based on c_i^p amd C_j^p
$c_{ij}(\alpha)$	Estimate of the energy cost of assigning a copy of the i[th] VM with processing size ratio of α to the j[th] server
y_{ij}^{α}	assignment parameter for j[th] server with VM with processing size ratio of α

A. Cloud Computing System

In the following paragraphs, we describe the type of the datacenter that we have assumed as well as our observations and key assumptions about where the performance bottlenecks are in the system and how we can account for the energy cost associated with a client's VM running in the datacenter.

A datacenter comprises of a number of potentially heterogeneous servers chosen from a set of known and well-characterized server types. In particular, servers of a given type are modeled by their processing capacity or CPU cycles (C_*^p) and memory bandwidth (C_*^m) as well as their operational expense (energy

cost), which is directly related to their average power consumption. We assume that local (or networked) secondary storage (disc) is not a system bottleneck.

The main part of the operational cost of the system is the total energy cost of serving clients' requests. The energy cost is calculated as the server power multiplied by the duration of each epoch in seconds (T_e). The power dissipation of a server is modeled as a constant power cost (P_*^0) plus another variable power cost, which is linearly related to the utilization of the server (with slope of P_*^p). This model is inspired by the previous works such as [41] and [17]. Note that the power cost of communication resources and air conditioning units are amortized over all servers and communication/networking gear in the datacenter, and are thus assumed to be relatively independent of the clients' workloads. More precisely, these costs are not included in the equation for power cost of the datacenter.

B. Client and Virtual Machines

Clients in the cloud computing system are represented as VMs. Based on the SLA contract or using workload prediction with consideration of the SLA, the amount of resources required for each client can be determined. These VMs are thus considered to have processing and memory bandwidth requests during the considered epoch. This assumption is applicable to online services (not for batch applications).

Each client's VM may be copied on different servers (i.e., requests generated by a single VM can be assigned to more than one server). This request distribution can decrease the quality of the service if the number of servers that process the client requests is large [30]. Therefore, we impose an upper bound on this number; precisely, L_i determines the maximum number of copies of any VM in the datacenter (this bound can be set to one if for some reason the VM should not be replicated). When multiple copies of a VM are active on different servers, the following constraints must be satisfied:

$$\sum_j \phi_{ij}^p C_j^p \geq c_i^p \quad (1)$$

$$\phi_{ij}^m C_j^m = y_{ij} c_i^m \quad (2)$$

where ϕ_{ij}^p and ϕ_{ij}^m denote the portion of the j[th] server CPU cycles and memory bandwidth allocated to the VM associated with client *i*. Constraint (1) enforces the summation of the reserved CPU cycles on the assigned servers to be equal or greater than the required CPU cycles for client *i*. Constraint (2) enforces the provided memory bandwidth on assigned servers to be equal to the required memory bandwidth for the VM. This constraint enforces the cloud provider not to sacrifice the Quality of Service (QoS) of clients. An example of VM1 being replicated as VM2 and VM3 is shown in Figure 3.

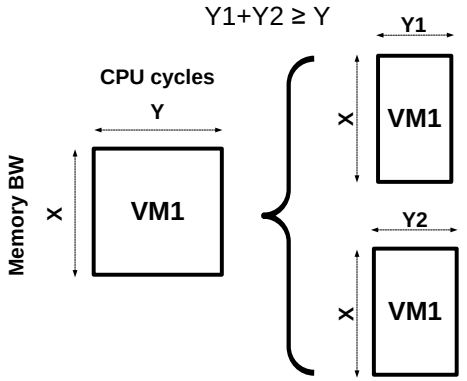


Figure 3. An example of multiple copies of a VM

C. VM Management System

The focus of the rest of this chapter is VM manager, which is responsible for determining resource requirements of the VMs and placing them on servers. Moreover, to address dynamic workload changes, VM manager may do VM migration. VM manager performs these tasks utilizing two different optimization procedures: semi-static optimization and dynamic optimization. The semi-static optimization procedure is performed periodically, whereas the dynamic optimization procedure is performed whenever it is needed.

In the semi-static optimization procedure, VM manager considers the full active set of VMs, the previous assignment solution, feedbacks generated by the power, thermal, and performance sensors, and

workload prediction in order to generate the best VM placement solution for the next epoch. The period for performing semi-static optimization depends on the type and size of the datacenter and workload characteristics. In the dynamic optimization procedure, VM manager finds a temporary VM placement solution by migrating, creating, or removing some VMs in respond to any performance, power budget, or critical temperature violations.

In this work, we focus on semi-static optimization procedure of VM manager. In this procedure, resource requirements of VMs are assumed to be determined based on SLA contracts and workload estimation for the next epoch. The duration of the epoch is long enough for one to neglect the VM migration delay penalty (it is typically less than 100ms for live migration [23]) with respect to the gain of the global optimization. Consequently, the energy cost optimization may be performed without the constraint of the state of the cloud computing system in the previous epoch.

The role of semi-static optimization procedure in VM manager is to answer the questions of (i) whether to create multiple copies of VMs on different servers or not and (ii) where to place these VMs. Considering fixed payments from clients for the cloud service, the goal of this optimization is to minimize the operational cost of the active servers in datacenter. An exemplary solution for assigning six VMs on two heterogeneous servers is shown in Figure 4.

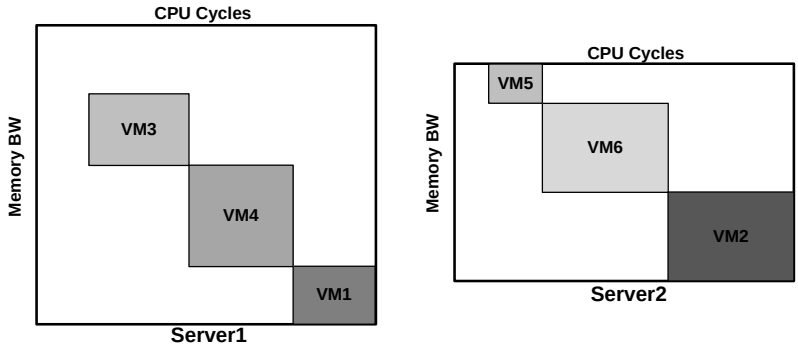


Figure 4. An exemplary solution for assigning six VMs on two different servers

IV. PROBLEM FORMULATION

In this chapter, a VM placement problem is considered with the objective of minimizing the total energy consumption in the next epoch while servicing all VMs in the cloud computing system.

