Rigorous results on the effectiveness of some heuristics for the consolidation of virtual machines in a cloud data center[*]

Zoltán Ádám Mann

Department of Computer Science and Information Theory,
Budapest University of Technology and Economics,
Hungary

Abstract

Dynamic consolidation of virtual machines (VMs) in a cloud data center can be used to minimize power consumption. Beloglazov et al. have proposed the MM (Minimization of Migrations) heuristic for selecting the VMs to migrate from under- or over-utilized hosts, as well as the MBFD (Modified Best Fit Decreasing) heuristic for deciding the placement of the migrated VMs. According to their simulation results, these heuristics work very well in practice. In this paper, we investigate what performance guarantees can be rigorously proven for the heuristics. In particular, we establish that MM is optimal with respect to the number of selected VMs of an over-utilized host and it is a 1.5-approximation with respect to the decrease in utilization. On the other hand, we show that the result of MBFD can be arbitrarily far from the optimum. Moreover, we show that even if both MM and MBFD deliver optimal results, their combination does not necessarily result in optimal VM consolidation, but approximation results can be proven under suitable technical conditions. To the best of our knowledge, these are the first rigorously proven results on the effectiveness of also practically useful heuristic algorithms for the VM consolidation problem.

Keywords: Cloud computing, Virtual machines, VM consolidation, Approximation algorithms

1 Introduction and previous work

In recent years, the increasing adoption of cloud computing has transformed the IT industry [1]. Large, virtualized data centers are serving the ever growing demand for computation, storage, and networking. Because of these trends, the efficient operation of data centers is increasingly important. One of the main concerns is energy consumption, because of both its costs and its environmental impact. According to a recent study, data center energy consumption is the fastest growing part of the energy consumption of the ICT ecosystem; moreover, the initial cost of purchasing the equipment for a data center is already outweighed by the cost of its ongoing electricity consumption [2].

An attractive option for saving energy in data centers is to consolidate the virtual machines (VMs) to the minimal number of physical hosts and switching the unused hosts off or at least to a less power-hungry mode of operation (e.g., sleep mode). However, too aggressive VM consolidation can lead to overloaded hosts with negative effects on the delivered quality of service (QoS), thus potentially violating the service level agreements (SLA) with the customers. Hence, VM consolidation must find the optimal balance between QoS and energy consumption [3, 4].

In their recent works, Beloglazov, Buyya and Abawajy proposed a combination of two heuristics for near-optimal VM consolidation [5, 6]. The first heuristic, called MM (Minimization of Migrations), selects the VMs that should be migrated from a given host. For this purpose, two thresholds are given: a lower and an upper threshold. If the utilization of a host drops below the lower threshold, then all VMs residing on that host should be removed so that the host can be switched off in order to save energy. If the utilization of

[*] This paper was published in Future Generation Computer Systems, volume 51, pages 1-6, 2015

the host is higher than the upper threshold, then some of the VMs residing on the host should be removed in order to avoid SLA violations. The MM heuristic selects the minimum number of VMs necessary to decrease the utilization below the upper threshold.

The other heuristic, called MBFD (Modified Best Fit Decreasing), addresses the allocation of VMs to hosts. This can be used for two purposes: (i) to accommodate customer requests for new VMs and (ii) to find a new allocation for the VMs that should be migrated from under- or over-utilized hosts. This problem is similar to the much-studied bin-packing problem, for which simple greedy algorithms like First Fit (FF), First Fit Decreasing (FFD), Best Fit (BF), and Best Fit Decreasing (BFD) perform well and even have rigorously proven worst-case approximation ratios [7, 8, 9, 10]. Accordingly, MBFD is also a greedy heuristic that iterates through the list of VMs in decreasing order of load and allocates each VM to the most power-efficient host that has sufficient free capacity to accommodate it.

Beloglazov et al. demonstrated with substantial empirical evidence that MM and MBFD perform well in practice and outperform other competing heuristics [5, 6]. In this paper, our aim is to investigate whether any performance guarantees can be established rigorously for these heuristics, either in terms of optimality or approximation ratio.

The novelty of our approach lies in the rigorous analysis of worst-case effectiveness. Most previous works on the optimization of VM provisioning used heuristics and showed their effectiveness by means of simulations or other empirical techniques [11, 12, 13, 14, 15, 16]. The drawback of such approaches is that, even if the proposed heuristics yield good results in the specific evaluation environment, there is no guarantee whatsoever that they will work similarly well under other circumstances (e.g., with other types of hosts and VMs, other workload characteristics etc.).

For example, Verma et al. compared four different heuristics using server trace data from a production data center [17]. From their plots it can be seen that there can be huge differences between the quality of the results found by those algorithms: in some cases, the placement delivered by the worst-performing algorithm consumes five times more power than the placement found by the best-performing algorithm. Concerning the number of SLA violations, the differences are sometimes even bigger (an order of magnitude or even more).

