Power-Efficient Assignment of Virtual Machines to Physical Machines

Jordi Arjona Aroca[1], Antonio Fernández Anta[2], Miguel A. Mosteiro[3(✉)], Christopher Thraves[4], and Lin Wang[5]

[1] Universidad Carlos III de Madrid, Madrid, Spain
jorge.arjona@imdea.org
[2] Institute IMDEA Networks, Madrid, Spain
antonio.fernandez@imdea.org
[3] Department of Computer Science, Kean University, Union, NJ, USA
mmosteir@kean.edu
[4] CNRS-LAAS and University of Toulouse - LAAS, Tolouse, France
cthraves@laas.fr
[5] Institute of Computing Technology, Chinese Academy of Sciences and University of Chinese Academy of Sciences, Beijing, China
wanglin@ict.ac.cn

Abstract. Motivated by current trends in cloud computing, we study a version of the generalized assignment problem where a set of virtual processors has to be implemented by a set of *identical* processors. For literature consistency, we say that a set of virtual machines (VMs) is assigned to a set of physical machines (PMs). The optimization criteria is to minimize the power consumed by all the PMs. We term the problem Virtual Machine Assignment (VMA). Crucial differences with previous work include a variable number of PMs, that each VM must be assigned to exactly one PM (i.e., VMs cannot be implemented fractionally), and a minimum power consumption for each active PM. Such infrastructure may be strictly constrained in the number of PMs or in the PMs' capacity, depending on how costly (in terms of power consumption) it is to add a new PM to the system or to heavily load some of the existing PMs. Low usage or ample budget yields models where PM capacity and/or the number of PMs may be assumed unbounded for all practical purposes. We study four VMA problems depending on whether the capacity or the number of PMs is bounded or not. Specifically, we study hardness and online competitiveness for a variety of cases. To the best of our knowledge, this is the first comprehensive study of the VMA problem for this cost function.

Keywords: Cloud computing · Generalized assignment · Scheduling · Load balancing · Power efficiency

This work has been supported in part by the Comunidad de Madrid grant Cloud4BigData-CM, the MINECO grant TEC2011-29688-C02-01, the National Natural Science Foundation of China grant 61020106002, the National Science Foundation (CCF-0937829, CCF-1114930), and Kean University UFRI grant.

F. Pop and M. Potop-Butucaru (Eds.): ARMS-CC 2014, LNCS 8907, pp. 71–88, 2014.
DOI: 10.1007/978-3-319-13464-2_6

1 Introduction

The current pace of technology developments, and the continuous change in business requirements, may rapidly yield a given proprietary computational platform obsolete, oversized, or insufficient. Thus, outsourcing has recently become a popular approach to obtain computational services without incurring in amortization costs. Furthermore, in order to attain flexibility, such service is usually virtualized, so that the user may tune the computational platform to its particular needs. Users of such service need not to be aware of the particular implementation, they only need to specify the virtual machine they want to use. This conceptual approach to outsourced computing has been termed *cloud computing*, in reference to the cloud symbol used as an abstraction of a complex infrastructure in system diagrams. Current examples of cloud computing providers include Amazon Web Services [3], Rackspace [34], and Citrix [17].

Depending on what the specific service provided is, the cloud computing model comes in different flavors, such as *infrastructure as a service*, *platform as a service*, *storage as a service*, etc. In each of these models, the user may choose specific parameters of the computational resources provided. For instance, processing power, memory size, communication bandwidth, etc. Thus, in a cloud-computing service platform, various ***virtual machines (VM)*** with user-defined specifications must be implemented by, or ***assigned to***[1], various ***physical machines (PM)***[2]. Furthermore, such a platform must be scalable, allowing to add more PMs, should the business growth require such expansion. In this work, we call this problem the ***Virtual Machine Assignment (VMA)*** problem.

The optimization criteria for VMA depends on what the particular objective function sought is. From the previous discussion, it can be seen that, underlying VMA, there is some form of bin-packing problem. However, in VMA the number of PMs (i.e., bins for bin packing) may be increased if needed. Since CPU is generally the dominant power consumer in a server [7], VMA is usually carried out according to CPU workloads. With only the static power consumption of servers considered, previous work related to VMA has focused on minimizing the number of active PMs (cf. [11] and the references therein) in order to minimize the total static energy consumption. This is commonly known as VM consolidation [26,32]. However, despite the static power, the dynamic power consumption of a server, which has been shown to be superlinear on the ***load*** of a given computational resource [9,23], is also significant and cannot be ignored. Since the definition of load is not precise, we borrow the definition in [7] and define the load of a server as the amount of active cycles per second a task requires, an absolute metric independent of the operating frequency or the number of cores of a PM. The superlinearity property of the dynamic power consumption is also confirmed

[1] The cloud-computing literature uses instead the term *placement*. We choose here the term assignment for consistency with the literature on general assignment problems.

[2] We choose the notation VM and PM for simplicity and consistency, but notice that our study applies to any computational resource assignment problem, as long as the minimization function is the one modeled here.

by the results in [7]. As a result, when taking into account both parts of power consumption, the use of extra PMs may be more efficient energy-wise than a minimum number of heavily-loaded PMs. This inconsistency with the literature in VM consolidation has been supported by the results presented in [7] and, hence, we claim that the way consolidation has been traditionally performed has to be reconsidered. In this work, we combine both power-consumption factors and explore the most energy-efficient way for VMA. That is, for some parameters $\alpha > 1$ and $b > 0$, we seek to minimize the sum of the α powers of the PMs loads *plus* the fixed cost b of using each PM.

Physical resources are physically constrained. A PMs infrastructure may be strictly constrained in the number of PMs or in the PMs CPU capacity. However, if usage patterns indicate that the PMs will always be loaded well below their capacity, it may be assumed that the capacity is unlimited. Likewise, if the power budget is very big, the number of PMs may be assumed unconstrained for all practical purposes. These cases yield 4 VMA subproblems, depending on whether the capacity and the number of PMs is limited or not. We introduce these parameters denoting the problem as **(C,m)-VMA**, where C is the PM CPU capacity, m is the maximum number of PMs, and each of these parameters is replaced by a dot if unbounded.

In this work, we study the hardness and online competitiveness of the VMA problem. Specifically, we show that VMA is NP-hard *in the strong sense* (in particular, we observe that (C, m)-VMA is strongly NP-complete). Thus, VMA problems do not have a fully polynomial time approximation scheme (FPTAS). Nevertheless, using previous results derived for more general objective functions, we notice that $(\cdot, m)$- and $(\cdot, \cdot)$-VMA have a polynomial time approximation scheme (PTAS). We also show various lower and upper bounds on the offline approximation and the online competitiveness of VMA. Rather than attempting to obtain tight bounds for particular instances of the parameters of the problem (C, m, α, b) we focus on obtaining *general bounds*, whose parameters can be instantiated for the specific application. The bounds obtained show interesting trade-offs between the PM capacity and the fixed cost of adding a new PM to the system. To the best of our knowledge, this is the first VMA study that is focused on power consumption.