The exact formulation of the aforesaid problem (called EMRA for Energy-efficient Multi-dimensional Resource Allocation) is provided below (cf. Table I.)

$$Min \quad T_e \sum_j x_j \left(P_j^0 + P_j^p \sum_i \phi_{ij}^p \right) \tag{3}$$

subject to:

$$\phi_j^p = \sum_i \phi_{ij}^p \leq 1 \qquad \forall j \tag{4}$$

$$\phi_j^m = \sum_i \phi_{ij}^m \leq 1 \qquad \forall j \tag{5}$$

$$\sum_j C_j^p \phi_{ij}^p \geq c_i^p \qquad \forall i,j \tag{6}$$

$$y_{ij} \geq \phi_{ij}^p, \qquad \forall i,j \tag{7}$$

$$\phi_{ij}^m C_j^m = y_{ij} c_i^m \qquad \forall i,j \tag{8}$$

$$\sum_i y_{ij} \leq L_i \qquad \forall i \tag{9}$$

$$x_j \geq \sum_i \phi_{ij}^p \qquad \forall j \tag{10}$$

$$y_{ij} \in \{0,1\}, x_j \in \{0,1\}, \phi_{ij}^p \geq 0, \phi_{ij}^m \geq 0 \qquad \forall i,j \tag{11}$$

where x_j is a pseudo-Boolean integer variable to determine if the j[th] server is ON (x_j=1) or OFF (x_j=0).

The objective function is the summation of the operation costs (energy dissipations) of the ON servers based on a fixed power factor and a variable power term linearly related to the server utilization. In this problem, x_j, y_{ij} and ϕ_{ij}^{p} denote the optimization variables.

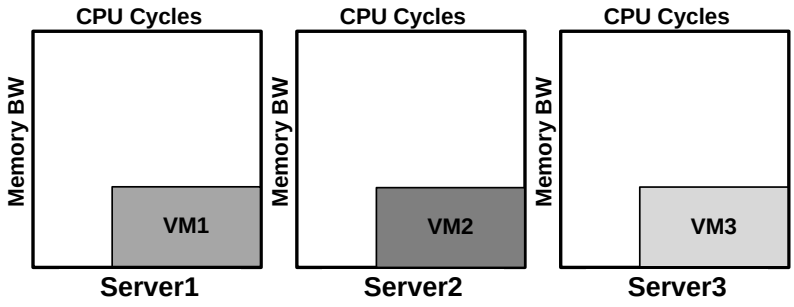
The constraints capture the limits on the number of available servers and clients. In particular, inequality constraints (4) and (5) represent the limit on the utilization of the processing and memory bandwidth in the j[th] server, respectively. Constraint (6) ensures that required processing demands for each VM is provided. Constraint (7) generates a pseudo-Boolean parameter that determines if a copy of a VM is assigned to a server ($y_{ij} = 1$) or not ($y_{ij} = 0$). Constraint (8) ensures the memory bandwidth needs of a VM that is assigned to a server are met whereas constraint (9) ensures that the number of copies of a VM does not exceed the maximum possible number of copies. Constraint (10) generates the pseudo-Boolean parameter related to the status of each server. Constraint (11) specifies the domains of optimization variables.

Theorem I: Generalized Assignment Problem (GAP) [24] can be reduced to EMRA problem.

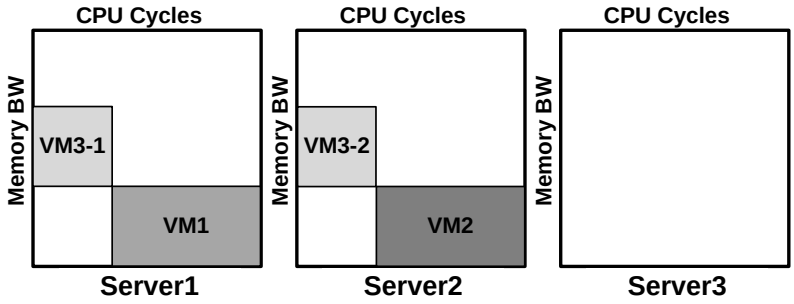
Proof: Consider a version of the EMRA problem in which P_j^0 is equal to zero for every server and L_i is equal to 1 for every VM. In this problem, assigning each VM (exactly one copy) to each server has different costs, and each server has two dimensional resources that can be assigned to VMs. So, we can solve any two-dimensional GAP problem by using the solution to EMRA problem for a special case. ■

Considering theorem I, the EMRA problem is NP-hard [24]. Indeed, similar to the GAP problem, even the question of deciding whether a feasible solution exists for this problem does not admit an efficient solution [24]. In this chapter, we consider a case in which the required resources for VMs are smaller than the available resources in the datacenter. This means that we consider energy minimization with a fixed set of VMs instead of maximizing the number of (or the profit for) a subset of VMs served in the datacenter. Therefore, a simple greedy algorithm (similar to First Fit Decreasing (FFD) heuristic [24]) will find a feasible solution to the EMRA problem. Another important observation about this problem is that the number of clients and servers in this problem are very large; therefore, a critical property of any proposed heuristic is its scalability.

An example of how multiple copies of VM can reduce energy consumption of the cloud system is seen when we compare Figure 5-a and Figure 5-b. Here, three homogenous VMs are assigned to three homogenous servers. The CPU cycle capacity of each server is strictly less than 2X the required CPU cycle count of each VM (say, 1.75 times) whereas the memory bandwidth capacity of each server is strictly more than 2X the required memory bandwidth of each VM (say, 3 times). In Figure 5, you can see that the assignment results in three active servers. If we consider VM replication, we can create two copies of the third VM with the same memory bandwidth requirements but smaller CPU cycle demands. Assigning the new set of VMs to servers can result in only two active servers with high CPU and memory bandwidth utilization, which may result in energy saving due to the energy non-proportionality behavior of the servers.



(a) Without VM replication



(b) With VM replication

Figure 5. An exemplary solution for assigning three VMs on three identical servers

V. PROPOSED ALGORITHM

In this section, a two-step heuristic for solving the EMRA problem is presented. In the first step, an algorithm based on Dynamic Programming (DP) is used to determine the number of copies for each VM and the assignment of these VMs to the servers. This decision determines (i) which servers will be active during the next epoch and (ii) the utilization of the active servers in that epoch. The goal of the algorithm is to minimize the total energy cost. In the second step, a local search is conducted to further reduce the power consumption by turning off some of the active servers and placing their VMs on other active servers.

In the beginning of the VM placement, clients are ordered in descending order of their CPU cycle demands. Based on this ordering, the optimal number of copies of the VMs are determined and they are placed on servers by using dynamic programming. In the local search method, servers are turned off based on their utilization and VMs assigned to them are moved to the rest of the active servers so as to minimize the energy consumption as much as possible.

Details of the Energy-efficient VM Placement algorithm (EVMP) are presented below.

A. Energy Efficient VM Placement Algorithm – Initial solution

Initially, the values of ϕ_j^p and ϕ_j^m for each server are set to zero. A constructive approach is used to place the VMs on the servers. VMs are sorted based on their processing requirements in a descending order. For each VM, a method based on DP is used to determine the number of copies that are placed on different servers.

To estimate the power consumption of assigning a copy of the i^{th} VM to the j^{th} server of type k, we use the following equation.

$$c_{ij}(\alpha) = \begin{cases} T_e(\phi_{ij}^p P_j^p + P_j^0 c_i^m / C_j^m) & If\ server\ is\ active \\ T_e(\phi_{ij}^p P_j^p + P_j^0) & otherwise \end{cases} \quad (12)$$

where α (between $1/L_i$ and 1) is the processing size ratio of the VM copy to that of the original VM. In other words α denotes the percentage of the original VM CPU cycles to be provided to the copy of VM.

