Energy Efficient VM Scheduling for Cloud Data Centers: Exact allocation and migration algorithms

Chaima Ghribi, Makhlouf Hadji and Djamal Zeghlache

Institut Mines-Télécom, Télécom SudParis
UMR CNRS 5157
9, Rue Charles Fourier
91011, Evry, France
Email: {chaima.ghribi, makhlouf.hadji, djamal.zeghlache}@it-sudparis.eu

Abstract—This paper presents two exact algorithms for energy efficient scheduling of virtual machines (VMs) in cloud data centers. Modeling of energy aware allocation and consolidation to minimize overall energy consumption leads us to the combination of an optimal allocation algorithm with a consolidation algorithm relying on migration of VMs at service departures. The optimal allocation algorithm is solved as a bin packing problem with a minimum power consumption objective. It is compared with an energy aware best fit algorithm. The exact migration algorithm results from a linear and integer formulation of VM migration to adapt placement when resources are released. The proposed migration is general and goes beyond the current state of the art by minimizing both the number of migrations needed for consolidation and energy consumption in a single algorithm with a set of valid inequalities and conditions. Experimental results show the benefits of combining the allocation and migration algorithms and demonstrate their ability to achieve significant energy savings while maintaining feasible convergence times when compared with the best fit heuristic.

Index Terms—Energy efficiency, VM placement, VM migration, Cloud data center, Linear integer programming.

I. INTRODUCTION

Energy efficiency is becoming increasingly important for data centers and clouds. The wider adoption of cloud computing and virtualization technologies has led to cluster sizes ranging from hundreds to thousands of nodes for mini and large data centers respectively. This evolution induces a tremendous rise of electricity consumption, escalating data center ownership costs and increasing carbon footprints. For these reasons, data centers now embed monitoring capabilities and probes such as smart power distribution units (PDUs) to achieve energy efficiency and reduce overall cost.

The fact that electricity consumption is set to rise 76% from 2007 to 2030 [1] with data centers contributing an important portion of this increase emphasizes the importance of reducing energy consumption in clouds. According to the Gartner Report [2], the average data center is estimated to consume as much energy as 25000 households, and according to McKinsey report [3], "The total estimated energy bill for data centers in 2010 is 11.5 billion and energy costs in a typical data center double every five years".

Reducing the overall energy bill requires first and foremost energy and ecology conscious designs of data centers hardware and software, the use of solar and photovoltaic energy, relying on renewable energy sources and efficient cooling systems. The second and apparently more modest contribution to energy efficient clouds is to introduce energy aware scheduling and placement algorithms and enhanced resource management. In fact, cloud data centers are electricity guzzlers especially if resources are permanently switched on even if they are not used. An idle server consumes about 70% of its peak power [4]. This waste of idle power is considered as a major cause of energy inefficiency.

This paper is a contribution to the reduction of such excessive energy consumption using energy aware allocation and migration algorithms to have a maximum number of idle servers to put into sleep mode. Intel’s Cloud Computing 2015 Vision [5] stresses also the need for such dynamic resource scheduling approaches to improve power efficiency of data centers by shutting down and putting to sleep idles servers. Our work proposes an exact energy aware allocation algorithm using the classical formulation of the Bin-Packing problem. The aim of this algorithm is to reduce the number of used servers or equivalently maximize the number of idle servers to put in sleep mode. To take into account workloads and service times a linear integer programming algorithm is used to optimize constantly the number of used servers after service departures. This migration algorithm is combined with the exact allocation algorithm to reduce overall energy consumption in the data centers.

The proposed algorithms act as an energy consumption aware VM scheduler and can be used to enhance current infrastructure managers and schedulers such as OpenNebula [6] and OpenStack [7]. The power consumption indicators can be provided by energy consumption estimation tools such as joulemeter [8]. A dedicated simulator is used to assess performance and crosscheck with the performance results produced by the exact algorithms. Evaluation results show that the exact allocation algorithm combined with migration reduces considerably the number of required servers to serve a given load and can thus minimize power consumption in data centers. Section II of this paper describes a related work addressing consolidation using different migration techniques. The system model and the derived algorithms are addressed in Section III. Performance evaluation is the object of section IV.