Another conclusion that can be drawn from the empirical results of that paper is that heuristics tend to have some critical parameters, the tuning of which may also result in large differences in algorithm effectiveness. For instance, their CBP heuristic has a so-called "correlation cutoff parameter"; different settings of this parameter may lead to power consumption values that are up to a factor of 2.5 apart. This may be a problem if workload characteristics are unknown – as is frequently the case for public Infrastructure-as-a-Service providers – because setting such parameters wrongly can lead to substantial degradation of algorithm effectiveness. Similar conclusions can be drawn from the results presented by Tomás and Tordsson, who showed the effect of data center overbooking on resource utilization and application response time [18]: beyond a – workload-dependent – overbooking threshold, application response time abruptly increases. As a consequence, if the target overbooking rate is wrongly selected, this may lead to severe SLA violations.

For these reasons, we believe that using heuristics without any performance guarantee is very dangerous for VM placement in practice.

There have also been some attempts to solve the VM consolidation problem optimally, by formulating it as a mathematical optimization problem, and solving it using off-the-shelf solvers. Such approaches included integer linear programming [19, 20], pseudo-Boolean optimization [21], mixed integer non-linear programming [22] and binary integer programming [23]. With these solutions, the above problems are non-existent since the results are guaranteed to be optimal. However, all of these approaches suffer from a scalability problem that renders them unusable in practice: the runtime becomes prohibitively large for instances of even moderate size.

On the other hand, exact solutions also shed some light on the effectiveness of heuristics. A perfect illustration is given by Ribas et al. [21]. They compare a pseudo-Boolean formulation using two exact pseudo-Boolean solvers (SAT4j and Bsolo) with two heuristics (Round-Robin and First-Fit). The bigger benchmarks can be solved only by the heuristics, because the pseudo-Boolean solvers time out. However, on instances that are within the reach of the exact methods, it can be observed that the heuristics' results are sometimes very far from the optimum: in extreme cases, the cost of the result of the First-Fit heuristic is three times as high as the optimum; for Round-Robin, this ratio is even worse.

Therefore, we believe that none of the approaches presented so far in the literature are completely satisfactory: they are either fast but unreliable heuristics without any guarantee on their effectiveness, or they are exact methods that deliver optimal results but require exorbitantly long runtimes. In this paper, we suggest a new way which seems to be a good compromise: to investigate whether formal performance guarantees can be proven for practically usable heuristics. This way we can have the best of both worlds: fast execution *and* formal guarantees that the heuristics will be fairly effective even for unknown workloads or for suboptimal parameter values. We make a first step in this direction by analyzing a pair of heuristics that have been empirically found to be useful. Our results demonstrate that it is indeed possible to prove bounds on the effectiveness of practical heuristics, thus guaranteeing that they will work well in any situation. Furthermore, our analysis also makes the conditions explicit under which these results hold, thus pinpointing the limitations of the heuristics and giving insight for future work on the design of improved algorithms.

For some related problems, there has been some work on approximation algorithms. Breitgand and Epstein presented a 2-approximation algorithm for the stochastic bin packing problem under the assumption of independent normally distributed random variables [24]. Alicherry and Lakshman derived some approximation algorithms and inapproximability results for the problem of minimizing the cost of communication among VMs [25, 26]. Breitgand et al. devised algorithms for profit optimization in a federated cloud and proved that, under certain conditions, the algorithm for one of the sub-problems, which is a greedy LP-rounding procedure, ensures 2-approximation [27]. However, none of these algorithms have been proven to work well in practice. Our approach is different: we analyze algorithms that we know are practically useful.

Section 2 of the paper is devoted to the MM heuristic, Section 3 to the MBFD heuristic. In both cases, we first describe the algorithms themselves, and then analyze their effectiveness. Section 4 is about the interplay of the two heuristics. Finally, Section 5 concludes the paper.

2 Analysis of the MM heuristic

We are given a host with capacity $C > 0$. There are k VMs currently allocated to this host with utilizations $0 < v_1, v_2, \ldots, v_k$. Obviously,

$$S := \sum_{i=1}^{k} v_i \leq C$$

must hold. The host is considered overloaded if the total utilization is higher than a given threshold, defined as a percentage τ of the total capacity ($0 < \tau < 1$). That is, the host is overloaded if $S > \tau C$.

Let $V = \{1, 2, \ldots, k\}$ denote the set of VMs currently allocated to the overloaded host. The objective is to select a subset of V with minimum cardinality, such that, after removing these VMs from the host, it will not be overloaded anymore. A subset $V' \subseteq V$ will be called *relieving* if

$$\sum_{i \in V \setminus V'} v_i \leq \tau C,$$

or equivalently,

$$\sum_{i \in V'} v_i \geq S - \tau C.$$

The objective is to find a relieving set $V' \subseteq V$ with minimum $|V'|$. Minimizing $|V'|$ is indeed useful because of the overhead associated with the migration of a VM from one host to another.

