



Optimal energy-efficient placement of virtual machines with divisible sizes

Gergely Halácsy^a, Zoltán Ádám Mann^{b,*}

^a Budapest University of Technology and Economics, Budapest, Hungary

^b University of Duisburg-Essen, Essen, Germany



ARTICLE INFO

Article history:

Received 18 July 2016

Received in revised form 17 March 2018

Accepted 5 June 2018

Available online xxxx

Communicated by Gregory Chockler

Keywords:

Cloud computing

Virtual machines

Bin packing

Divisible item sizes

Analysis of algorithms

ABSTRACT

A key problem in the management of data centers is how to provision virtual machines based on the available physical machines, because an optimized placement can lead to significant reduction in energy consumption. This problem can be formulated as bin packing with heterogeneous bin types, where the cost of a bin depends on how full it is. We prove that under suitable conditions, an extended version of the First-Fit-Decreasing heuristic delivers optimal results for this problem.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Data centers serve an ever-growing demand for computational power. The resulting growth in the energy consumption of data centers has both huge environmental impact and huge costs [19]. Therefore, the energy-efficient management of data centers has received much attention in the last couple of years.

Hardware vendors have devised techniques to reduce the power consumption of resources that are idle or lightly loaded [2]. As a result, the power consumption of a server depends significantly on its load. For example, the power consumption of a HP ProLiant DL380 G7 server varies from 280W (zero load) to 540W (full load) [10]. It is important to take into account the power characteristics (i.e., how power consumption depends on the server's load) when allocating workload on the servers. Most previous research assumed linear power characteristics [24], but some used more sophisticated power models, e.g., Hsu and

Poole found the function $P(u) = P_{idle} + (P_{peak} - P_{idle}) \cdot u^{0.75}$ to yield the most accurate results [11].

A key technology for enabling effective data center management is virtualization. In a virtualized data center, the applications are deployed in virtual machines (VMs), which are in turn deployed on physical machines (PMs). This way, multiple applications can co-exist on the same PM in an isolated manner. Through virtualization, the workload can be allocated on a few highly utilized PMs and other PMs can be turned off, resulting in significant savings in energy consumption. This leads to an important optimization problem called the *VM allocation problem*: given the capacity and power consumption characteristics of the PMs and the load of the VMs, how to map the VMs on the PMs so that the overall energy consumption is minimal.

Several slightly different versions of the VM allocation problem have been addressed [16]. In many cases, the objective was to minimize the number of turned-on PMs as a proxy for energy consumption [3,20,22,23]. This is only an approximation though since the power consumption of

* Corresponding author.

turned-on PMs can vary heavily. And with the advancement of technology, the dynamic power range (i.e., the difference between maximum and idle power consumption) of PMs is increasing.

In terms of the proposed algorithmic techniques, some works suggested exact methods but the majority applied heuristics. The proposed exact methods rely almost always on some form of mathematic programming (e.g., integer linear programming) and off-the-shelf solvers [9,20]. Unfortunately, these approaches do not scale to practical problem sizes.

There is a natural connection between the VM allocation problem and bin-packing: VMs with their loads can be seen as items with given sizes that need to be packed into bins (PMs) of given capacity. On one hand, this proves that VM allocation is NP-hard. On the other hand, several researchers suggested to adopt bin-packing heuristics like First-Fit, Best-Fit, First-Fit-Decreasing etc. to the VM allocation problem [4,14,25]. However, VM placement is in several ways more complex than bin-packing, so that the known approximation results concerning these algorithms on bin-packing [6,7] cannot be transferred to VM placement; in other words, they remain heuristics the results of which can be arbitrarily far from the optimum [15,17,18].

Some relevant generalizations of bin-packing have also been considered, mainly from an approximation point of view. Epstein and Levin showed an asymptotic polynomial-time approximation scheme (APTAS) for the problem in which different bin types are considered, each bin type is associated with a capacity and a cost, and the aim is to pack the items into a set of bins with minimum total cost, assuming that a sufficient number of bins are available from each type [8]. The problem that we are addressing now is more complex in the sense that the cost of a bin (its energy consumption) is not constant, but depends on how full it is. If each bin's cost depends linearly on how full it is, and this linear function is the same for each bin, then it is easy to see that the problem admits an APTAS [17]. However, the problem in which the cost of different bins can be described by different and not necessarily linear functions (which is a more realistic model of the VM allocation problem), has to our knowledge not been addressed yet.