Roadmap. The paper is organized as follows. In what remains of this section, we define formally the $(\cdot, \cdot)$-VMA problem, we overview the related work, and we describe our results in detail. Section 2 includes some preliminary results that will be used throughout the paper. The offline and online analyses are included in Sects. 3 and 4 respectively. Section 5 discusses some practical issues and provides some useful insights regarding real implementation. For succinctness, many of the proofs are left to the full version of this paper in [8].

1.1 Problem Definition

We describe the $(\cdot, \cdot)$-VMA problem now. Given a set $S = \{s_1, \ldots, s_m\}$ of $m > 1$ identical physical machines (PMs) of capacity C; rational numbers μ, α and b, where $\mu > 0$, $\alpha > 1$ and $b > 0$; a set $D = \{d_1, \ldots, d_n\}$ of n virtual machines and a function $\ell : D \rightarrow \mathbb{R}$ that gives the CPU load each virtual machine incurs[3], we aim to obtain a partition $\pi = \{A_1, \ldots, A_m\}$ of D, such that $\ell(A_i) \leq C$, for all i. Our objective will be then minimizing the power consumption given by the function

$$P(\pi) = \sum_{i \in [1,m]: A_i \neq \emptyset} \left(\mu \Big(\sum_{d_j \in A_i} \ell(d_j) \Big)^{\alpha} + b \right). \quad (1)$$

Let us define the function $f(\cdot)$, such that $f(x) = 0$ if $x = 0$ and $f(x) = \mu x^{\alpha} + b$ otherwise. Then, the objective function is to minimize $P(\pi) = \sum_{i=1}^{m} f(\ell(A_i))$. The parameter μ is used for consistency with the literature. For clarity we will consider $\mu = 1$ in the rest of the paper. All the results presented apply for other values of μ.

We also study several special cases of the VMA problem, namely (C, m)-VMA, $(C, \cdot)$-VMA, $(\cdot, m)$-VMA and $(\cdot, \cdot)$-VMA. (C, m)-VMA refers to the case where both the number of available PMs and its capacity are fixed. $(\cdot, \cdot)$-VMA, where $(\cdot)$ denotes unboundedness, refers to the case where both the number of available PMs and its capacity are unbounded (i.e., C is larger than the total load of the VMs that can ever be in the system at any time, or m is larger than the number of VMs that can ever be in the system at any time). $(C, \cdot)$-VMA and $(\cdot, m)$-VMA are the cases where the number of available PMs and their capacity is unbounded, respectively.

1.2 Related Work

To the best of our knowledge, previous work on VMA has been only experimental [16, 27, 30, 36] or has focused on different cost functions [1, 11, 15, 18]. First, we provide an overview of previous theoretical work for related assignment problems (storage allocation, scheduling, network design, etc.). The cost functions considered in that work resemble or generalize the power cost function under consideration here. Secondly, we overview related experimental work.

Chandra and Wong [15], and Cody and Coffman [18] study a problem for storage allocation that is a variant of $(\cdot, m)$-VMA with $b = 0$ and $\alpha = 2$. Hence, this problem tries to minimize the sum of the squares of the machine-load vector for a fixed number of machines. They study the offline version of the problem and provide algorithms with constant approximation ratio. A significant leap was taken by Alon et al. [1], since they present a PTAS for the problem of minimizing the L_p norm of the load vector, for any $p \geq 1$. This problem has the previous one as special case, and is also a variant of the $(\cdot, m)$-VMA problem when $p = \alpha$ and $b = 0$. Similarly, Alon et al. [2] extended this work for a more general set of functions, that include $f(\cdot)$ as defined above. Hence, their results can be directly applied in the $(\cdot, m)$-VMA problem. Later, Epstein et al. [20] extended [2] further for the uniformly related machines case. We will use these results in Sect. 3 in the analysis of the offline case of $(\cdot, m)$-VMA and $(\cdot, \cdot)$-VMA.

[3] For convenience, we overload the function $\ell(\cdot)$ to be applied over sets of virtual machines, so that for any set $A \subseteq D, \ell(A) = \sum_{d_j \in A} \ell(d_j)$.

Bansal, Chan, and Pruhs minimize arbitrary power functions for speed scaling in job scheduling [9]. The problem is to schedule the execution of n computational jobs on a *single* processor, whose speed may vary within a countable collection of intervals. Each job has a release time, a processing work to be done, a weight characterizing its importance, and its execution can be suspended and restarted later without penalty. A scheduler algorithm must specify, for each time, a job to execute and a speed for the processor. The goal is to minimize the weighted sum of the flow times over all jobs plus the energy consumption, where the flow time of a job is the time elapsed from release to completion and the energy consumption is given by s^α where s is the processor speed and $\alpha > 1$ is some constant. For the online algorithm *shortest remaining processing time first*, the authors prove a $(3 + \epsilon)$ competitive ratio for the objective of total weighted flow plus energy. Whereas for the online algorithm *highest density first (HDF)*, where the density of a job is its weight-to-work ratio, they prove a $(2 + \epsilon)$ competitive ratio for the objective of fractional weighted flow plus energy.

Recently, Im, Moseley, and Pruhs studied online scheduling for general cost functions of the flow time, with the only restriction that such function is non-decreasing [24]. In their model, a collection of jobs, each characterized by a release time, a processing work, and a weight, must be processed by a *single* server whose speed is variable. A job can be suspended and restarted later without penalty. The authors show that HDF is $(2+\epsilon)$-speed $O(1)$-competitive against the optimal algorithm on a unit speed-processor, for all non-decreasing cost functions of the flow time. Furthermore, they also show that this ratio cannot be improved significantly proving impossibility results if the cost function is not uniform among jobs or the speed cannot be significantly increased.

A generalization of the above problem is studied by Gupta, Krishnaswamy, and Pruhs in [23]. The question addressed is how to assign jobs, *possibly fractionally*, to unrelated parallel machines in an online fashion in order to minimize the sum of the α-powers of the machine loads plus the assignment costs. Upon arrival of a job, the algorithm learns the increase on the load and the cost of assigning a unit of such job to a machine. Jobs cannot be suspended and/or reassigned. The authors model a greedy algorithm that assigns a job so that the cost is minimized as solving a mathematical program with constraints arriving online. They show a competitive ratio of α^α with respect to the solution of the dual program which is a lower bound for the optimal. They also show how to adapt the algorithm to integral assignments with a $O(\alpha)^\alpha$ competitive ratio, which applies directly to our $(\cdot, m)$-VMA problem. References to previous work on the particular case of minimizing energy with deadlines can be found in this paper.