The MM heuristic considers the VMs in decreasing order of utilization. Without loss of generality, we can assume that $v_1 \geq v_2 \geq \ldots \geq v_k$. Let $1 \leq \ell \leq k$ be such that

$$\sum_{i=1}^{\ell-1} v_i < S - \tau C, \text{ but } \sum_{i=1}^{\ell} v_i \geq S - \tau C.$$

This means that the first ℓ VMs build a relieving set, but the first $\ell - 1$ VMs would not be sufficient to be relieving. Obviously, there is a unique index ℓ with this property, and it can be found in linear time.

The output of the MM heuristic is $V' := \{1, \ldots, \ell-1\} \cup \{j\}$, where $j \in \{\ell, \ldots, k\}$ with the following properties: (i) V' is a relieving set and (ii) v_j is minimal among the possibilities. Again, it is clear that such a j exists and can be found in linear time. It is also obvious that the result is indeed a relieving set. However, Beloglazov et al. did not prove that it is of minimum size. However, this is true:

Theorem 1. *The relieving set V' returned by the MM heuristic has minimal cardinality among all relieving sets.*

Proof. Let us assume that $V'' \subseteq V$ is a relieving set with minimal cardinality and $t := |V''| < |V'|$. It follows that $t \leq \ell - 1$. Since V'' is a relieving set,

$$\sum_{i \in V''} v_i \geq S - \tau C.$$

Hence, there are $t \leq \ell - 1$ elements in V such that the sum of the corresponding v_i values is at least $S - \tau C$. On the other hand, since the v_i values are positive and are in decreasing order, the sum of the first $\ell - 1$ values cannot be lower than the sum of those t values. Hence,

$$\sum_{i=1}^{\ell-1} v_i \geq S - \tau C,$$

which contradicts the choice of ℓ. □

Although Beloglazov et al. did not state this explicitly, but it is clear that the MM heuristic has a secondary objective as well. If minimizing the number of selected VMs were the only objective, then it could simply select the first ℓ VMs. From the way the last VM (j) is selected, it is clear that the secondary objective is to minimize the total utilization of the selected VMs. This is a plausible goal since the selected VMs have to be migrated to other hosts; minimizing their total utilization makes it easier to find new hosts to accommodate them.

More precisely, the goal of MM can be formulated as follows: find a relieving set with minimal cardinality (primary goal) that has minimal total utilization among all relieving sets of minimal cardinality (secondary goal).

Hence the question arises whether the output of MM is optimal also with respect to the secondary goal. Unfortunately, this is not always case, as demonstrated by the following example.

Example 1. *Let us assume a host with $C = 10$, and let $\tau = 0.8$. The host currently accommodates 9 VMs, with the following utilizations:*

$$v_1 = 2 - \varepsilon,\ v_2 = v_3 = \ldots = v_9 = 1,$$

where ε is a small positive number. The host thus has a current utilization of almost 100% and so it is overloaded. An optimal relieving set consists of two VMs with utilization of 1 each – this is clearly optimal both with respect to the number of selected VMs and also with respect to their total utilization. On the other hand, the MM heuristic will select the VM with utilization $2 - \varepsilon$ and another one with utilization 1. Thus, the total utilization of the relieving set returned by MM is $3 - \varepsilon$. □

Since ε can be arbitrarily small, this construction shows that the performance of MM can be up to 50% off the optimum. The next theorem shows that this is the worst possible case (and the example shows that the following result is tight).

Theorem 2. *The total utilization of the relieving set V' returned by the MM heuristic is at most 3/2 times the optimum.*

Proof. As above, let $\ell = |V'|$. We must differentiate between three cases according to the value of ℓ.

Case 1: $\ell \geq 3$. V' consists of the $\ell - 1$ elements of V with the highest utilization, plus one more: $V' = \{1, \ldots, \ell-1, j\}$. Since $v_1 \geq \ldots \geq v_{\ell-1} \geq v_j$, it follows that

$$v_j \leq \frac{1}{\ell - 1} \cdot \sum_{i=1}^{\ell-1} v_i.$$

As a consequence, the total utilization of V' is

$$\sum_{i\in V'} v_i \leq \left(1+\frac{1}{\ell-1}\right)\cdot\sum_{i=1}^{\ell-1} v_i < \left(1+\frac{1}{\ell-1}\right)\cdot(S-\tau C), \tag{1}$$

where the last inequality uses the fact that the first $\ell-1$ VMs are not sufficient for a relieving set. On the other hand, any relieving set must have a total utilization of at least $S-\tau C$. Hence, (1) shows that the total utilization of V' is at most $\left(1+\frac{1}{\ell-1}\right)$ times the optimum, which is at most 3/2 for any $\ell \geq 3$.