The top branch of equation (12) estimates the energy cost of assigning a copy of the i^{th} VM to an already active server and is comprised of a utilization-proportional power consumption of the server plus

a fraction of the idle power consumption based on the (normalized) required memory bandwidth of the assigned VM. Similarly, the bottom branch of equation (12) estimates the energy cost of assigning a copy of the i^{th} VM to a currently inactive server (but to become active soon). This energy cost estimate includes the utilization-proportional power consumption of the server and accounts for the whole idle power consumption of the server. The additional power consumption in the bottom branch compared to the one in the top branch captures the risk of turning on a server if no other VM is assigned to that server for the next epoch. The average utilization of the server type ($\bar{\phi}_j^p$ and $\bar{\phi}_j^m$) in the previous epochs can also be used to replace the equation in the bottom branch with $T_e(\phi_{ij}^p P_j^p + P_j^0 \min(1, \max(c_i^m/\bar{\phi}_j^m, c_i^m/\bar{\phi}_j^m)))$ to more accurately account for the energy cost risk of turning on a server for VM assignment.

ϕ_{ij}^p is a function of the VM, the server, and α. It can be calculated as shown below.

$$\phi_{ij}^p = f(\alpha) c_i^p / C_j^p \quad (13)$$

where $f(\alpha)$ is a function of the processing size ratio of the VM. We know that in any type of VM and servers, $f(0)$ is equal to 0 while $f(1)$ is equal to 1. f is a monotonically increasing function. Considering the beginning and endpoint of this function at 0 and 1 and considering constraint (1), for any value between 0 and 1, the value of function f can be between α and 1. For example, if half of the CPU cycle requirement of the VM is provided by a copy of the VM, $\phi_{ij}^p = f(1/2) c_i^p / C_j^p$ which is greater than or equal to $0.5 c_i^p / C_j^p$. If this property does not hold for a small portion of the spectrum, we can create a solution with multiple VM copies which require less than c_i^p resources collectively and violate the constraint (1).

The presented algorithm is based on a general function f with the mentioned behavior but an example of this function based on a performance model is presented in subsection B.

For each VM, both versions of equation (12) are calculated for each server type and different values of α (between $1/L_i$ and 1 with steps of $1/L_i$). Moreover, for each server type, L_i active servers and L_i inactive servers that can service at least the smallest copy of the VM are selected as candidate hosts. For assigning the VM to any of the candidate servers, the cost is determined by the top or bottom branch of equation (12) as the case may be.

After selecting active and inactive candidate servers for each server type and calculating cost for each possible assignment, the problem is reduced to (14).

$$Min \ \sum_{j \in P} y_{ij}^{\alpha} c_{ij}(\alpha) \quad (14)$$

subject to:

$$\sum_{j \in P} \alpha y_{ij}^{\alpha} = L_i \quad (15)$$

where y_{ij}^{α} denotes the assignment parameter for j^{th} server for a VM copy with processing size ratio of α (1 if assigned and 0 otherwise). Moreover, P denotes the set of candidate servers for this assignment.

The DP method is used to solve this problem and find the best assignment decision. In this DP method, candidate servers can be processed in any order. This method examines all the possible VM placement solution efficiently without calculating every possible solution in a brute-force manner. Using this method, the optimal solution for problem presented in (14) can be found.

Algorithm 1 shows the pseudo code for this assignment solution for each VM. Complexity of this DP solution is $O(2L_i^2 K)$, where K denotes the number of server types that are considered for this assignment. The complexity is calculated from the number of cost calculation in line 23 of the pseudo code. After finding the assignment solution ϕ_j^p and ϕ_j^m for the selected servers are updated. Then, the next VM is chosen and this procedure is repeated until all VMs are placed.

Algorithm 1: Energy Efficient VM Placement

Inputs: C_j^m, C_j^p, P_j^0, P_j^p, c_i^m, c_i^p, L_i

Outputs: ϕ_{ij}^p, ϕ_{ij}^m (i is constant in this algorithm)

$P = \{\}$
For (k = 1 to number of server types)
  ON=0; OFF=0;
  For ($\alpha = 1/L_i$ to L_i)
    $\phi_{ij}^p = f(\alpha) c_i^p / C_j^p$
    $c_{ij}^{active}(\alpha) = \phi_{ij}^p P_j^p + P_j^0 c_i^m / C_j^m$
    $c_{ij}^{inactive}(\alpha) = \phi_{ij}^p P_j^p + P_j^0$
  End
  $J^{ON} = \{j \in s_k | (1 - \phi_j^m) \geq c_i^m / C_j^m \ \& \ (1 - \phi_j^p) \geq c_i^p / L_i C_j^p\}$
  $J^{OFF} = \{j \in s_k | \phi_j^p = 0, (1 - \phi_j^m) \geq c_i^m / C_j^m\}$
  Foreach ($j \in s_k$)
    If ($j \in J^{ON}$ & $ON < L_i$)
      $P = P \cup \{j\}$, $ON{+}{+}$, $c_{ij}(\alpha) = c_{ij}^{active}(\alpha)$
    Else if (j $\in$ J^{OFF} & OFF< L_i)
      $P = P \cup \{j\}$, $OFF{+}{+}$, $c_{ij}(\alpha) = c_{ij}^{inactive}(\alpha)$
  End
End
$X = L_i$, and $Y = size\ (P)$
Foreach ($j \in P$)
  For ($x = 1$ to X)
    $D[x, y]$ = infinity; //Auxiliary $X \times Y$ matrix used for DP
    For ($z = 1$ to x)
      $D[x, y] = min(D[x, y], D[x - 1, y - z] + c_{ij}(z))$
    $D[x, y] = min(D[x, y], D[x - 1, y])$
  End
End
Back-track to find best ϕ_{ij}'s to minimize cost and update ϕ_j's

B. Example of function $f(\alpha)$ for a performance model

To better appreciate the concept of function f(α), a performance model for VM is briefly presented.

To model the response time of the VMs, we assume that the inter-arrival times of the requests for each VM follow an exponential distribution function similar to the inter-arrival times of the requests in the e-commerce applications [28]. The average inter-arrival time (λ_i) of the requests for each VM can be estimated from analyzing workload traces [60].

In case of more than one copy of a VM, requests are assigned probabilistically i.e., α portion of the incoming requests are forwarded to the j^{th} server (i.e., the host for some copy of the VM) for execution, independently of the past or future forwarding decisions. Based on this assumption, the request arrival rate in each server follows the Poisson distribution function.

An exponential distribution function can be used to model the service time of the clients in this system. Based on this model, the response time distribution of a VM (placed on server *j*) is an exponential distribution with the following expected value:

$$\bar{R}_{ij} = \frac{1}{C_j^p \phi_{ij}^p \mu_{ij} - \alpha \lambda_i} \tag{16}$$

where μ_{ij} denotes the service rate of the i^{th} client on the j^{th} server when a unit of processing capacity is allocated to the VM of this client.