The hardness of bin-packing arises partly because arbitrary sizes can appear in the input. Constraints on the allowed sizes can make the problem much easier. If the item sizes form a *divisible series* – i.e., for each pair of different item sizes, the smaller is a divisor of the larger – bin-packing can be solved optimally in polynomial time [5]. In this paper, we use a similar approach to devise a polynomial-time exact algorithm for the generalized problem in which the cost of bins depends on how full they are, under the assumption that the item sizes form a divisible series. This assumption is not too strict for VM allocation, because PM capacities and VM sizes are often powers of 2 [21], leading to divisible item size series.

The contributions of this paper are as follows:

- We address a version of the VM allocation problem, aiming to minimize total power consumption, where the power consumption of each type of PM is given

by some function of its load. In contrast to previous works, we make only very light assumptions on these functions: they must be monotonously increasing and concave. These assumptions hold for example for the linear power consumption characteristics used by several researchers [1,24] and also for the more sophisticated power consumption characteristics proposed by Hsu and Poole [11]. Further, we assume that VM sizes form a divisible series. This assumption holds for example if the VM sizes are all powers of two, which occurs frequently in practice, as reported by Shen et al. [21].

- We devise an algorithm called OptDiv for this problem. OptDiv is based on the First-Fit-Decreasing (FFD) heuristic, but FFD is oblivious of bin costs, so OptDiv extends it with informed decisions relating to power consumption.
- We prove that, under the given assumptions, OptDiv results in optimal total power consumption.
- OptDiv is very fast, in contrast to the proposed APTASs that are often not practical because of the huge constants in their execution times [13].

2. Preliminaries

In the *bin-packing* problem, we are given a set of n items with sizes $s_1 \leq s_2 \leq \dots \leq s_n$ and a sufficiently large set of bins, each having capacity C , where $s_n \leq C$. The aim is to pack all the items into a minimal number m of bins B_1, \dots, B_m so that for each B_i , the sum of the sizes of the items in B_i is at most C . B_i denotes both the i th bin and the set of items in that bin.

The items arrive in some order s_{i_1}, \dots, s_{i_n} . The *First-Fit* heuristic (FF) processes the items in this order: it starts by opening a bin B_1 for item s_{i_1} . In a general step, when bins B_1, \dots, B_k are already in use and the next item s_{i_j} is considered, the algorithm puts s_{i_j} into the first bin from B_1, \dots, B_k where it fits if there is such a bin, otherwise it opens a new bin B_{k+1} . The *First-Fit-Decreasing* heuristic (FFD) consists of first sorting the items in non-increasing order of their size, and then performing FF in this order of the items. Both FF and FFD are known to be approximation algorithms for the bin-packing problem [6,7].

Let x, y be arbitrary real numbers. We call x a divisor of y and y a multiple of x if there exists an integer z such that $x \cdot z = y$. This relation is denoted as $x \mid y$. The input of bin-packing is *weakly divisible* if $s_i \mid s_j$ holds for all $1 \leq i < j \leq n$. If, in addition, also $s_n \mid C$ holds, then the input is *strongly divisible*.

The following known results about divisible inputs to bin-packing will be important.

Lemma 1 ([5], Fact 2). *Let s_i be an item size in a weakly divisible input and let T be a subset of the items such that no item in T has size larger than s_i but the sum of the sizes of the items in T is at least s_i . Then, there is a subset $T' \subseteq T$ such that the sum of the sizes of the items in T' equals s_i .*

Theorem 2 ([5], Theorem 2). *For weakly divisible inputs, the FFD algorithm always uses the minimal number of bins.*

Download English Version:

<https://daneshyari.com/en/article/6874131>

Download Persian Version:

<https://daneshyari.com/article/6874131>

[Daneshyari.com](https://daneshyari.com)