II. RELATED WORK

Authors in [9] consolidate applications or tasks on reduced number of physical machines to switch off machines in
surplus. This approach differs from ours because consolidation is achieved at the task level, rather than the virtual machine (VM) level (or IaaS level) in our case, and hence fits better the Platform or Software as a Service (PaaS, SaaS) levels. Allocation or placement is also static as opposed to our dynamic placement according to workload where we apply migration to allocate and reallocate VMs.

Reference [10] addresses policies for dynamic VMs reallocation using VMs migration according to CPU performance requirements. Their most effective policy, a double threshold policy, is based on the idea of setting upper and lower utilization thresholds for hosts and keeping the total utilization of the CPU of all the VMs between these thresholds. If the CPU utilization of a host exceeds the upper threshold, some VMs are migrated and if it falls below the lower threshold, all the hosted VMs should be migrated.

One of the closest work to our paper is found in [11]. Authors treat the problem of consolidating VMs in a server by migrating VMs with steady and stable capacity needs. They proposed an exact formulation based on a linear program described by a too small number of valid inequalities. Indeed, this description does not allow solving, in reasonable time and in an optimal way, problems involving allocation of a large number of items (or VMs) to many bins (or Servers). In order to scale and find solutions for large sizes, the authors resorted to a heuristic using a static and a dynamic consolidation of VMs to reduce energy consumption of the hosting nodes or servers.

In [12], authors presented a server consolidation (Sercon) algorithm which consists of minimizing the number of used nodes in a data center and minimizing the number of migrations at the same time. They compared their algorithm with the well known placement heuristic FFD (First-Fit Decreasing) [13], to solve the Bin-Packing problem, and have shown the efficiency of Sercon to consolidate VMs and minimize migrations. However, Sercon is a heuristic that can not always reach or find the optimal solution. Our proposed algorithm is based on an exact formulation of the consolidation problem with migrations. We manage to optimally consolidate VMs in servers while minimizing the energy cost of migrations.

In [14], authors presented an approach EnaCloud for dynamic live placement taking into account energy efficiency in a cloud platform. They proposed an energy-aware heuristic algorithm in order to save energy by minimizing the number of running servers. Another study relying on dynamic resource allocation is presented in [15]. The authors presented a nature-inspired VM consolidation algorithm inspired from an Ant Colony Optimization. This algorithm aims at reducing the number of used physical machines and thus saves energy.

III. The System Model

The model considers infrastructure providers allocating physical resources instances to host users’ and tenants’ applications or equivalently for this paper VMs. The physical resources are seen as servers. It is assumed that applications are packaged into virtual machines to be hosted by the infrastructure providers. The cloud providers save energy and reduce power consumption by packing and consolidating through migration of VMs to maximize the number of idle servers to put to sleep mode.

Figure 1 depicts the system model composed of the proposed energy efficient allocation and migration algorithms contributing to scheduling, an energy consumption estimator and cloud managers that handle infrastructure resource instantiation and management. Each module is briefly described to set the stage for the analytical modeling of the energy efficient resource allocation problem in clouds.

- **Cloud IaaS manager** (e.g. OpenStack [7], OpenNebula [6] and Eucalyptus [16]) control and manage cloud resources and handle clients requests, VM scheduling and fetch and store images in storage spaces.
- **Energy estimation module** is an intermediate module between the cloud infrastructure manager and the energy-aware scheduler. The module can rely for example on an energy estimation tool as in Joulemeter [8] that uses power models to infer power consumption of VMs or servers from resource usage.
- **Energy-aware VM scheduler** responsible for the energy aware VM placement in the data center is the focus of our energy consumption optimization model. This green scheduler is basically composed of two modules. An allocation module and a migration module. The role of the allocation module is to perform the initial VM placement using our exact VM allocation algorithm. The dynamic consolidation of virtual machines is handled by the migration module that minimizes the number of used or activated servers thanks to our exact VM migration algorithm. The unused servers are shut down or put into sleep mode. All the needed information (servers and VMs) to run the algorithms are retrieved via the Cloud IaaS manager that is also used to execute the VM deployment and migration actions.