Most response-time sensitive applications have a contract with the cloud provider to guarantee that the response time of their requests does not go over a certain threshold. The constraint on the response time of the i^{th} client may be expressed as:

$$Prob\{R_i > R_i^c\} \leq h_i^c \tag{17}$$

where R_i and R_i^c denote the actual and target response times for the i^{th} client's requests, respectively.

Based on the presented model and constraint (17), the response time constraint for each copy of a VM can be expressed as follows:

$$e^{-\left(C_j^p \phi_{ij}^p \mu_{ij} - \alpha\lambda_i\right)R_i^c} \leq h_i^c \Rightarrow \phi_{ij}^p \geq (\alpha\lambda_i - \ln h_i^c / R_i^c)/\mu_{ij} C_j^p \quad (18)$$

If there is only one copy of VM, c_i^p can be calculated as follows:

$$c_i^p = \lambda_i/\mu_{ij} C_j^p + (-\ln h_i^c / R_i^c)/\mu_{ij} C_j^p \quad (19)$$

Considering the presented performance model, c_i^p varies based on the server type. If the processing size ratio of α is considered for the VM copy, lower bound of ϕ_{ij}^p has a similar formula as (19) with the first term multiplied by α. The first term of ϕ_{ij}^p is the portion that scales with the processing size ratio of the VM. The second term of ϕ_{ij}^p is a constant value based on the SLA contract parameters, service rate, and processing capacity of the server. Note that the second term does not scale with α and exists in even the smallest VM copy to guarantee that the request is serviced with an acceptable response time.

Having multiple copies of VM requires to account for the second term multiple times. For example, if there are three active copies of a VM, independent of the value for α parameter for each copy, the summation of ϕ_{ij}^p is equal to $\lambda_i/\mu_{ij} C_j^p + 3 * (-\ln h_i^c / R_i^c)/\mu_{ij} C_j^p$. This value would be larger than the c_i^p. The function f for this performance model is presented below.

$$f(\alpha) = \frac{\alpha\,\lambda_i/\mu_{ij} C_j^p + (-\ln h_i^c / R_i^c)/\mu_{ij} C_j^p}{\lambda_i/\mu_{ij} C_j^p + \left(-\ln h_i^c / R_i^c\right)/\mu_{ij} C_j^p} = \alpha + \frac{(1-\alpha)\,(-\ln h_i^c / R_i^c)/\mu_{ij} C_j^p}{\lambda_i/\mu_{ij} C_j^p + \left(-\ln h_i^c / R_i^c\right)/\mu_{ij} C_j^p} \quad (20)$$

The behavior of f is determined from the ratio between the first and second terms in equation (19). When this ratio is big, creating a limited number of copies from that VM is reasonable since $f(\alpha) \cong \alpha$ and the total amount of processing power reserved and used for multiple copies of VM is approximately equal to the processing power needed for the original VM. On the other hand, when the aforesaid ratio is small, then the VM is not a good candidate for replication since the total processing power required for multiple copies of that VM is multiple times larger than the required processing power for the original VM. However, as shown before, in some scenarios the increase in utilization of servers and turning off some other servers by creating multiple copies of VM can decrease the overall operational cost of the datacenter and cloud system. The proposed algorithm can capture this trade-off and come up with the near optimal solution.

C. Energy Efficient VM Placement Algorithm – Local Search

The constructive nature of the proposed algorithm can cause a situation in which some servers are not well utilized. However the large number of clients makes this problem less severe. To improve the results of the proposed VM placement algorithm, a local search method is used.

In order to select the candidate servers for turning OFF, utilization of the server needs to be defined. Due to heterogeneity of the server resources and VM resource requirements, it is possible that the utilization ratio of the server along different resource dimensions will be different. Since saturation of each resource type in the server results in a resource-saturated server (to be called a fully-utilized server), we define the utilization of a server as the maximum resource utilization along different resource dimensions. For example if $\phi_j^p = 0.5$ and $\phi_j^m = 0.3$, we consider the utilization of the server to be 50%. To minimize the total energy consumption in the system, all servers with utilization less than a threshold will be examined in this local search. This threshold can be specified by the cloud provider.

To examine these under-utilized servers, each of them is turned off one by one (starting from servers with lowest utilization) and total energy consumption is found by placing their VMs on other active servers using the proposed DP placement method. If the total cost of the new placement is less than the previous total cost, the new configuration is fixed and the rest of under-utilized servers are examined, otherwise the option of turning off that server is removed and the other candidate servers are examined. Algorithm 2 shows a high-level pseudo code for the proposed local search step.

Algorithm 2: Local Search Algorithm

Inputs: Current VM assignment and x_j

Outputs: New VM assignment and x_j

$\phi_j = max(\phi_j^m, \phi_j^p)$
$J = \{j | \phi_j > 0\}$
While (ϕ_j <threshold OR timeout)
    $j = \text{argmin}_{j \in J | \phi_j > 0} \phi_j$
    $I = \{i | \phi_{ij}^m > 0\}$
    $OPEX_{old}$ =Total operational cost based on the current assignment
    Foreach ($i \in I$)
        Find a new placement on set of active servers
    End
    $OPEX_{new}$ =Total operational cost based on the new assignment
    If ($OPEX_{new} < OPEX_{old}$)
        $x_j = 0$ and fix the new VM assignment
    Else
        $x_j = 1$ and keep the old VM assignment
    $J = J - j$
End
Finalize the set of active servers and VM assignment for the current epoch

VI. SIMULATION RESULTS

To evaluate the effectiveness of the proposed VM placement algorithm, a simulation framework is implemented. Simulation setups, baseline heuristics and numerical results of this implementation are presented in this section.

A. Simulation Setup

For simulation purposes, model parameters are generated from real world examples. The number of server types is set to 8. For each server type, some arbitrary number of servers are provisioned in datacenter. Processors for each server type are selected from the Intel portfolio of processors (e.g. Atom, i5, i7 and Xeon) [61] with different number of cores, cache sizes, power consumptions, and clock frequencies. Peak power consumptions for different servers (excluding the processor itself) are set uniformly to be between two to four times the power consumption of the corresponding fully-utilized processor. The memory bandwidth requirements of the servers are selected based on the maximum memory bandwidth of these processors multiplied by a factor of 0.4. For example if the maximum memory bandwidth of a processor is 20 GB/s, the available memory bandwidth for this processor is set to 8 GB/s.

The processing (CPU cycle count) requirement for each VM is selected uniformly between 1 and 18 billion CPU cycles per second. In order to observe the effect of function $f(\alpha)$, we ran the experimental results twice for each setting. The first time considering $f(\alpha) = \alpha$ and the second time with $f(\alpha) = (\alpha + 1)/2$. f_1 and f_2 denote the first and second observed values of $f(\alpha)$. As described in subsection V.B, $f(\alpha)$ is a function of the type of VM and is not constant for all VMs in datacenter. The purpose of considering two different $f(\alpha)$ for the simulation setup is to show how the algorithm works with different VM replication costs.