Similar cost functions have been considered for the minimum cost network-design problem. In this problem, packets have to be routed through a (possibly multihop) network of speed scalable routers. There is a cost associated to assigning a packet to a link and to the speed or load of the router. The goal is to route all packets minimizing the aggregated cost. In [4,5] the authors show offline algorithms for this problem with undirected graph and homogeneous link cost functions that achieve polynomial and poly-logarithmic approximation, respectively.

The cost function is the α-th power of the link load plus a link assignment cost, for any constant $\alpha > 1$. The same problem and cost function is studied in [23]. Bansal *et al.* [10] study a minimum-cost virtual circuit multicast routing problem with speed scalable links. They give a polynomial-time $O(\alpha)$-approximation offline algorithm and a polylog-competitive online algorithm, both for the case with homogeneous power functions. They also show that the problem is APX-hard in the case with heterogeneous power functions and there is no polylog-approximation when the graph is directed. Recently, Antoniadis *et al.* [6] improved the results by providing a simple combinatorial algorithm that is $O(\log^{\alpha} n)$-approximate, from which we can construct an $\widetilde{O}(\log^{3\alpha+1} n)$-competitive online algorithm. The $(\cdot, m)$-VMA problem can be seen as a especial case of the problem considered in these papers in which there are only two nodes, source and destination, and m parallel links connecting them.

To the best of our knowledge, the problem of minimizing the power consumption (given in Eq. 1) with capacity constraints (i.e., the (C, m)-VMA and $(C, \cdot)$-VMA problems) has received very limited attention, in the realm of both VMA and network design, although the approaches in [5,10] are related to or based on the solutions for the capacitated network-design problem [14].

The experimental work related to VMA is vast and its detailed overview is out of the scope of this paper. Some of this work does not minimize energy [13,28,31] or it applies to a model different than ours (VM migration [33,35], knowledge of future load [29,35], feasibility of allocation [11], multilevel architecture [25,30,33], interconnected VMs [12], etc.). On the other hand, some of the experimental work where minimization of energy is evaluated focus on a more restrictive cost function [25,38,40].

In [35], the authors focus on an energy-efficient VM placement problem with two requirements: CPU and disk. These requirements are assumed to change dynamically and the goal is to consolidate loads among servers, possibly using migration at no cost. In our model VMs assignment is based on a CPU requirement that does not change and migration is not allowed. Should any other resource be the dominating energy cost, the same results apply for that requirement. Also, if loads change and migration is free, an offline algorithm can be used each time that a load changes or a new VM arrives. In [35] it is shown experimentally that energy-efficient VMA does not merely reduce to a packing problem. That is, to minimize the number of PMs used even if their load is close to their maximum capacity. For our model, we show here that the optimal load of a given server is a function only of the fixed cost of being active (b) and the exponential rate of power increase on the load (α). That is, the optimal load is not related to the maximum capacity of a PM.

1.3 Our Results

In this work, we study offline and online versions of the four versions of the VMA problem. For the offline problems, the first fact we observe is that there is a hard decision version of (C, m)-VMA: Is there a feasible partition π of the set D of VMs? By reduction from the 3-Partition problem, it can be shown that this decision problem is strongly NP-complete.

We then show that the $(\cdot,\cdot)$-VMA, $(C,\cdot)$-VMA, and $(\cdot,m)$-VMA problems are NP-hard in the strong sense, even if α is constant. This result implies that these problems do not have FPTAS, even if α is constant. However, we show that the $(\cdot,\cdot)$-VMA and $(\cdot,m)$-VMA problems have PTAS, while the $(C,\cdot)$-VMA problem can not be approximated beyond a ratio of $\frac{3}{2}\cdot\frac{\alpha-1+(\frac{2}{3})^\alpha}{\alpha}$ (unless P = NP). On the positive side, we show how to use an existing Asymptotic PTAS [21] to obtain algorithms that approximate the optimal solution of $(C,\cdot)$-VMA. (See Table 1.)

Then we move on to online VMA algorithms. We show various upper and lower bounds on the competitive ratio of the four versions of the problem. (See Table 1.) Observe that the results are often different depending on whether x^* is smaller than C or not. In fact, when $x^* < C$, there is a lower bound of $\frac{(3/2)2^\alpha-1}{2^\alpha-1}$

Table 1. Summary of bounds on the approximation/competitive ratio ρ. All lower bounds are existential. The number of PMs in an optimal $(C,\cdot)$-VMA solution is denoted as m^*. The number of PMs in an optimal Bin Packing solution is denoted as $\overline{m}$. The load that minimizes the ratio power consumption against load is denoted as x^*. The subset of VMs with load smaller than x^* is denoted as D_s.

VMA subprob.	$x^* < C$	$x^* \geq C$
$(C,\cdot)$ offline	$\rho \geq \frac{3}{2}\frac{\alpha-1+(2/3)^\alpha}{\alpha}$	$\rho \geq \frac{3}{2}\frac{\alpha-1+(2/3)^\alpha}{\alpha}$
	$\rho < \frac{\overline{m}}{m^*}\left(1+\epsilon+\frac{1}{\alpha-1}+\frac{1}{\overline{m}}\right)$	$\rho < 1+\epsilon+\frac{C^\alpha}{b}+\frac{1}{\overline{m}}$
$(C,\cdot)$ online	$\rho \geq \frac{(3/2)2^\alpha-1}{2^\alpha-1}$	$\rho \geq \frac{C^\alpha+2b}{b+\max\{C^\alpha,2(C/2)^\alpha+b\}}$
	$\rho = 1$ if $D_s = \emptyset$, else $\rho \leq \left(1-\frac{1}{\alpha}\left(1-\frac{1}{2^\alpha}\right)\right)\left(2+\frac{x^*}{\ell(D_s)}\right)$	$\rho \leq \frac{2b}{C}\left(1+\frac{1}{(\alpha-1)2^\alpha}\right)\left(2+\frac{C}{\ell(D)}\right)$
(C,m) online	$\rho \geq \frac{(3/2)2^\alpha-1}{2^\alpha-1}$	$\rho \geq \frac{C^\alpha+2b}{b+\max\{C^\alpha,2(C/2)^\alpha+b\}}$
$(\cdot,\cdot)$ online	$\rho \geq \frac{(3/2)2^\alpha-1}{2^\alpha-1}$	not applicable
	$\rho = 1$ if $D_s = \emptyset$, else $\rho \leq \left(1-\frac{1}{\alpha}\left(1-\frac{1}{2^\alpha}\right)\right)\left(2+\frac{x^*}{\ell(D_s)}\right)$	
$(\cdot,m)$ online	$\rho \geq \max\{\frac{(3/2)2^\alpha-1}{2^\alpha-1},\frac{3^\alpha}{2^{\alpha+2}+\epsilon}\}$	not applicable
	$\rho \leq O(\alpha)^\alpha$ In [23]	
$(\cdot,2)$ online	$\rho \geq \max\{\frac{3^\alpha}{2^{\alpha+1}},\frac{(3/2)2^\alpha-1}{2^\alpha-1},\frac{3^\alpha}{2^{\alpha+2}+\epsilon}\}$	not applicable
	$\rho = 1$ if $\ell(D) \leq \sqrt[\alpha]{b/(2^\alpha-2)}$, else $\rho \leq \max\{2,\left(\frac{3}{2}\right)^{\alpha-1}\}$	