Case 2: $\ell = 1$. In this case, the MM heuristic returns the VM with smallest utilization that is sufficient to relieve the host, i.e. with utilization at least $S-\tau C$. This is obviously the optimal choice.

Case 3: $\ell = 2$. In this case, $V' = \{1, j\}$. Let $V'' = \{x, y\}$ be a relieving set with cardinality two, for which the total utilization is minimal. If $x = 1$ or $y = 1$, then V'' cannot be better than V', which means that V' is also optimal; hence we assume that $x \neq 1$ and $y \neq 1$. Since V'' is a relieving set and $v_1 \geq v_x$, it follows that

$$S-\tau C \leq v_x+v_y \leq v_1+v_y.$$

This means that $\{1, y\}$ is also a relieving set. Since the MM algorithm chose j and not y, this means that $v_j \leq v_y$. It can be shown analogously that $v_j \leq v_x$. These two inequalities together lead to

$$v_j \leq \frac{1}{2}\cdot(v_x+v_y). \tag{2}$$

On the other hand, $\{1\}$ is not a relieving set but $\{x, y\}$ is, and thus $v_1 < v_x + v_y$. Together with (2), this leads to

$$v_1+v_j < v_x+v_y+\frac{1}{2}\cdot(v_x+v_y) = \frac{3}{2}\cdot(v_x+v_y),$$

which completes the proof since $v_x + v_y$ is the optimum. □

To sum up: the MM heuristic is guaranteed to deliver an optimal result with respect to its primary goal (minimization of the number of VMs to migrate) and it is a 3/2-approximation with respect to its secondary goal (minimization of the total utilization of the VMs to migrate).

3 Analysis of the MBFD heuristic

We are given n VMs with performance need $v_1, v_2, \ldots, v_n$. These VMs should be allocated to hosts, either because they represent new customer requests or because they have been selected for migration from their old hosts. Also, we are given m hosts with available capacity $C'_1, C'_2, \ldots, C'_m$. (Note the difference between the capacity C_j and available capacity $C'_j \leq C_j$ of a host. With the notation of Section 2, $C'_j = \tau C_j - S_j$, where S_j is the current utilization of the host.) Furthermore, each host j has a specific power efficiency that can be characterized by the power consumption per unit load, denoted by P_j. That is, allocating VM i to host j increases power consumption by $v_i P_j$. The task is to allocate the VMs to the hosts.

Beloglazov et al. do not define explicitly the objective function for this task. However, they mention that they use a modification of the Best Fit Decreasing (BFD) heuristic because it is guaranteed to use at most $11/9OPT + 1$ bins in the bin packing problem which is strongly related to this problem, where OPT is the minimal number of bins necessary. They also mention that they modified the heuristic in order to make it sensitive to the differences in power efficiency between the hosts [6]. Based on these remarks we can conclude that the objective is twofold: to minimize the number of hosts used for the allocation and to minimize the overall energy consumption of the allocation. Both are indeed plausible objectives: minimizing the number of used hosts allows switching off some hosts and preferring the more energy efficient hosts also saves energy.

The MBFD heuristic works as follows. It iterates once through the VMs in decreasing order of their performance need. (Just like in Section 2, we will assume that the VMs are already ordered appropriately, i.e., $v_1 \geq v_2 \geq \ldots \geq v_n$.) For a VM i, it establishes the set of hosts H_i having sufficient free capacity to accommodate VM i. From H_i, the host with the lowest P_j value (i.e., the most power-efficient) is selected, and the VM is allocated to this host.

In the classic bin packing problem, all bin sizes are equal, and the aim is to minimize the number of bins used. Thus, our problem reduces to the classic bin packing problem if all hosts have the same available capacity and we focus on the first objective, the minimization of the number of hosts used in the allocation. By sorting the hosts in decreasing order of power efficiency – i.e., assuming that $P_1 \leq P_2 \leq \ldots \leq P_m$ – the MBFD heuristic selects always the first host that has sufficient free capacity to accommodate the next VM. This means that MBFD is actually equivalent to the FFD heuristic. (It should be noted that the BFD heuristic is somewhat different as BFD would select the fullest host that has enough capacity to accommodate the VM.) As a result, the same approximation ratio holds for MBFD as is known for FFD [8]:

Theorem 3. *If all hosts have the same available capacity, then the number of hosts used by the MBFD heuristic is at most* $11/9OPT + 1$*, where* OPT *denotes the minimum number of necessary hosts.* □

Looking at the other objective, the total energy consumption of the allocation, no such approximation result can be proven because unfortunately the result of MBFD can be arbitrarily far from the optimum, as demonstrated by the following example.