The memory bandwidth requirements for clients are also selected uniformly between 768MB/s and 4GB/s. The selection of processing resource requirement is based on the fact that the base-line algorithms do not automatically support multiple copies of VMs. This means that the required processing capacity of each VM should be less than the maximum available processing capacity in the datacenter; otherwise, the base-line algorithms cannot handle the VM placement problem. On the other hand, EVMP algorithm is capable of generating a VM placement solution if the memory bandwidth requirement of each VM is less than the maximum memory bandwidth supported by the available servers in the datacenter.

Upper bound on the number of copies for each VM is set between 1 and 5 based on the value of the required processing resources, e.g. if the processing requirement for a VM is equal to maximum processing requirements, L_i is set to 5 and if the value of processing requirement for a VM is less than ¼ of the maximum value, L_i is set to one (no copy is allowed).

Each simulation is repeated at least 1,000 times to generate acceptable average results for each case.

B. Heuristics for Comparison

We implemented the *min Power Parity* (mPP) heuristic [23] as one of the state-of-the-art energy-aware VM placement techniques. This heuristic is based on first fit decreasing heuristic [24] for the bin packing problem. This heuristic tries to minimize the overall power consumed by active servers in the datacenter. mPP heuristic works in two steps. In the first step, a target utilization for each server is found based on the power model for the servers. The target utilization of the servers is found by minimizing the power consumption of assigning the total required CPU utilization of all VMs on the current servers. In the second step, FFD heuristic is used to assign VMs to the selected set of the active servers. More details of mPP can be found in [23].

To show the effectiveness of our proposed approach for placing multiple copies of VMs on servers, along with mPP, a version of our algorithm in which L_i is set to one for all *i* is also considered. We refer to this version of the algorithm with the name of *baseline method* in the figures.

Moreover, to show the effect of distributed resource assignment and constant power cost for active servers, we implement a procedure to find a lower bound on the total energy cost with relaxation of these obstacles. To calculate this lower bound, for each VM, total energy cost ($c_i^p / C_j^p \left(P_j^p + P_j^0\right) T_e$) of serving that VM on each server is calculated and the smallest energy cost is selected. Summation of these energy costs generates a lower bound on the total energy cost in the system.

C. Numerical Results

Normalized total energy cost in the system using the EVMP algorithm, baseline method, and mPP algorithm is presented in Figure 6. EVMP-f1 results show the results for the first $f(\alpha)$ function whereas EVMP-f2 shows the results for the second $f(\alpha)$ function as discussed earlier.

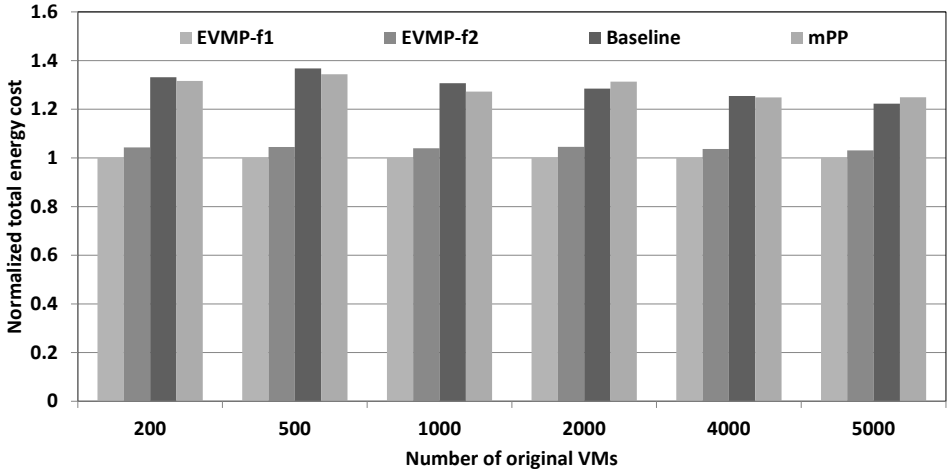


Figure 6. Normalized total energy cost of the system

As can be seen, EVMP reduces the total energy cost of VM placement solution by 24 to 36% with respect to the mPP algorithm. This amount of energy decrease is significant in cloud computing systems and can help reduce the operational cost of computing.

As can be seen, changing $f(\alpha)$ function from $f1$, which represent ideal VM copying case, to $f2$, which captures a scenario in which the cost of creating multiple copies of VMs is rather large, does not significantly increase the energy consumption of the datacenter (between 3 and 4% increase). This is due to the fact that EVMP algorithm adapts the decision regarding VM copying based on the $f(\alpha)$ function. This means that having a higher cost associated with creating copies of some VM, results in fewer number of VM copies being created by the algorithm.

Performance of the baseline algorithm which is based on assigning VMs using DP method is slightly worse than the performance of mPP method (~3% range) because baseline method does not place the VM on the server with the least resource availability and instead chooses the host server randomly in the selected server type.

Table II shows the relative performance of EVMP-f1 with respect to the derived lower bound on the total energy cost. There are two reasons behind the difference between the result of EVMP and the lower bound: i) imperfection of the algorithm, and ii) constant power consumption of the servers (independent from their utilization) and effect of the distributed resources in the datacenter.

TABLE II. PERFORMANCE OF THE EVMP-F1 W.R.T. LOWER BOUND COST AND AVERAGE NUMBER OF VM COPIES

# of original VMs	Performance w.r.t Lower bound
200	1.02
500	1.01
1000	1.05
2000	1.01
4000	1.08
5000	1.05

TABLE III shows the average number of VM copies created in EVMP-f1 and EVMP-f2 runs. The average number of VM copies on the final solution of the EVMP-f1 and EVMP-f2 is small compared to the average L_i for VMs which is 3. This shows that the EVMP algorithm does not create multiple copies of a VM unless it is beneficial for the energy cost of the system. Moreover, the average number of VM copies created in EVMP-f2 is smaller than the same number for EVMP-f1 which shows the adaptiveness of the EVMP algorithm in creating VM copies based on $f(\alpha)$ function.

TABLE III. PERFORMANCE OF THE EVMP-F2 W.R.T. LOWER BOUND COST AND AVERAGE NUMBER OF VM COPIES

# of original VMs	average # of VM copies for EVMP-F1	average # of VM copies for EVMP-F2
200	1.83	1.76
500	1.78	1.71
1000	1.78	1.72
2000	1.74	1.67
4000	1.71	1.66
5000	1.69	1.64

Effect of different L_i values on the performance of EVMP-f1 is reported in Figure 7. In this figure the normalized total energy costs of the VM placement solutions when using the EVMP algorithm and for different L_i values are shown. As can be seen, the cost difference between the EVMP solution and the solution of a version of EVMP that restricts the number of VM copies to two is 7% (on average). This shows around 20% energy reduction compared to the mPP algorithm even if the number of allowed VM copies is limited to two. The cost difference between the EVMP solution and the solution of a version of EVMP that restricts the number of VM copies to ten is around 10% (on average). Note that the function used to calculate the resource requirement for each VM copies for EVMP-f1 only accounts for the lower-bound amount of the processing resources required for each VM copy. Figure 8 shows the same comparison for EVMP using the second $f(\alpha)$ function. As can be seen, the energy cost reduction when the maximum allowable number of VM copies is increased to ten, is smaller (about 6% improvement) in case of using the second $f(\alpha)$ function. This is due to the fact that f2 function adds an energy cost penalty every time a new VM copy is added to the system.