Table 2. Summary of bounds on the approximation/competitive ratio ρ for $\alpha = 3$, $b = 2$, and $C = 2$ on the left and $C = 1$ on the right. All lower bounds are existential. The number of PMs in an optimal $(C, \cdot)$-VMA solution is denoted as m^*. The number of PMs in an optimal Bin Packing solution is denoted as $\overline{m}$. The load that minimizes the ratio power consumption against load is denoted as x^*. The subset of VMs with load smaller than x^* is denoted as D_s.

VMA subprob.	$x^* < C$	$x^* \geq C$
$(C, \cdot)$ offline	$\rho \geq \frac{11}{9}$	$\rho \geq \frac{11}{9}$
	$\rho < \frac{\overline{m}}{m^*}\left(\frac{3}{2} + \epsilon + \frac{1}{\overline{m}}\right)$	$\rho < \frac{3}{2} + \epsilon + \frac{1}{\overline{m}}$
$(C, \cdot)$ online	$\rho \geq \frac{11}{7}$	$\rho \geq \frac{20}{17}$
	$\rho \leq \frac{17}{12}\left(1 + \frac{1}{2\ell(D_s)}\right)$	$\rho \leq \frac{17}{2}\left(1 + \frac{1}{2\ell(D)}\right)$
(C, m) online	$\rho \geq \frac{11}{7}$	$\rho \geq \frac{20}{17}$
$(\cdot, \cdot)$ online	$\rho \geq \frac{11}{7}$	not applicable
	$\rho \leq \frac{17}{12}\left(1 + \frac{1}{2\ell(D_s)}\right)$	
$(\cdot, m)$ online	$\rho \geq \frac{11}{7}$	not applicable
$(\cdot, 2)$ online	$\rho \geq \frac{11}{7}$	not applicable
	$\rho \leq \frac{9}{4}$	

that applies to all versions of the problem. The bounds are given as a function of the input parameters of the problem, in order to allow for tighter expressions. To provide intuition on how tight the bounds are, we instantiate them for a realistic[4] value of $\alpha = 3$, and normalized values of $b = 2$ and $C \in \{1, 2\}$. The resulting bounds are shown in Table 2. As can be observed, the resulting upper and lower bounds are not very far in general.

2 Preliminaries

The following claims will be used in the analysis. We call ***power rate*** the power consumed per unit of load in a PM. Let x be the load of a PM. Then, its power rate is computed as $f(x)/x$. The load at which the power rate is minimized, denoted x^*, is the ***optimal load***, and the corresponding rate is the ***optimal power rate*** $\varphi^* = f(x^*)/x^*$. Using calculus we get the following observation.

[4] The values for α in the servers studied in [7] (denoted as Erdos and Nemesis) are close to 1.5 and 3 and x^* values of $0.76C$ and $0.9C$ respectively.

Observation 1. *The optimal load is* $x^* = (b/(\alpha-1))^{1/\alpha}$. *Additionaly, for any* $x \neq x^*$, $f(x)/x > \varphi^*$.

The following lemmas will be used in the analysis.

Lemma 1. *Consider two solutions* $\pi = \{A_1, \ldots, A_m\}$ *and* $\pi' = \{A'_1, \ldots, A'_m\}$ *of an instance of the VMA problem, such that for some* $x, y \in [1, m]$ *it holds that*

- $A_x \neq \emptyset$ *and* $A_y \neq \emptyset$;
- $A'_x = A_x \cup A_y$, $A'_y = \emptyset$, *and* $A_i = A'_i$, *for all* $i \neq x$ *and* $i \neq y$; *and*
- $\ell(A_x) + \ell(A_y) \leq \min\{x^*, C\}$.

Then, $P(\pi') < P(\pi)$.

From this lemma, it follows that the global power consumption can be reduced by having 2 VMs together in the same PM, when its aggregated load is smaller than $\min\{x^*, C\}$, instead of moving one VM to an unused PM. When we keep VMs together in a given partition we say that we are *using* Lemma 1.

Lemma 2. *Consider two solutions* $\pi = \{A_1, \ldots, A_m\}$ *and* $\pi' = \{A'_1, \ldots, A'_m\}$ *of an instance of the VMA problem, such that for some* $x, y \in [1, m]$ *it holds that*

- $A_x \cup A_y = A'_x \cup A'_y$, *while* $A_i = A'_i$, *for all* $x \neq i \neq y$;
- *none of* A_x, A_y, A'_x, *and* A'_y *is empty; and*
- $|\ell(A_x) - \ell(A_y)| < |\ell(A'_x) - \ell(A'_y)|$.

Then, $P(\pi) < P(\pi')$.

Corollary 1. *Consider a solution* $\pi = \{A_1, \ldots, A_m\}$ *of an instance of the VMA problem with total load* $\ell(D)$, *such that exactly* k *of the* A_x *sets,* $x \in [1, m]$, *are non-empty (hence it uses* k *PMs). Then, the power consumption is lower bounded by the power of the (maybe unfeasible) solution that balances the load evenly, i.e.,* $P(\pi) \geq kb + k(\ell(D)/k)^\alpha$.

3 Offline Analysis

3.1 NP-Hardness

As was mentioned, it can be shown that deciding whether there is a feasible solution for an instance of the (C, m)-VMA problem is NP-complete or not, by a direct reduction from the 3-Partition problem. However, this result does not apply directly to the $(C, \cdot)$-VMA, $(\cdot, m)$-VMA, and $(\cdot, \cdot)$-VMA problems. We show now that these problems are NP-hard. We first prove the following lemma.

Lemma 3. *Given an instance of the VMA problem, any solution* $\pi = \{A_1, \ldots, A_m\}$ *where* $\ell(A_i) \neq x^*$ *for some* $i \in [1, m] : A_i \neq \emptyset$, *has power consumption* $P(\pi) > \rho^* \ell(D) = \rho^* \sum_{d \in D} \ell(d)$.