Example 2. *There are 3 hosts with* $C'_1 = C'_2 = C'_3 = 1$ *and* $P_1 = P_2 = \alpha$*,* $P_3 = \beta$*, where* $\alpha < \beta$*. There are 6 VMs with* $v_1 = v_2 = 0.4$*,* $v_3 = v_4 = v_5 = v_6 = 0.3$*. The optimal allocation maps e.g. VMs 1, 3, and 4 onto host 1 and the other VMs onto host 2, without using host 3 at all. The optimal power consumption is thus* 2α*. MBFD, on the other hand, allocates VMs 1 and 2 to host 1, VMs 3, 4, and 5 to host 2, leaving the last VM to host 3. The resulting power consumption is* $1.7\alpha + 0.3\beta$*. The ratio of the two results is*

$$\frac{1.7\alpha + 0.3\beta}{2\alpha} = \frac{1.7}{2} + \frac{0.3}{2} \cdot \frac{\beta}{\alpha}.$$

Since β/α *can be arbitrarily large, this means that the result of MBFD can be arbitrarily far from the optimum.* □

The next question is what can be stated about the general case, i.e. when the available capacity of the hosts is not equal. Unfortunately, in this case, even the number of hosts required by MBFD can be arbitrarily far from the optimum, as demonstrated by the following example.

Example 3. *There are* r *hosts with* $C'_j = 1$ *and* $P_j = 1$*, plus one more host with* $C'_j = r + 1$ *and* $P_j = 1 + \varepsilon$*. There are* $r + 1$ *VMs, each with* $v_i = 1$*. The optimum allocation is to map each VM to the host with capacity* $r + 1$*, thus using only 1 host. On the other hand, MBFD prefers the hosts with* $P_j = 1$*, so that the first* r *VMs are mapped to those hosts and only the last VM is mapped to the high-capacity host. Thus the number of hosts used by MBFD is* $r + 1$*.* □

One may argue that such big differences in the capacities of the hosts are unlikely because a cloud provider typically uses a large number of hosts of the same or similar type. However, even if the full capacity of the hosts are equal, their available capacity can be very different because of the workload that they are already serving.

To sum up: if the available capacity of each host is equal, then MBFD is guaranteed to use at most $11/9OPT + 1$, where OPT denotes the optimum; however, the used power consumption can be arbitrarily far from the optimum. If the hosts can have different available capacity, then even the number of hosts used by MBFD can be arbitrarily far from the optimum.

4 Interplay of the two heuristics

In this section we investigate to what extent the combination of MM and MBFD can give satisfactory results concerning the overall objective of consolidating the VMs to the minimum number of hosts while avoiding overloading of hosts. For the purposes of this section, we will assume that both heuristics yield optimal results to the sub-problem that they solve, i.e.

- For all over-utilized hosts, MM yields a relieving set with minimum cardinality, and with minimum total utilization among all relieving sets of minimum cardinality.
- MBFD finds an allocation for the selected VMs using the minimum number of hosts.

As we will see, the overall result is not necessarily optimal even under these assumptions, but some approximation guarantees can be given for certain cases and scenarios.

We are given m hosts, each of them with capacity C. As before, a host is considered overloaded if its utilization exceeds τC. The hosts are serving n VMs with capacity needs $v_1, \dots, v_n$; for each VM i, $v_i \leq \tau C$. In a *consolidation step*, some VMs from the overloaded hosts must be migrated either to other already active hosts or to newly switched-on hosts (also having capacity C) beyond the m already active; after the consolidation step, there must be no overloaded host.

Lemma 1. *Let V' be the relieving set found by MM for an overloaded host. If $\tau \geq 2/3$, then $\sum_{i \in V'} v_i \leq \tau C$, and thus V' can be accommodated by a (single) host.*

Proof. As before, let $\ell = |V'|$. If $\ell = 1$, then the statement follows directly from the fact that each VM i has $v_i \leq \tau C$. Now consider the case $\ell \geq 2$. Just like in the proof of Theorem 2, we have

$$\sum_{i \in V'} v_i < \left(1 + \frac{1}{\ell - 1}\right) \cdot (S - \tau C).$$

(See inequality (1).) Using $S \leq C$ and $\ell \geq 2$, it follows that

$$\sum_{i \in V'} v_i < 2(1 - \tau)C.$$

Since $\tau \geq 2/3$, we have $2(1 - \tau)C \leq 2 \cdot 1/3 \cdot C \leq \tau C$, which completes the proof. □

The condition $\tau \geq 2/3$ is indeed necessary: e.g. if a host accommodates three VMs with $v_1 = v_2 = v_3 = C/3$ and $\tau = 2/3 - \varepsilon$, then a minimum relieving set has total utilization $2/3C > \tau C$.

Assuming that $\tau \geq 2/3$ is quite reasonable. For example, Beloglazov et al. considered 70%, 80%, and 90% as possible values for the upper threshold of the MM algorithm [6]. Tomás and Tordsson suggested target utilization levels of 70-80% [18]. Lago et al. try to keep utilization between 80% and 100% [28]. Ghosh and Naik used upper thresholds of 70% and 95% [29].