To derive the system model, we consider the size $n$ of client requests in terms of the number of required VMs and the types of desired VM instances (e.g., small, medium, large). Each $VM_i$ is characterized by a lifetime $t_i$ and a maximum power consumption $p_i$. Each server or hosting node $j$, from the data center, has a power consumption limit or power cap noted $P_{j,Max}$. This can be fixed by cloud administrators. We assume that all servers are homogeneous; extending the model to heterogeneous servers is trivial but will increase complexity and will not necessarily provide additional insight.

The approach adopted to achieve energy efficiency in our proposal is to use a bin packing algorithm for optimal placement of user requests and to follow with dynamic consolidation once a sufficient number of departures have occurred. The dynamic consolidation is handled by the migration algorithm which regroups VMs to free as many servers as possible to put them into sleep mode or to shut them down.

A. Exact VM Allocation Algorithm

The proposed exact VM allocation algorithm is an extended Bin-Packing approach through the inclusion of valid conditions expressed in the form of constraints or inequalities. The objective is to pack items (VMs in our case) into a set of bins (servers or nodes hosting the VMs) characterized by their power consumptions. In addition to $n$, the number
available in the data center. The servers are assumed to have the same power consumption limit: $P_{j,Max}, \{j = 1, 2, ..., m\}$. At run-time, each server $j$ hosting a number of VMs is characterized by its current power consumption: $P_{j,Current}$.

Since the objective is to minimize the energy consumption of the data centers, we define as key decision variable $e_j$ for each server $j$ that is set to 1 if server $j$ is selected to host VMs, 0 if it is not selected. In addition, we define the bivalent variable $x_{ij}$ to indicate that $VM_i$ has been placed in server $j$ and set $x_{ij} = 1$; $x_{ij} = 0$ otherwise. The objective function to place all the demands (or VMs) in a minimum number of servers can be expressed using:

$$\min Z = \sum_{j=1}^{m} e_j$$  

This optimization is subject to a number of linear constraints reflecting the capacity limits of the servers and obvious facts such as a VM can only be assigned to one server or a server can only host VMs according to the amount of remaining resources:

1) Each server has a power limit $P_{j,Max}$ that cannot be exceeded when serving or hosting VMs and this occurs according to remaining capacity:

$$\sum_{i=1}^{n} p_i x_{ij} \leq P_{j,Max} e_j - P_{j,Current}, \forall j = 1, ..., m$$  

2) A cloud provider has to fulfill all requests within a prescribed SLA or quota and each requested VM will be assigned to one and only one server:

$$\sum_{j=1}^{m} x_{ij} = 1, \forall i = 1, ..., n$$  

3) For servers verifying the condition $P_{j,Max} > P_{j,Current}$ and $P_{j,Current} \neq 0$, the total number of used servers is lower bounded by $\left\lceil \frac{\sum_{j=1}^{m} P_{j,Current}}{P_{j,Max}} \right\rceil$. This adds the following inequality to the model:

$$\sum_{j=1}^{m} e_j \geq \left\lceil \frac{\sum_{j=1}^{m} P_{j,Current}}{P_{j,Max}} \right\rceil$$