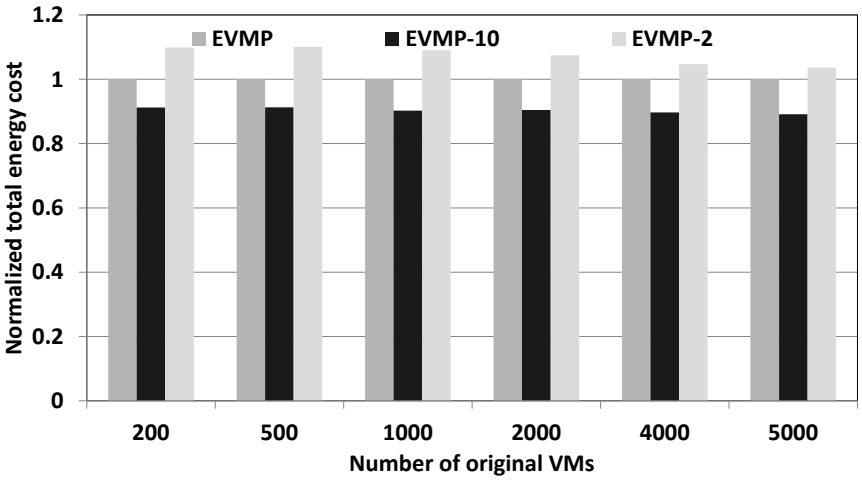


Figure 7. Normalized total energy cost of the VM placement solution using for different L_i for EVMP-f1

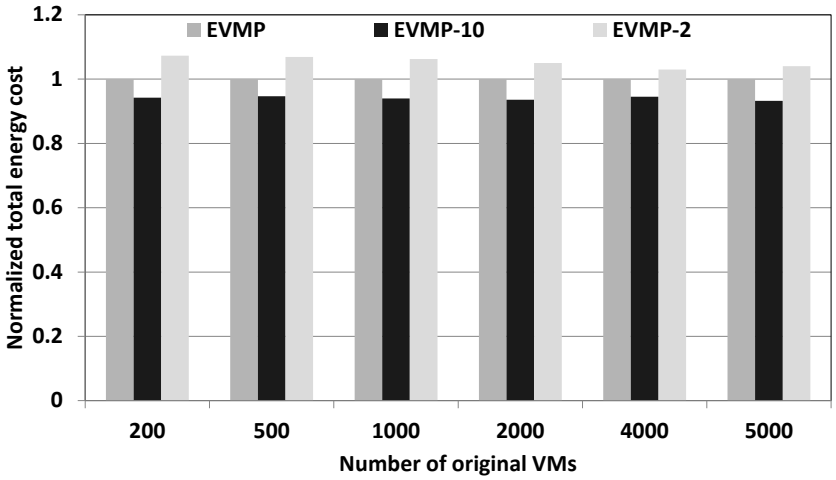


Figure 8. Normalized total energy cost of the VM placement solution using for different L_i for EVMP-f2

Figure 9 shows the average run-time of the EVMP, baseline and mPP methods for different number of VMs. Note that VM placement algorithm is called only a few times in each charge cycle (one hour in Amazon EC2 service [62]), e.g. 2-3 times per hour. Also to reduce the time complexity of the EVMP algorithm in case of bigger number of VMs, we can use a partitioning algorithm to assign a set of VMs to a cluster and then apply EVMP in each cluster in parallel.

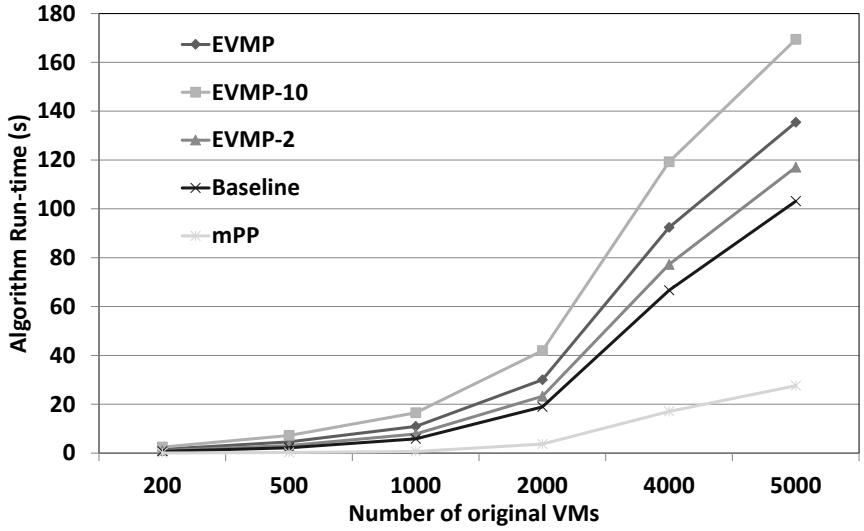


Figure 9. Run-time of EVMP for different number of VMs on 2.4GHZ E6600 server with 3GB of RAM from Intel

VII. Conclusion and Future Research Direction

A. Conclusions

In this chapter, we presented a review of the literature focusing on resource and power managers in datacenters. Moreover, we proposed a novel solution to increase the energy efficiency in datacenter that relies on generating multiple copies of each VM. To guarantee QoS for each VM, we considered fixed memory bandwidth requirement for each VM copy, added a limitation on the number of VM copies, and considered a VM replication energy and resource overhead. An algorithm based on dynamic programming and local search was proposed to determine the number of VM copies, then place them on servers to minimize the total energy cost in the cloud computing system. Using simulation results, we showed that this approach reduces the total energy cost ~20% with respect to the prior VM placement techniques. The effect of different parameters on the system performance was also evaluated using simulation results.

The proposed solution provides a flexible method to increase the energy efficiency of the cloud computing system and increases the resource availability in the datacenter. Cloud provider can decide how to service VMs with big processing resource requirements and how to distribute their requests among the servers to maximize the energy efficiency.

B. Possible Research Direction on energy-efficient datacenter design

There are plenty of opportunities to improve the state of the art in resource and power managers in datacenters. Advancing the design and adaptive control of datacenters with energy efficiency, SLAs, and total cost of ownership are the primary areas that one can contribute on, as detailed below.