For these reasons, we will henceforth assume that $\tau \geq 2/3$.

Theorem 4. *With the above assumptions, the number of hosts needed for a consolidation step by the optimal MM+MBFD combination is less than twice the optimum.*

Proof. Let k be the number of overloaded hosts. MM determines for each of them a relieving set; according to Lemma 1, each of these relieving sets can be accommodated by a new host. Thus, the number of hosts required by MM+MBFD is $m' \leq m + k \leq 2m$. On the other hand, the optimum is at least m, which can be achieved only if all relieving sets can be accommodated on the existing hosts. Therefore, $OPT \geq m$ and thus $m' \leq 2m \leq 2OPT$. □

This result is tight, as shown by the next example.

Example 4. *Each of the m hosts accommodates a 'big' VM with utilization $\tau C - \varepsilon$ and two 'small' VMs with utilization 2ε each, where ε is a small positive number. Thus, each host is overloaded. MM selects for each host a relieving set consisting of the big VM because this is the only relieving set of cardinality 1. Since the VMs selected this way do not fit on the already active hosts, MBFD needs to switch on a new host for each of them. Therefore, MM+MBFD needs $2m$ hosts. On the other hand, if we select the relieving set consisting of the two small VMs from each host, then we need new hosts only for these small VMs. Choosing ε in such a way that $4m\varepsilon \leq \tau C$, the $2m$ VMs with capacity need 2ε each will fit on a single new host; thus the optimum is $m + 1$. The ratio of the result of MM+MBFD to the optimum is $2m/(m + 1)$, which can be arbitrarily close to 2.* □

This example demonstrates that although MM delivers an optimal result with respect to its local subgoal, this may not be optimal on a global scale.

Until now, we were focusing on the handling of overloaded hosts. However, in its original form, MM also selects all VMs residing on under-utilized hosts for migration to other hosts, with the aim of switching off the emptied hosts. Let $0 < \lambda < \tau < 1$ be the thresholds for under- and over-utilization: that is, a host with capacity C is under-utilized if its utilization is below λC and – just as before – over-utilized if its utilization exceeds τC.

Theorem 5. *Let us assume that MM+MBFD succeeds in keeping the utilization of each host between λC and τC. Let m_A denote the number of hosts used by MM+MBFD. Then, $m_A \le \tau/\lambda \; OPT$, where OPT is the minimum number of necessary hosts.*

Proof. Since in the allocation established by MM+MBFD, each host has utilization at least λC, it follows that $m_A \lambda C \le \sum_{i=1}^{n} v_i$. In the optimal allocation, each host has utilization at most τC, and thus $\sum_{i=1}^{n} v_i \le OPT \tau C$. From these two inequalities, we have $m_A \lambda C \le OPT \tau C$, from which the theorem follows immediately. □

It is interesting to consider how good or bad the approximation factor of τ/λ may be. For example, Beloglazov et al. found that a difference of $\tau - \lambda = 0.4$ is practical and worked with (λ, τ) pairs of $(0.3, 0.7)$, $(0.4, 0.8)$, $(0.5, 0.9)$ [6]. These cases lead to τ/λ values of 2.33, 2, and 1.8, respectively. Reiss et al. found in the analysis of real-world traces from a large Google cluster CPU utilization levels between 0.3 and 0.6 [30]; using these values as λ and τ would also lead to $\tau/\lambda = 2$.

The result of Theorem 5 is again tight, as shown by the next example.

Example 5. *For simplicity, let $C = 1$. We show the example for the threshold values $\lambda = 0.3$ and $\tau = 0.8$ but it can be easily generalized to other threshold values as well. Starting with the empty state (no VM), altogether $24k$ VM requests (where k is a positive integer) arrive with the following sequence of capacity needs: $0.3, 0.3, 0.2, 0.3, 0.3, 0.2$, and so on, i.e., $8k$ times the $0.3, 0.3, 0.2$ sequence. MBFD allocates these VMs to $8k$ hosts, each now hosting two VMs with utilization 0.3 and a third VM with utilization 0.2. Afterwards, the utilization of each VM drops to 0.1; the allocation remains. Note that no consolidation step occurs during the whole process because the utilization of each host is always between 0.3 and 0.8. Thus, the number of used hosts is $8k$. On the other hand, the optimal allocation for $24k$ VMs with capacity need 0.1 requires only $3k$ hosts, each of them accommodating 8 VMs. Hence in this case $m_A/OPT = (8k)/(3k) = \tau/\lambda$.* □

This example shows that the policy of only migrating VMs from under- or over-utilized hosts may lead to a suboptimal overall allocation.

5 Conclusion

In this paper, we investigated two heuristics for the consolidation of virtual machines in a cloud data center: the MM and MBFD heuristics [6]. In contrast to most previous work that reported only empirical results, we investigated formally provable performance guarantees for the algorithms.