The exact and extended Bin-Packing VM allocation model can be summarized by lumping the objective function with all the constraints and conditions and constraints into the following set of equations:

$$\min Z = \sum_{j=1}^{m} e_j$$  

Subject To:

$$\sum_{i=1}^{n} p_i x_{ij} \leq P_{j,Max} e_j - P_{j,Current}, \forall j = 1, ..., m$$  

$$\sum_{j=1}^{m} x_{ij} = 1, \forall i = 1, ..., n$$  

$$\sum_{j=1}^{m} e_j \geq \left\lceil \frac{\sum_{j=1}^{m} P_{j,Current}}{P_{j,Max}} \right\rceil$$  

$$e_j = \begin{cases} 1, & \text{if the server } j \text{ is used;} \\ 0, & \text{otherwise.} \end{cases}$$  

$$x_{ij} = \begin{cases} 1, & \text{if the } VM_i \text{ is placed in server } j; \\ 0, & \text{otherwise.} \end{cases}$$

All the variables and constants used in the model are listed for easy reference below:

- $n$ is the size of the request in number of requested VMs.
- $m$ is the number of servers in the data center.
- $p_i$ represents the power consumption of $VM_i$.
- $x_{ij}$ is a bivalent variable indicating that $VM_i$ is assigned to a server $j$.
- $e_j$ is a variable used to indicate whether the server $j$ is used or not.
- $P_{j,Max}$ represents the maximum power consumption of server $j$.
- $P_{j,Current}$ represents the current power consumption of server $j$ ($P_{j,Current} = P_{j,idle} + \sum_k P_k$ with VM$s_k$ hosted by server $j$).
- $P_{j,idle}$ represents the power consumption of server $j$ when it is idle.

Constraints in server CPU, memory and storage are also added to the model to confine even further the model convex hull:

$$\sum_{i=1}^{n} cpu_i x_{ij} \leq CPU_j e_j$$  

where $cpu_i$ is the requested CPU by $VM_i$; $CPU_j$ is the CPU capacity of server $j$.

$$\sum_{i=1}^{m} mem_i x_{ij} \leq MEM_j e_j$$  

where $mem_i$ is the requested memory by $VM_i$ and $MEM_j$ is the memory capacity of server $j$.

$$\sum_{i=1}^{m} sto_i x_{ij} \leq STO_j e_j$$  

where $sto_i$ is the requested storage by $VM_i$ and $STO_j$ is the storage capacity of server $j$. 

![Diagram](image-url)
In this paper we assume that these constraints are met and verified and we hence only need to focus on the energy efficiency constraints through (2).

B. Best-Fit formulation and heuristic algorithm

The exact and extended Bin-Packing is compared to a Best-Fit heuristic adaptation of the well known Best-Fit algorithm [13]. The heuristic proposed to achieve energy efficient VM placement consists of two steps:

1) sorting the requested VMs in decreasing order of power consumption. This builds somehow an ordered stack that is used in the second step for packing VMs in available servers;

2) The sorted VMs are handled starting from the top of the stack and attempting to place the most power consuming VMs in the server with the smallest remaining power consumption budget until a VM down the stack fits in this target server. The process repeats or continues until all VMs in the stack are placed and packed as much as possible in the most occupied servers. This will tend to free servers for sleep mode or switching off.

As this Best-Fit heuristic algorithm tries to approximate the Bin-Packing algorithm, it is selected for comparison with our exact VM allocation proposal. The allocation algorithms are combined with a migration algorithm to minimize overall data center power consumption. In our case, the objective is to benchmark the exact VM allocation and migration algorithms with at least one well known heuristic algorithm. The Best-Fit heuristic was selected since it is known to achieve good suboptimal performance compared with classical Bin-Packing.

C. Exact VMs Migration

The placed and running VMs in the servers will gradually leave the system as their related jobs end. These departures are the opportunity to re-optimize the placement by migrating VMs always in the system for consolidation in a minimum number of fully packed servers. A migration algorithm based on an integer linear program (ILP) is presented to achieve the consolidation. This ILP algorithm consists in introducing a number of valid inequalities to reduce the span of the convex hull of the migration problem.

The mathematical model for the VM consolidation via migration relies on a linear integer programming formulation. The objective for the algorithm is to migrate VMs from nodes selected as source nodes (those the algorithm aims at emptying so they can be turned off) to other selected destination nodes (those the algorithm aims at filling so they serve a maximum number of VMs within their capacity limits).