We show now in the following theorem that the different versions of the (C, m)-VMA problem with unbounded C or m are NP-hard.

Theorem 1. *The* $(C,\cdot)$*-VMA,* $(\cdot,m)$*-VMA and* $(\cdot,\cdot)$*-VMA problems are strongly NP-hard, even if* α *is constant.*

It is known that strongly NP-hard problems cannot have a fully polynomial-time approximation scheme (FPTAS) [37]. Hence, the following corollary.

Corollary 2. *The* $(C,\cdot)$*-VMA,* $(\cdot,m)$*-VMA and* $(\cdot,\cdot)$*-VMA problems do not have fully polynomial-time approximation schemes (FPTAS), even if* α *is constant.*

In the following sections we show that, while the $(\cdot,m)$-VMA and $(\cdot,\cdot)$-VMA problems have polynomial-time approximation schemes (PTAS), the $(C,\cdot)$-VMA problem cannot be approximated below $\frac{3}{2}\cdot\frac{\alpha-1+(2/3)^\alpha}{\alpha}$.

3.2 The $(\cdot,m)$-VMA and $(\cdot,\cdot)$-VMA Problems have PTAS

We have proved that the $(\cdot,m)$-VMA and $(\cdot,\cdot)$-VMA problems are NP-hard in the strong sense and that, hence, there exists no FPTAS for them. However, Alon et al. [2], proved that if a function $f(\cdot)$ satisfies a condition denoted $F*$, then the problem of scheduling jobs in m identical machines so that $\sum_i f(M_i)$ is minimized has a PTAS, where M_i is the load of the jobs allocated to machine i. This result implies that if our function $f(\cdot)$ satisfies condition $F*$, the same PTAS can be used for the $(\cdot,m)$-VMA and $(\cdot,\cdot)$-VMA problems. From Observation 6.1 in [20], it can be derived that, in fact, our power consumption function $f(\cdot)$ satisfies condition $F*$. Hence, the following theorem.

Theorem 2. *There are polynomial-time approximation schemes (PTAS) for the* $(\cdot,m)$*-VMA and* $(\cdot,\cdot)$*-VMA problems.*

3.3 Bounds on the Approximability of the $(C,\cdot)$-VMA Problem

We study now the $(C,\cdot)$-VMA problem, where we consider an unbounded number of machines with bounded capacity C. We will provide a lower bound on its approximation ratio, independently on the relation between x^* and C; and upper bounds for the cases when $x^* \geq C$ and $x^* < C$.

Lower Bound on the Approximation Ratio. The following theorem shows a lower bound on the approximation ratio of any offline algorithm for $(C,\cdot)$-VMA.

Theorem 3. *No algorithm achieves an approximation ratio smaller than* $\frac{3}{2}\cdot\frac{\alpha-1+(\frac{2}{3})^\alpha}{\alpha}$ *for the* $(C,\cdot)$*-VMA problem unless* $\mathrm{P}=\mathrm{NP}$.

Upper Bound on the Approximation Ratio for $x^* \geq C$. We study now an upper bound on the competitive ratio of the $(C,\cdot)$-VMA problem for the case when $x^* \geq C$. Under this condition, the best is to load each PM to its full capacity. Intuitively, an optimal solution should load every machine up to its maximum capacity or, if not possible, should balance the load among PMs to maximize the average load. The following lemma formalizes this observation.

Lemma 4. *For any system with unbounded number of PMs where $x^* \geq C$ the power consumption of the optimal assignment π^* is lower bounded by the power consumption of a (possibly not feasible) solution where $\ell(D)$ is evenly distributed among $\overline{m}$ PMs, where $\overline{m}$ is the minimum number of PMs required to allocate all VMs (i.e., the optimal solution of the packing problem). That is, $P(\pi^*) \geq \overline{m} \cdot b + \overline{m}(\ell(D)/\overline{m})^\alpha$.*

Now we prove an upper bound on the approximation ratio showing a reduction to bin packing [22]. The reduction works as follows. Let each PM be seen as a bin of capacity C, and each VM be seen as an object to be placed in the bins, whose size is the VM load. Then, a solution for this bin packing problem instance yields a feasible (perhaps suboptimal) solution for the instance of $(C, \cdot)$-VMA. Moreover, using any bin-packing approximation algorithm, we obtain a feasible solution for $(C, \cdot)$-VMA that approximates the minimal number of PMs used. The power consumption of this solution approximates the power consumption of the optimal solution π^* of the instance of $(C, \cdot)$-VMA. In order to compute an upper bound on the approximation ratio of this algorithm, we will compare the power consumption of such solution against a lower bound on the power consumption of π^*. The following theorem shows the approximation ratio obtained.

Theorem 4. *For every $\epsilon > 0$, there exists an approximation algorithm for the $(C, \cdot)$-VMA problem when $x^* \geq C$ that achieves an approximation ratio of*

$$\rho < 1 + \epsilon + \frac{C^\alpha}{b} + \frac{1}{\overline{m}},$$

where $\overline{m}$ is the minimum number of PMs required to allocate all the VMs.

Upper Bound on the Approximation Ratio for $x^* < C$. We study now the $(C, \cdot)$-VMA problem when $x^* < C$. In this case, the optimal load per PM is less than its capacity, so an optimal solution would load every PM to x^* if possible, or try to balance the load close to x^*. In this case we slightly modify the bin packing algorithm described above, reducing the bin size from C to x^*. Then, using an approximation algorithm for this bin packing problem, the following theorem can be shown.

Theorem 5. *For every $\epsilon > 0$, there exists an approximation algorithm for the $(C, \cdot)$-VMA problem when $x^* < C$ that achieves an approximation ratio of*

$$\rho < \frac{\overline{m}}{m^*}\left((1+\epsilon) + \frac{1}{\alpha - 1}\right) + \frac{1}{m^*},$$

where m^* is the number of PMs used by the optimal solution of $(C, \cdot)$-VMA, and $\overline{m}$ is the minimum number of PMs required to allocate all the VMs without exceeding load x^* (i.e., the optimal solution of the bin packing problem).

4 Online Analysis

In this section, we study the online version of the VMA problem, i.e., when the VMs are revealed one by one. We first study lower bounds and then provide online algorithms and prove upper bounds on their competitive ratio.

4.1 Lower Bounds

In this section, we compute lower bounds on the competitive ratio for $(\cdot,\cdot)$-VMA, $(C,\cdot)$-VMA, $(\cdot,m)$-VMA, (C,m)-VMA and $(\cdot,2)$-VMA problems. We start with one general construction that is used to obtain lower bounds on the first four cases. Then, we develop special constructions for $(\cdot,m)$-VMA and $(\cdot,2)$-VMA that improve the lower bounds for these two problems.