5.1 Summary of results

Our findings can be summarized as follows:

- The MM heuristic delivers optimal result concerning the cardinality of the found relieving set of VMs.
- The relieving set delivered by MM is at most 3/2 times the optimum concerning total utilization.
- If all hosts have the same available capacity, then the result of MBFD is at most $11/9OPT + 1$ concerning the number of hosts.

- The result of MBFD can be arbitrarily far from the optimum concerning energy consumption, even in the case of equal available capacities.
- If the available capacity of the hosts can be different, then the result of MBFD can be arbitrarily far from the optimum also concerning the number of hosts.
- If the capacity of the hosts is equal and MM and MBFD behave optimally, then the consolidation step of MM+MBFD results in less than twice the optimum number of hosts.
- If the capacity of the hosts is equal and MM+MBFD succeed in keeping the utilization of the hosts between λ and τ, then the number of hosts used is at most τ/λ times the optimum.
- We showed with appropriate examples that all of the above approximation ratios are tight.

5.2 Implications

The analysis has shown the strengths and limitations of the investigated algorithms, as well as the areas for future research. In particular, it has been demonstrated that even quite simple algorithms can be proven to have strong approximation ratios. Several similar heuristics have been proposed in the literature for different versions of the VM allocation problem; we expect that their approximation characteristics can be analyzed with similar techniques. However, the approximation properties of slightly different algorithms can be very different, also depending heavily on subtleties of the problem formulation. Therefore, understanding approximation properties precisely is important, especially to derive guarantees for usage in large productive data centers. Also some significantly more complicated heuristics have been proposed in the literature, where such an analysis may be prohibitively difficult. The lack of formally proven performance guarantees may be a significant obstacle to the real-world application of such algorithms.

It is important to note that the formal analysis of approximation properties should not replace experimental evaluations. For example, given two algorithms with the same approximation ratio, it is possible that one of them offers considerably better results in practice, and this is not apparent from the theoretical analysis, only from the experimental evaluation. On the other hand, this information can be very important – for instance, a 2-approximation may be an important theoretical contribution, but in practice, one would strongly prefer if the error is not more than 10-20%. In this respect, formal analysis and experimental evaluation complement each other.

From our work, also the limitations of the applicability of results from bin-packing to VM consolidation have become apparent. Moreover, the results show the impact that the decomposition of the VM consolidation problem into two subproblems has on the achievable effectiveness. This is an important question that arises in other contexts as well: from an engineering point of view, decomposing the problem into subproblems and solving them independently is useful to reduce complexity; however, this way we may lose the ability to find an optimal solution to the overall problem. On the other hand, attacking the whole problem at once may be too difficult because the search space is so huge; in this respect, a decomposition approach may be preferable. Similar questions arise also in the context of other decompositions of the VM allocation problem: e.g., whether data nodes should be placed first and compute nodes only afterwards [26] or the placement of data nodes and compute nodes should be optimized together [31]. This is a highly non-trivial trade-off that must be resolved for every optimization problem and every potential decomposition approach separately.

Acknowledgements

This work was partially supported by the Hungarian Scientific Research Fund (Grant Nr. OTKA 108947) and the János Bolyai Research Scholarship of the Hungarian Academy of Sciences.

References

[1] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, I. Brandic, Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility, Future Generation Computer Systems 25 (6) (2009) 599–616.

[2] Digital Power Group, The cloud begins with coal – Big data, big networks, big infrastructure, and big power, 2013.

[3] R. Buyya, A. Beloglazov, J. Abawajy, Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges, in: Proceedings of the 2010 International Conference on Parallel and Distributed Processing Techniques and Applications, CSREA Press, 2010, pp. 6–17.

[4] S. Srikantaiah, A. Kansal, F. Zhao, Energy aware consolidation for cloud computing, Cluster Computing 12 (2009) 1–15.

[5] A. Beloglazov, R. Buyya, Energy efficient allocation of virtual machines in cloud data centers, in: 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing, 2010, pp. 577–578.

[6] A. Beloglazov, J. Abawajy, R. Buyya, Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing, Future Generation Computer Systems 28 (2012) 755–768.

[7] D. S. Johnson, A. J. Demers, J. D. Ullman, M. R. Garey, R. L. Graham, Worst-case performance bounds for simple one-dimensional packing algorithms, SIAM Journal on Computing 3 (4).

[8] G. Dósa, The tight bound of first fit decreasing bin-packing algorithm is $FFD(I) \leq 11/9OPT(I) + 6/9$, in: Combinatorics, Algorithms, Probabilistic and Experimental Methodologies, Springer, 2007, pp. 1–11.

[9] G. Dósa, J. Sgall, First fit bin packing: A tight analysis, in: 30th Symposium on Theoretical Aspects of Computer Science (STACS), 2013, pp. 538–549.

[10] J. Sgall, A new analysis of best fit bin packing, in: Fun with Algorithms, 2012, pp. 315–321.