Ideally, the algorithm should minimize the number of active nodes, maximize the overall number of VMs handled by the active nodes and hence maximize the number of unused, empty or idle nodes. The algorithm should also minimize the power consumption caused by migrations. If the power consumption or cost of VM migration is uniform or homogeneous across hosting nodes or servers, the objective reduces to minimizing the number of migrations.

The migration concerns the set of non idle servers $m'$, $m' < m$, whose power consumptions are lower than $P_{j,\text{Max}}$ with $j$ in $m'$. Despite the slight reduction in size $m' < m$, the problem remains NP-hard. Hence, we resort to an exact algorithm based on linear integer programming to address optimal migration for practical problem sizes or number of instances.

![Fig. 2. Example of VMs’ migration](image)

The objective function for the optimal VM migration and consolidation can be expressed as the maximization of the number of idle servers in the infrastructure:

$$\text{max } M = \sum_{i=1}^{m'} P_{i,\text{idle}} y_i - \sum_{i=1}^{m'} \sum_{j=1}^{m'} q_i p_k' z_{ijk} \quad (14)$$

where $y_i = 1$ is used to indicate that server $i$ is idle and $y_i = 0$ means that at least one VM is active in server $i$. $P_{i,\text{idle}}$ is the power consumed by idle servers, $p_k'$ is the cost in terms of consumed power when migrating $VM_k$. A fixed power consumption $p_k'$ is assigned to each VM instance according to instance type to differentiate between small, medium and large VMs.

Variable $z_{ijk}$ is the bivalent variable expressing migration of $VM_k$ from server $i$ to server $j$. Variable $q_i$ is the total number of VMs hosted on server $i$ and that are candidate for migration into destination servers, especially server $j$ in equation (14).

The objective function (14) is subject to the migration constraints cited earlier. These conditions are formally expressed through valid inequalities and constraints that have to be respected when minimizing overall energy consumption.

1) When migrating $VM_k$ from a server $i$ to a server $j$ (see figure 3), the algorithm must prevent backward migrations and can only migrate into one specific destination node. Stated in an equivalent way: if a $VM_k$ is migrated from a server $i$ (source) to a server $j$ (destination), it can not be migrated to any other server $l$ ($l \neq j$). The proposed inequality (15) also ensures that VMs in destination node and VMs migrated to destination nodes are not migrated as we are aiming at filling these nodes instead of emptying them obviously. This is reflected by the inequality:

$$z_{ijk} + z_{jlk} \leq 1; \quad (15)$$

2) To strengthen further the previous condition, a valid inequality is added to ensure that when a $VM_k$ is migrated from server $i$ to server $j$ (see figure 4), migration to other nodes $l$ ($l \neq j$) are prevented or forbidden:

$$\sum_{j=1, j \neq i}^{m'} z_{ijk} \leq 1; \quad (16)$$
Subject To:

$$\forall i = 1, \ldots, m', \forall j = 1, \ldots, m', \forall k, \forall k' = 1, \ldots, q_i, \forall l \neq j, k \neq k'.$$

$$\sum_{j=1, j \neq i}^{m'} z_{ijk} \leq 1$$  (23)

$$\forall i = 1, \ldots, m', \forall j = 1, \ldots, m', \forall k = 1, \ldots, q_i, \forall l = 1, \ldots, m', l \neq j.$$

$$\sum_{i=1, m, j \neq i}^{m'} \sum_{k=1}^{q_i} p_k z_{ijk} \leq (P_{j, Max} - P_{j, Current}) (1 - y_j)$$  (24)

$$\forall j = 1, \ldots, m, j \neq i$$

$$\sum_{i=1, m, j \neq i}^{m'} \sum_{k=1}^{q_i} z_{ijk} = q_i y_i, \forall i = 1, \ldots, m', j \neq i$$  (25)

$$\sum_{i=1}^{m'} y_i \leq m' - \left[ \sum_{j=1}^{m'} \frac{P_{j, Current}}{P_{j, Max}} \right]$$  (26)

$$z_{ijk} \Delta t_k \geq T_0,$$  (20)

where $\Delta t_k = t_k - CurrentTime$, where $CurrentTime$ represents current or VM migration handling time.