General Construction. We prove lower bounds on the competitive ratio of $(\cdot,\cdot)$-VMA, $(C,\cdot)$-VMA, $(\cdot,m)$-VMA and (C,m)-VMA problems. These lower bounds are shown in the following two theorems. In Theorem 6, we prove a lower bound on the competitive ratio that is valid in the cases when C is unbounded and when it is larger or equal than x^*. The case $C \leq x^*$ is covered in Theorem 7.

Theorem 6. *There exists an instance of problems* $(\cdot,\cdot)$*-VMA,* $(\cdot,m)$*-VMA,* $(C,\cdot)$*-VMA and* (C,m)*-VMA when* $C > x^*$*, such that no online algorithm can guarantee a competitive ratio smaller than* $\frac{(3/2)2^\alpha-1}{2^\alpha-1}$.

Theorem 7. *There exists an instance of problems* $(C,\cdot)$*-VMA and* (C,m)*-VMA when* $C \leq x^*$ *such that no online algorithm can guarantee a competitive ratio smaller than* $(C^\alpha+2b)/(b+\max(C^\alpha, 2(C/2)^\alpha+b))$.

Special Constructions for $(\cdot,m)$-VMA and $(\cdot,2)$-VMA. We show first that for m PMs there is a lower bound on the competitive ratio that improves the previous lower bound when $\alpha > 4.5$. Secondly, we prove a particular lower bound for problem $(\cdot,2)$-VMA, that improves the previous lower bound when $\alpha > 3$.

Theorem 8. *There exists an instance of problem* $(\cdot,m)$*-VMA such that no online algorithm can guarantee a competitive ratio smaller than* $3^\alpha/(2^{\alpha+2}+\epsilon)$ *for any* $\epsilon > 0$.

Now, we show a stronger lower bound on the competitive ratio for $(\cdot,2)$-VMA problem.

Theorem 9. *There exists an instance of problem* $(\cdot,2)$*-VMA such that no online algorithm can guarantee a competitive ratio smaller than* $3^\alpha/2^{\alpha+1}$.

4.2 Upper Bounds

Now, we study upper bounds for $(\cdot,\cdot)$-VMA, $(C,\cdot)$-VMA, and $(\cdot,2)$-VMA problems. We start giving an online VMA algorithm that can be used in $(\cdot,\cdot)$-VMA and $(C,\cdot)$-VMA problems. The algorithm uses the load of the new revealed VM in order to decide the PM where it will be assigned. If the load of the revealed VM is strictly larger than $\min\{x^*,C\}/2$, the algorithm assigns this VM to a new PM without any other VM already assigned to it. Otherwise, the algorithm schedules the revealed VM to any loaded PM whose current load is smaller or equal than $\frac{\min\{x^*,C\}}{2}$. Hence, when this new VM is assigned, the load of this PM remains smaller than $\min\{x^*,C\}$. If there is no such loaded PM, the revealed VM is assigned to a new PM. Note that, since the case under consideration assumes the existence of an unbounded number of PMs, there exists always one new PM. A detailed description of this algorithm is shown in Algorithm 1. As before, A_j denotes the set of VMs assigned to PM s_j at a given time.

Algorithm 1. Online algorithm for $(\cdot,\cdot)$-VMA and $(C,\cdot)$-VMA problems.

for *each VM* d_i **do**
  if $\ell(d_i) > \frac{\min\{x^*,C\}}{2}$ **then**
    d_i is assigned to a new PM
  else
    d_i is assigned to any loaded PM s_j where $\ell(A_j) \leq \frac{\min\{x^*,C\}}{2}$. If such loaded PM does not exist, d_i is assigned to a new PM

We prove the approximation ratio of Algorithm 1 in the following two theorems.

Theorem 10. *There exists an online algorithm for $(\cdot,\cdot)$-VMA and $(C,\cdot)$-VMA when $x^* < C$ that achieves the following competitive ratio:*

$$\rho = 1, \text{ if no VM } d_i \text{ has load such that } \ell(d_i) < x^*,$$
$$\rho \leq \left(1 - \frac{1}{\alpha}\left(1 - \frac{1}{2^\alpha}\right)\right)\left(2 + \frac{x^*}{\ell(D_s)}\right), \text{ otherwise.}$$

Theorem 11. *There exists an online algorithm for $(C,\cdot)$-VMA when $x^* \geq C$ that achieves competitive ratio $\rho \leq \frac{2b}{C}\left(1 + \frac{1}{(\alpha-1)2^\alpha}\right)\left(2 + \frac{C}{\ell(D)}\right)$.*

Proof. We proceed with the analysis of the competitive ratio of Algorithm 1 in the case when $x^* \geq C$. The analysis uses the same technique used in the proof for the previous theorem. Hence, we just show the difference.

On one hand, when $x^* \geq C$, it holds that $f(\ell(A_i))/\ell(A_i) \geq f(C)/C$ due to the fact that $f(x)/x$ is monotone decreasing in interval $(0,C]$. It is also obvious that all the PMs will be loaded no more C. As a result, the optimal power consumption for $(C,\cdot)$-VMA can be bounded by $P(\pi^*) \geq f(C)\ell(D)/C$. On the other hand, the solution given by Algorithm 1 can also be upper bounded. We consider the following two cases.

Case 1: $\ell(\hat{A}_i) \geq C/2$ for all i. In this case, every PM will be loaded between $C/2$ and C. Consequently,

$$P(\pi) = \sum_{\frac{C}{2} \leq \ell(\hat{A}_i) \leq C} f(\ell(\hat{A}_i)) \leq \frac{f(\frac{C}{2})}{\frac{C}{2}} \ell(D).$$

The competitive ratio ρ then satisfies

$$\rho \leq \frac{\frac{f(\frac{C}{2})}{\frac{C}{2}} \ell(D)}{\frac{f(C)}{C} \ell(D)} = 2 \frac{f(\frac{C}{2})}{f(C)} \leq \frac{2b}{C} \left(1 + \frac{1}{(\alpha - 1)2^{\alpha}}\right).$$

Case 2: there exists s_i such that $\ell(\hat{A}_i) < C/2$. In this case, it holds:

$$\begin{aligned} P(\pi) &= \sum_{\frac{C}{2} \leq \ell(\hat{A}_i) \leq C} f(\ell(\hat{A}_i)) + f(\ell(\hat{A}_{s'})) \\ &\leq \frac{f(\frac{C}{2})}{\frac{C}{2}} \left(\sum_{d_i : \ell(d_i) \leq C} \ell(d_i) - \ell(\hat{A}_{s'}) \right) + f(\ell(\hat{A}_{s'})) \\ &= \frac{f(\frac{C}{2})}{\frac{C}{2}} \left(\ell(D) - \ell(\hat{A}_{s'}) \right) + \ell(\hat{A}_{s'})^{\alpha} + b. \end{aligned}$$