[11] N. Bobroff, A. Kochut, K. Beaty, Dynamic placement of virtual machines for managing SLA violations, in: 10th IFIP/IEEE International Symposium on Integrated Network Management, 2007, pp. 119–128.

[12] E. Casalicchio, D. A. Menascé, A. Aldhalaan, Autonomic resource provisioning in cloud systems with availability goals, in: Proceedings of the 2013 ACM Cloud and Autonomic Computing Conference, 2013.

[13] C. Hyser, B. McKee, R. Gardner, B. J. Watson, Autonomic virtual machine placement in the data center, Tech. rep., HP Laboratories (2008).

[14] L. Liu, H. Wang, X. Liu, X. Jin, W. B. He, Q. B. Wang, Y. Chen, GreenCloud: A new architecture for green data center, in: Proceedings of the 6th International Conference on Autonomic Computing and Communications, 2009, pp. 29–38.

[15] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wang, X. Zhu, No "power" struggles: Coordinated multi-level power management for the data center, in: Proceedings of the 13th International Conference on Architectural Support for Programming Languages and Operating Systems, 2008, pp. 48–59.

[16] J. Xu, J. A. B. Fortes, Multi-objective virtual machine placement in virtualized data center environments, in: Green Computing and Communications (GreenCom), 2010 IEEE/ACM International Conference on Cyber, Physical and Social Computing (CPSCom), 2010, pp. 179–188.

[17] A. Verma, G. Dasgupta, T. K. Nayak, P. De, R. Kothari, Server workload analysis for power minimization using consolidation, in: Proceedings of the 2009 USENIX Annual Technical Conference, 2009, pp. 355–368.

[18] L. Tomás, J. Tordsson, An autonomic approach to risk-aware data center overbooking, IEEE Transactions on Cloud Computing 2 (3) (2014) 292–305.

[19] D. M. Batista, N. L. S. da Fonseca, F. K. Miyazawa, A set of schedulers for grid networks, in: Proceedings of the 2007 ACM Symposium on Applied Computing (SAC'07), 2007, pp. 209–213.

[20] B. Guenter, N. Jain, C. Williams, Managing cost, performance, and reliability tradeoffs for energy-aware server provisioning, in: Proceedings of IEEE INFOCOM, IEEE, 2011, pp. 1332–1340.

[21] B. C. Ribas, R. M. Suguimoto, R. A. N. R. Montano, F. Silva, L. de Bona, M. A. Castilho, On modelling virtual machine consolidation to pseudo-boolean constraints, in: 13th Ibero-American Conference on AI, 2012, pp. 361–370.

[22] M. Guazzone, C. Anglano, M. Canonico, Exploiting vm migration for the automated power and performance management of green cloud computing systems, Tech. Rep. TR-INF-2012-04-02-UNIPMN, University of Piemonte Orientale (2012).

[23] W. Li, J. Tordsson, E. Elmroth, Virtual machine placement for predictable and time-constrained peak loads, in: Proceedings of the 8th International Conference on Economics of Grids, Clouds, Systems, and Services (GECON 2011), Springer, 2011, pp. 120–134.

[24] D. Breitgand, A. Epstein, Improving consolidation of virtual machines with risk-aware bandwidth oversubscription in compute clouds, in: Proceedings of IEEE Infocom, 2012, pp. 2861–2865.

[25] M. Alicherry, T. Lakshman, Network aware resource allocation in distributed clouds, in: Proceedings of IEEE Infocom, 2012, pp. 963–971.

[26] M. Alicherry, T. Lakshman, Optimizing data access latencies in cloud systems by intelligent virtual machine placement, in: Proceedings of IEEE Infocom, 2013, pp. 647–655.

[27] D. Breitgand, A. Marashini, J. Tordsson, Policy-driven service placement optimization in federated clouds, Tech. rep., IBM Research Report, H-0299 (H1102-014) (2011).

[28] D. Lago, E. Madeira, L. Bittencourt, Power-aware virtual machine scheduling on clouds using active cooling control and DVFS, in: Proceedings of the 9th International Workshop on Middleware for Grids, Clouds and e-Science, 2011.

[29] R. Ghosh, V. K. Naik, Biting off safely more than you can chew: Predictive analytics for resource over-commit in IaaS cloud, in: 5th International Conference on Cloud Computing, IEEE, 2012, pp. 25–32.

[30] C. Reiss, A. Tumanov, G. R. Ganger, R. H. Katz, M. A. Kozuch, Heterogeneity and dynamicity of clouds at scale: Google trace analysis, in: ACM Symposium on Cloud Computing (SoCC), 2012.

[31] M. Korupolu, A. Singh, B. Bamba, Coupled placement in modern data centers, in: IEEE International Symposium on Parallel and Distributed Processing (IPDPS), 2009, pp. 1–12.