The optimal VM consolidation and migration model and objective function (14) can be summarized for convenience with all the valid conditions as:

$$\max M = \sum_{i=1}^{m'} P_{i,d} e_i y_i - \sum_{i=1}^{m'} \sum_{j=1}^{m'} \sum_{k=1}^{q_i} p_k z_{ijk}$$  (21)

Subject To:
smallest possible set of nodes. All emptied or freed servers (or nodes) are turned off to minimize energy consumption.

The consolidation is achieved by the exact migration algorithm that moves VMs from selected source nodes to selected destination nodes. The end result is the activation and use of the smallest set of nodes in the data centers.

IV. Numerical results

Our proposed algorithms are evaluated through a Java language implementation and the linear solver CPLEX [17]. A dedicated simulator is developed to conduct the performance assessments and the comparison. The objective of the numerical evaluation is to quantify the percentage of energy savings or power consumption savings that can be expected when combining the exact allocation algorithm and the consolidation process using our proposed exact migration algorithm. The answers provided by the numerical analysis concern also the scalability and complexity of the proposed algorithms in the size of the data centers and the arrival rate of requests for resources to host VMs which is also synonymous to load on the system. Note, however, that the simulation are conducted for an arrival rate strictly lower than the rate of VM job departures from the system; thus simulations correspond to cases where the likelihood of finding an optimal or a good solution is high.

The assessment scenarios correspond to data centers with 100 servers or nodes for the first five experiments. We collect essentially as performance indicators, the percentage of used servers (which automatically provides the energy consumed or saved by the algorithms) and the time required for the algorithms to find their best solutions (optimal for the exact algorithms). All the servers have a power consumption cap \( P_{j,\text{Max}} \) set to 200 watts (the peak power of a typical server is around 250 watts [18]). To perform per-VM power estimation we referred to a power estimation model proposed in [19]. Three \( SPEC\text{cpu2006} \) [20] workloads (454.calculix, 482.sphinx and 435.gromacs) with high, medium and low power consumption were considered. Their associated power consumption is close to 13 watts, 11 watts and 10 watts respectively. The power estimation model proposed in [8] provided additional insight. The power consumption of other \( SPEC\text{cpu} \) workloads (471.omnetpp, 470.lbm and 445.gobmk) were evaluated. Estimated power consumptions were found to be between 25 and 28 watts for these elements. Without loss of generality and to ease intuitive verification of the results, we refer to these published consumption to associate to each VM type (small, medium and large) an energy consumption \( p_i \) respectively equal to 10 watts (low), 20 watts (medium) and 30 watts (high) to stay in line with published values. The requests for resources to serve VMs have a constant arrival rate. The requested VM instance types are discrete uniform in \([1,3]\) (1 for small, 2 for medium and 3 for large instances). The VM sizes are arbitrarily drawn as uniform in \([1,3]\) and classified according to their type. We retained only the random drawings that fulfill the VM characteristics in size and type.

Figure 6 depicts results of a comparison between the adapted Best-Fit heuristic and our exact extended Bin-Packing allocation algorithms. The simulations correspond to 100 servers and resource requests in number of VMs in the \([1,200]\) range. The lifetime of the VMs are uniform in \([30s, 180s]\). That is VM jobs last at least 30s and will terminate in less than 180s. The exact allocation algorithm as expected outperforms the Best-Fit heuristic for the 1000s time interval simulated and reported in Figure 6. The Best-Fit heuristic uses more often all available nodes or servers (100% ordinate value in Figure 6) while the exact algorithm manages to use fewer nodes with 10 to 50% more unused servers that Best-Fit.