The competitive ratio ρ then satisfies

$$\begin{aligned} \rho &\leq \frac{P(\pi)}{\frac{f(C)}{C} \ell(D)} \leq \frac{2b}{C} \left(1 + \frac{1}{(\alpha - 1)2^{\alpha}}\right) + \frac{\ell(\hat{A}_{s'})^{\alpha} - \ell(\hat{A}_{s'}) \frac{f(\frac{C}{2})}{\frac{C}{2}} + b}{\frac{f(C)}{C} \ell(D)} \\ &\leq \frac{2b}{C} \left(1 + \frac{1}{(\alpha - 1)2^{\alpha}}\right) + \frac{\ell(\hat{A}_{s'})^{\alpha} + b}{\frac{f(C)}{C} \ell(D)} \\ &\leq \frac{2b}{C} \left(1 + \frac{1}{(\alpha - 1)2^{\alpha}}\right) + \frac{(\frac{C}{2})^{\alpha} + b}{\frac{f(C)}{C} \ell(D)} \\ &= \frac{2b}{C} \left(1 + \frac{1}{(\alpha - 1)2^{\alpha}}\right) \left(2 + \frac{C}{\ell(D)}\right). \end{aligned}$$

Upper Bounds for $(\cdot, 2)$-VMA Problem. We now present an algorithm (detailed in Algorithm 2) for $(\cdot, 2)$-VMA problem and show an upper bound on its competitive ratio. A_1 and A_2 are the sets of VMs assigned to PMs s_1 and s_2, respectively, at any given time.

Algorithm 2. Online algorithm for $(\cdot, 2)$-VMA.

for *each VM* d_i **do**
  if $\ell(d_i) + \ell(A_1) \leq (b/(2^\alpha - 2))^{1/\alpha}$ **or** $\ell(A_1) \leq \ell(A_2)$ **then**
    d_i is assigned to s_1;
  else
    d_i is assigned to s_2;

We prove the approximation ratio of Algorithm 2 in the following theorem.

Theorem 12. *There exists an online algorithm for* $(\cdot, 2)$*-VMA that achieves the following competitive ratios.*

$$\rho = 1, \ \textit{for } \ell(D) \leq \left(\tfrac{b}{2^\alpha - 2}\right)^{1/\alpha},$$

$$\rho \leq \max\left\{2, \left(\frac{3}{2}\right)^{\alpha-1}\right\}, \ \textit{for } \ell(D) > \left(\tfrac{b}{2^\alpha - 2}\right)^{1/\alpha}.$$

5 Discussion

We discuss in this section practical issues that must be addressed to apply our results to production environments.

Heterogeneity of Servers. For the sake of simplicity, we assume in our model that all servers in a data center are identical. We believe this reasonable, considering that modern data centers are usually built with homogeneous commodity hardware. Nevertheless, the proposed model and derived results are also amenable to heterogeneous data center environments. In a heterogeneous data center, servers can be categorized into several groups with identical servers in each group. Then, different types of applications can be assigned to server groups according to their resource requirements. The VMA model presented here can be applied to the assignment problem of allocating tasks from the designated types of applications (especially CPU-intensive ones) to each group of servers. The approximation results we derive in this paper can be then combined with server-group assignment approximation bounds (out of the scope of this paper) for energy-efficient task assignment in real data centers, regardless of the homogeneity of servers.

Consolidation. Traditionally, consolidation has been understood as a bin packing problem [31,39], where VMs are assigned to PMs attempting to minimize the number of active PMs. However, the results we derived in this paper, as well as the results in [7], show that such approach is not energy-efficient. Indeed, we showed that PM's should be loaded up to x^* to reduce energy consumption, even if this requires having more active PMs.

VM arrival and departure. When a new VM arrives to the system, or an assigned VM departs, adjustments to the assignment may improve energy efficiency. Given that the cost of VM migration is nowadays decreasing dramatically, our offline positive results can also be accommodated by reassigning VMs whenever the set of VM demands changes. Should the cost of migration be high to reassign after each VM arrival or departure, time could be divided in epochs buffering newly arrived VM demands until the beginning of the next epoch, when all (new and old) VMs would be reassigned (if necessary) running our offline approximation algorithm.

Multi-resource scheduling. This work focuses on CPU-intensive jobs (VMs) such as MapReduce-like tasks [19] which are representative in production datacenters. As the CPU is generally the dominant energy consumer in a server, assigning VMs according to CPU workloads entails energy efficiency. However, there exist types of jobs demanding heavily other computational resources, such as memory and/or storage. Although these resources have limited impact on a server's energy consumption, VMs performance may be degraded if they become the bottleneck resource in the system. In this case, a joint optimization of multiple resources (out of the scope of this paper) is necessary for VMA.

References

1. Alon, N., Azar, Y., Woeginger, G.J., Yadid, T.: Approximation schemes for scheduling. In: Saks, M.E. (ed.) SODA, pp. 493–500. ACM/SIAM (1997)
2. Alon, N., Azar, Y., Woeginger, G.J., Yadid, T.: Approximation schemes for scheduling on parallel machines. J. Sched. **1**(1), 55–66 (1998)
3. Amazon. Amazon web services. http://aws.amazon.com. Accessed 27 August 2012
4. Andrews, M., Fernández Anta, A., Zhang, L., Zhao, W.: Routing for power minimization in the speed scaling model. IEEE/ACM Trans. Netw. **20**(1), 285–294 (2012)
5. Andrews, M., Antonakopoulos, S., Zhang, L.: Minimum-cost network design with (dis)economies of scale. In: Proceedings of 51-st Annual IEEE Symposium on Foundations of Computer Science, pp. 585–592 (2010)
6. Antoniadis, A., Im, S., Krishnaswamy, R., Moseley, B., Nagarajan, V., Pruhs, K., Stein, C.: Energy efficient circuit routing. In: SODA, Hallucination helps (2014)
7. Arjona Aroca, J., Chatzipapas, A., Fernández Anta, A., Mancuso, V.: A measurement-based analysis of the energy consumption of data center servers. In: e-Energy. ACM (2014)
8. Arjona Aroca, J., Fernández Anta, A., Mosteiro, M.A., Thraves, C., Wang, L.: Power-efficient assignment of virtual machines to physical machines (2013). arXiv:1304.7121v2 [cs.DS]. http://arxiv.org/abs/1304.7121
9. Bansal, N., Chan, H.-L., Pruhs, K.: Speed scaling with an arbitrary power function. In: Proceedings of 20-th Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 693–701 (2009)
10. Bansal, N., Gupta, A., Krishnaswamy, R., Nagarajan, V., Pruhs, K., Stein, C.: Multicast routing for energy minimization using speed scaling. In: Even, G., Rawitz, D. (eds.) MedAlg 2012. LNCS, vol. 7659, pp. 37–51. Springer, Heidelberg (2012)
11. Bellur, U., Rao, C.S., Madhu Kumar, SD.: Optimal placement algorithms for virtual machines (2010). arXiv:1011.5064 (http://arxiv.org/abs/1011.5064)