The first step is to develop a theory for understanding the energy complexity of computational jobs. Today, energy efficiency is benchmarked relative to last year's product; any efficiency gain is touted as success. Instead, we wish to ask what level of efficiency is possible and measure solutions relative to this limit. One must thus develop key scientific principles to measure the energy complexity of applications. By combining energy complexity with time complexity of applications, we can then perform fundamental energy-performance tradeoffs at application programming level.

Informed by this new theory, one can then reconsider the design of the hardware platforms that comprise the energy-efficient datacenters. Key sources of inefficiency are the lack of energy proportional hardware and the overprovisioning of these servers to meet SLAs given the time-varying application resource demands. An energy-efficient datacenter exploits hardware heterogeneity and employs dynamic adaptation. Heterogeneity allows energy-optimized components to be brought to bear as an application characteristics change. Dynamic adaptation allows the datacenter to adapt and provision hardware components to meet varying workload and performance requirements, which, in turn, eliminates over-provisioning. Computing, storage, and networking subsystems of current datacenters exhibit dismal energy proportionality. One must attempt to redesign server architectures and network protocols with energy efficiency and energy-proportionality as the driving design constraint. On the storage front, we must construct hybrid storage systems that assign data to devices based on a fundamental understanding of access patterns and capacity-performance-efficiency tradeoffs.

To go beyond the incremental energy efficiency gains possible from component-wise optimization, one must consider the coordination and control of storage, networking, memory, compute, and physical infrastructure. By tackling the optimization problem for the datacenter as a whole, one can develop solutions at one layer that will be exploited at other layers. By using the mathematical underpinnings of control theory and stochastic modeling, these approaches enable reasoning about worst-case and average-case behavior of multi loop compositions of control approaches. One can then develop algorithms to globally manage compute, storage, and cyber-physical resources with the objective of minimizing the total energy dissipation while meeting SLAs.

Finally, to evaluate datacenter designs, one must develop new methodologies and simulation infrastructure to quantify the impact and prototype research ideas. Because of the complexity and scale of datacenter applications, conventional evaluation approaches cannot evaluate new innovations with reasonable turnaround time. Hence, we must design hierarchical models, which integrate performance and

energy estimates across detail and time granularities, and parallel cluster-on-a-cluster simulation techniques, which together allow us to quantitatively evaluate systems at an entirely new scale.

REFERENCES

[1] "Datacenter Dynamics Global Industry Census 2011," [Online]. Available: http://www.datacenterdynamics.com.br/research/market-growth-2011-2012.

[2] G. Cook, "How clean is your cloud? Catalysing an energy revolution," Greenpeace International, 2012.

[3] ENERGY STAR, "Report to Congress on Server and Datacenter Energy Efficiency Public Law 109-431," U.S.Environmental Protection Agency, Washington, D.C., 2007.

[4] D. Meisner, B. Gold and T. Wenisch, "PowerNap: eliminating server idle power," in *Proceedings of the ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, Washington, DC, 2009.

[5] S. Pelley, D. Meisner, T. F. Wenisch and J. VanGilder, "Understanding and abstracting total datacenter power," in *workhop on Energy-Efficient Design*, 2009.

[6] "EPA confenrece on Enterprise Servers and Datacenters: Opportunities for Energy Efficiency," EPA, Lawrence Berkeley National Laboratory, 2006.

[7] L. A. Barroso and U.Hölzle, "The Case for Energy-Proportional Computing," *IEEE Computer*, vol. 40, 2007.

[8] L. A. Barroso and U. Holzle, The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines, Morgan & Claypool Publishers, 2009.

[9] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt and A. Warfield, "Xen and the art of virtualization," in *19th ACM Symposium on Operating Systems Principles*, 2003.

[10] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinsk, G. Lee, D. Patterson, A. Rabkin, I. Stoica and M. Zaharia, "A view of cloud computing," *Commun ACM*, vol. 53, no. 4, pp. 50-58, 2010.

[11] R. Buyya, "Market-oriented cloud computing: Vision, hype, and reality of delivering computing as the 5th utility," in *9th IEEE/ACM International Symposium on Cluster Computing and the Grid, CCGRID*, 2009.

[12] A. Karve, T. Kimbre, G. Pacifici, M. Spreitzer, M. Steinder, M. Sviridenko and A. Tantawi, "Dynamic placement for clustered web applications," in *15th International Conference on World Wide Web, WWW'06*, 2006.

[13] C. Tang, M. Steinder, M. Spreitzer and G. Pacifici, "A scalable application placement controller for enterprise datacenters," in *16th International World Wide Web Conference, WWW2007*, 2007.

[14] F. Chang, J. Ren and R. Viswanathan, "Optimal resource allocation in clouds," in *3rd IEEE International Conference on Cloud Computing, CLOUD 2010*, 2010.

[15] J. S. Chase, D. C. Anderson, P. N. Thakar, A. M. Vahdat and R. P. Doyle, "Managing energy and server resources in hosting centers," in *18th ACM Symposium on Operating Systems Principles (SOSP'01)*, 2001.

[16] E. Pakbaznia, M. GhasemAzar and M. Pedram, "Minimizing datacenter cooling and server power costs," in *Proc. of Design Automation and Test in Europe*, 2010.

[17] S. Srikantaiah, A. Kansal and F. Zhao, "Energy aware consolidation for cloud computing," in *Conference on Power aware computing and systems (HotPower'08)*, 2008.

[18] J. Kim, M. Ruggiero, D. Atienza and M. Lederberger, "Correlation-aware virtual machine allocation for energy-efficient datacenters," in *Proceedings of the Conference on Design, Automation and Test in Europe*, 2013.

[19] I. Hwang and M. Pedram, "Portfolio theory-based resource assignment in a cloud computing system," in *IEEE 5th International Conference on Cloud Computing (CLOUD)*, 2012.

[20] A. Corradi, M. Fanelli and L. Foschini, "VM consolidation: A real case based on OpenStack Cloud," *Future Generation Computer Systems*, pp. 118-127, 2014.

[21] Z. Xiao, W. Song and Q. Chen, "Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 6, pp. 1107,1117, 2013.

[22] D. Novakovic, N. Vasic, S. Novakovic, D. Kostic and R. Bianchini, "Deepdive: Transparently identifying and managing performance interference in virtualized environments," EPFL, 2013.

[23] A. Verrna, P. Ahuja and A. Neogi, "pMapper: Power and migration cost aware application placement in virtualized systems," in *ACM/IFIP/USENIX 9th International Middleware Conference*, 2008.

[24] S. Martello and P. Toth, Knapsack Problems: Algorithms and Computer Implementations, Wiley, 1990.

[25] S. Takeda and T. Takemura, "A rank-based vm consolidation method for power saving in datacenters," *Information and Media Technologies*, vol. 5, no. 3, pp. 994-1002, 2010.

[26] H. Goudarzi and M. Pedram, "Maximizing profit in the cloud computing system via resource allocation," in *Proc. of international workhop on Datacenter Performance*, 2011.

[27] B. Urgaonkar, P. Shenoy and T. Roscoe, "Resource Overbooking and Application Profiling in Shared Hosting Platforms," in *Symposium on Operating Systems Design and Implementation*, 2002.