12. Botero, J.F., Hesselbach, X., Duelli, M., Schlosser, D., Fischer, A., de Meer, H.: Energy efficient virtual network embedding. IEEE Commun. Lett. **16**(5), 756–759 (2012)
13. Cardosa, M., Singh, A., Pucha, H., Chandra, A.: Exploiting spatio-temporal tradeoffs for energy-aware mapreduce in the cloud. In: 2011 IEEE International Conference on Cloud Computing (CLOUD), pp. 251–258 (2011)
14. Chakrabarty, D., Chekuri, C., Khanna, S., Korula, N.: Approximability of capacitated network design. In: Günlük, O., Woeginger, G.J. (eds.) IPCO 2011. LNCS, vol. 6655, pp. 78–91. Springer, Heidelberg (2011)
15. Chandra, K.A., Wong, C.K.: Worst-case analysis of a placement algorithm related to storage allocation. SIAM J. Comput. **4**(3), 249–263 (1975)
16. Chen, S.-C., Lee, C.-C., Chang, H.-Y., Lai, K.-C., Li, K.-C., Rong, C.: Energy-aware task consolidation technique for cloud computing. In: Proceedings of the IEEE Third International Conference on Cloud Computing Technology and Science, pp. 115–121 (2011)
17. Citrix. Citrix. http://www.citrix.com. Accessed 27 August 2012
18. Cody, R.A., Coffman Jr, E.G.: Record allocation for minimizing expected retrieval costs on drum-like storage devices. J. ACM **23**(1), 103–115 (1976)
19. Dean, J., Ghemawat, S.: Mapreduce: simplified data processing on large clusters. Commun. ACM **51**(1), 107–113 (2008)
20. Epstein, L., Sgall, J.: Approximation schemes for scheduling on uniformly related and identical parallel machines. Algorithmica **39**(1), 43–57 (2004)
21. Fernandez de la Vega, W., Lueker, G.S.: Bin packing can be solved within $1 + \epsilon$ in linear time. Combinatorica **1**(4), 349–355 (1981)
22. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co., New York (1979)
23. Gupta, A., Krishnaswamy, R., Pruhs, K.: Online primal-dual for non-linear optimization with applications to speed scaling. In: Erlebach, T., Persiano, G. (eds.) WAOA 2012. LNCS, vol. 7846, pp. 173–186. Springer, Heidelberg (2013)
24. Im, S., Moseley, B., Pruhs, K.: Online scheduling with general cost functions. In: Proceedings of 23-rd Annual ACM-SIAM Symposium on Discrete Algorithms, pp. 1254–1265 (2012)
25. Jansen, R., Brenner, P.R.: Energy efficient virtual machine allocation in the cloud. In: 2011 International Green Computing Conference and Workshops (IGCC), pp. 1–8 (2011)
26. Kusic, D., Kephart, J.O., Hanson, J.E., Kandasamy, N., Jiang, G.: Power and performance management of virtualized computing environments via lookahead control. Cluster Comput. **12**(1), 1–15 (2009)
27. Liu, N., Dong, Z., Rojas-Cessa, R.: Task and server assignment for reduction of energy consumption in datacenters. In: Proceedings of the IEEE 11-th International Symposium on Network Computing and Applications, pp. 171–174 (2012)
28. Machida, F., Kawato, M., Maeno, Y.: Redundant virtual machine placement for fault-tolerant consolidated server clusters. In: 2010 IEEE Network Operations and Management Symposium (NOMS), pp. 32–39 (2010)
29. Mark, C.C.T., Niyato, D., Chen-Khong, T.: Evolutionary optimal virtual machine placement and demand forecaster for cloud computing. In: 2011 IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 348–355 (2011)
30. Mills, K., Filliben, J., Dabrowski, C.: Comparing vm-placement algorithms for on-demand clouds. In: Proceedings of the IEEE Third International Conference on Cloud Computing Technology and Science, pp. 91–98 (2011)

31. Mishra, M., Sahoo, A.: On theory of vm placement: anomalies in existing methodologies and their mitigation using a novel vector based approach. In: 2011 IEEE International Conference on Cloud Computing (CLOUD), pp. 275–282 (2011)
32. Nathuji, R., Schwan, K.: Virtualpower: coordinated power management in virtualized enterprise systems. In: SOSP, pp. 265–278 (2007)
33. Van Nguyen, H., Tran, F.D., Menaud, J.-M.: Autonomic virtual resource management for service hosting platforms. In: Proceedings of the 2009 ICSE Workshop on Software Engineering Challenges of Cloud Computing, CLOUD '09, pp. 1–8. IEEE Computer Society (2009)
34. Rackspace. Rackspace. http://www.rackspace.com. Accessed 27 August 2012
35. Srikantaiah, S., Kansal, A., Zhao, F.: Energy aware consolidation for cloud computing. In: Proceedings of the 2008 Conference on Power Aware Computing and Systems, HotPower'08, p. 10. USENIX Association (2008)
36. Van den Bossche, R., Vanmechelen, K., Broeckhove, J.: Cost-efficient scheduling heuristics for deadline constrained workloads on hybrid clouds. In: Proceedings of the IEEE Third International Conference on Cloud Computing Technology and Science, pp. 320–327 (2011)
37. Vazirani, V.V.: Approximation Algorithms. Springer, Heidelberg (2004)
38. Viswanathan, H., Lee, E.K., Rodero, I., Pompili, D., Parashar, M., Gamell, M.: Energy-aware application-centric vm allocation for HPC workloads. In: 2011 IEEE International Symposium on Parallel and Distributed Processing Workshops and Phd Forum (IPDPSW), pp. 890–897 (2011)
39. Wang, M., Meng, X., Zhang, L.: Consolidating virtual machines with dynamic bandwidth demand in data centers. In: IEEE INFOCOM, pp. 71–75 (2011)
40. Xu, J., Fortes, J.: A multi-objective approach to virtual machine management in datacenters. In: Proceedings of the 8th ACM International Conference on Autonomic Computing, ICAC '11, pp. 225–234. ACM (2011)