[28] Z. Liu, M. S. Squillante and J. L. Wolf, "On maximizing service-level-agreement profits," in *Third ACM Conference on Electronic Commerce*, 2001.

[29] K. Le, R. Bianchini, T. D. Nguyen, O. Bilgir and M. Martonosi, "Capping the brown energy consumption of internet services at low cost," in *International Conference on Green Computing (Green Comp)*, 2010.

[30] L. Zhang and D. Ardagna, "SLA based profit optimization in autonomic computing systems," in *Proceedings of the Second International Conference on Service Oriented Computing*, 2004.

[31] D. Ardagna, M. Trubian and L. Zhang, "SLA based resource allocation policies in autonomic environments," *Journal of Parallel and Distributed Computing*, vol. 67, no. 3, pp. 259-270, 2007.

[32] D. Ardagna, B. Panicucci, M. Trubian and L. Zhang, "Energy-Aware Autonomic Resource Allocation in Multi-Tier Virtualized Environments," *IEEE Transactions on Services Computing*, vol. 99, 2010.

[33] H. Goudarzi and M. Pedram, "Multi-dimensional SLA-based resource allocation for multi-tier cloud computing systems," in *proceeding of 4th IEEE conference on cloud computing (Cloud 2011)*, 2011.

[34] G. Tesauro, N. K. Jong, R. Das and M. N. Bennani, "A hybrid reinforcement learning approach to autonomic resource allocation," in *Proceedings of International Conference on Autonomic Computing (ICAC '06)*, 2006.

[35] D. Kusic, J. O. Kephart, J. E. Hanson, N. Kandasamy and G. Jiang, "Power and performance management of virtualized computing environments via lookahead control," in *Proceedings of International Conference on Autonomic Computing (ICAC '08)*, 2008.

[36] H. Goudarzi, M. Ghasemazar and M. Pedram, "SLA-based Optimization of Power and Migration Cost in Cloud Computing," in *12th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid)*, 2012.

[37] E. Feller, C. Morin and A. Esnault, "A case for fully decentralized dynamic VM consolidation in clouds," in *IEEE 4th International Conference on Cloud Computing Technology and Science (CloudCom)*, 2012.

[38] C. Mastroianni, M. Meo and G. Papuzzo, "Probabilistic Consolidation of Virtual Machines in Self-Organizing Cloud Data Centers," *IEEE Transactions on Cloud Computing*, vol. 1, no. 2, 2013.

[39] M. N. Bennani and D. A. Menasce, "Resource allocation for autonomic datacenters using analytic performance models," in *Second International Conference on Autonomic Computing*, 2005.

[40] B. Urgaonkar, G. Pacifici, P. Shenoy, M. Spreitzer and A. Tantawi, "An analytical model for multi-tier internet services and its applications," in *SIGMETRICS 2005: International Conference on Measurement and Modeling of Computer Systems*, 2005.

[41] M. Pedram and I.Hwang, "Power and performance modeling in a virtualized server system," in *39th International Conference on Parallel Processing workhops (ICPPW)*, 2010.

[42] A. Chandra, W. Gongt and P. Shenoy, "Dynamic resource allocation for shared datacenters using online measurements," in *International Conference on Measurement and Modeling of Computer Systems ACM SIGMETRICS*, 2003.

[43] N. Bobroff, A. Kochut and K. Beaty, "Dynamic Placement of Virtual Machines for Managing SLA Violations," in *Proceedings of the 10th IFIP/IEEE International Symposium on Integrated Management (IM2007)*, 2007.

[44] A. Beloglazov and R. Buyya, "Adaptive threshold-based approach for energy-efficient consolidation of virtual machines in cloud data centers," in *Proceedings of the 8th International Workshop on Middleware for Grids, Clouds and e-Science*, 2010.

[45] A. Beloglazov and R. Buyya, "Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers," *Concurrency and Computation: Practice and Experience*, pp. 1397-142, 2012.

[46] M. Wang, X. Meng and L. Zhang, "Consolidating virtual machines with dynamic bandwidth demand in data centers," in *IEEE INFOCOM*, 2011.

[47] T. Hirofuchi, H. Nakada, S. Itoh and S. Sekiguchi, "Reactive consolidation of virtual machines enabled by postcopy live migration," in *Proceedings of the 5th international workshop on Virtualization technologies in*

distributed computing, 2011.

[48] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang and X. Zhu, "No "power" struggles: Coordinated multi-level power management for the datacenter," *ACM SIGPLAN Notices*, vol. 43, no. 3, pp. 48-59, 2008.

[49] X. Fan, W. Weber and L. A. Barroso, "Power provisioning for a warehouse-sized computer," in *Proceedings of the 34th Annual International symposium on Computer Architecture*, San Diego, CA, 2007.

[50] S. Pelley, D. Meisner, P. Zandevakili, T. F. Wenisch and J. Underwood, "Power Routing : Dynamic Power Provisioning in the Datacenter," in *ASPLOS '10: Architectural Support for Programming Languages and Operating Systems*, 2010.

[51] A. Gandhi, M. Harchol-Balter, R. Das and C. Lefurgy, "Optimal power allocation in server farms," in *international joint conference on Measurement and modeling of computer systems (SIGMETRICS '09)*, 2009.

[52] W. Felter, K. Rajamani, T. Keller and C. Rusu, "A performance-conserving approach for reducing peak power consumption in server systems," in *19th annual international conference on Supercomputing (ICS '05)*, 2005.

[53] M. Srivastava, A. Chandrakasan and R. Brodersen, "Predictive system shutdown and other architectural techniques for energy efficient programmable computation," *IEEE Trans. on VLSI*, 1996.

[54] Q. Qiu and M. Pedram, "Dynamic Power Management Based on Continuous-Time Markov Decision Processes," in *ACM design automation confernece (DAC'99)*, 1999.

[55] G. Dhiman and T. S. Rosing, "Dynamic power management using machine learning," in *ICCAD '06*, 2006.

[56] E. Elnozahy, M. Kistler and R. Rajamony, "Energy-Efficient Server Clusters," in *Proc. 2nd workhop Power-Aware Computing Systems,*, 2003.

[57] D. Meisner, C. Sadler, L. Barroso, W. Weber and T. Wenisch, "Power Management of Online Data-Intensive Services," in *Proceedings of the 38th Annual International symposium on Computer Architecture*, 2011.

[58] X. Wang and Y. Wang, "Co-con: Coordinated control of power and application performance for virtualized server clusters," in *IEEE 17th International workhop on Quality of Service (IWQoS)*, 2009.

[59] R. Buyya and A. Beloglazov, "Energy efficient resource management in virtualized cloud datacenters," in *10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing (CCGrid)*, 2010.

[60] "Google Cloud Platform Trace," [Online]. Available: https://cloud.google.com/tools/cloud-trace.

[61] "http://ark.intel.com/," [Online].

[62] "http://aws.amazon.com/ec2/#pricing," [Online].