Figure 7 extends the analysis for the exact and extended Bin-Packing allocation algorithm by comparing its performance with and without consolidation. When the exact algorithm is combined with the migration algorithm (that uses migration to empty some nodes) it can significantly provide additional power savings or energy consumption reduction. The average gain can be estimated to be 10 to 20% more servers that could be turned off. The average line for the exact algorithm is around 80% of servers used while the average for the exact algorithm with migration is more in the order of 60%.

Figure 8 pursues the analysis for the exact bin-Packing VM allocation algorithm by reporting performance as a function
of data center sizes and VM requests induced load. The time before convergence to the optimal placement is reported as a function of data center size (from 100 to 1000 nodes or servers) for request sizes ranging from 50 to 500 VMs. Clearly, because the problem is NP-Hard, the convergence time of the exact algorithm grows exponentially for requests exceeding 300 VMs; especially for number of servers beyond 400. The time needed to find the optimal solutions remains acceptable and reasonable, within 10 s, for data center sizes below 500 receiving requests less than 400 VMs. The time needed for convergence grows unacceptably high outside of this operating range for the simulated scenarios (tens of seconds to few minutes). This motivated the use of the Best-Fit algorithm to find solutions faster even if they are bound to be suboptimal as reported in Figure 6.

Figures 9, 10 and 11 address the performance of the consolidation algorithm via the analysis of the time needed to achieve migrations of VMs from source nodes to destination nodes in order to free as many servers as possible and gain the opportunity to shut them down. The assessment is performed consequently on the active servers, i.e. those currently serving VMs and candidate for consolidation. The performance as a function of increasing number of active nodes $m'$ to consolidate is reported in the three figures for $m' = 5$, $m' = 10$ and $m' = 20$. For $m' = 5$, consolidation after migration from source to destination nodes can be achieved in the milliseconds time scales (few to 300 ms in Figure 9). The number of hosted VMs to consolidate varies from 5 to 30 for this simulation.

The time needed for consolidation increases to seconds in Figure 10 for $m' = 10$. The curve is reported for up to 60 VMs hosted in the $m'$ nodes considered or subject to consolidation/migration. When the number of servers to consolidate increases further, as shown in Figure 11 for $m' = 20$, the convergence times move to orders of tens to hundreds of minutes (for the extreme case, on the curve upper right corner, this reaches 180 minutes for 120 hosted VMs). These three figures highlight the limits of the exact migration algorithm with increasing number of servers to consolidate.

The next experiments and simulations address the achievable energy savings using different VM requests inter-arrival times (noted by $\lambda^{-1}$) and lifetimes (represented by $\mu^{-1}$) that also reflects the service rate $\mu^{-1}$ in order to assess the performance for variable system loads since the ration $\lambda/\mu$ governs performance. The number of servers has been fixed to 200 hosting nodes for the reported results in Table I. One hundred (100) simulation runs are average for each parameter setting in the table. Table I reports the energy savings with
the migration algorithm compared to the allocation algorithm without migration.

<table>
<thead>
<tr>
<th>$\lambda^{-1}(s)$</th>
<th>5</th>
<th>10</th>
<th>15</th>
<th>20</th>
<th>25</th>
<th>30</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>35.55</td>
<td>36.59</td>
<td>00.00</td>
<td>00.00</td>
<td>00.00</td>
<td>00.00</td>
</tr>
<tr>
<td>20</td>
<td>27.29</td>
<td>34.00</td>
<td>35.23</td>
<td>38.50</td>
<td>00.00</td>
<td>00.00</td>
</tr>
<tr>
<td>30</td>
<td>17.48</td>
<td>27.39</td>
<td>35.21</td>
<td>40.32</td>
<td>41.89</td>
<td>36.58</td>
</tr>
<tr>
<td>40</td>
<td>16.77</td>
<td>18.83</td>
<td>22.02</td>
<td>32.31</td>
<td>39.90</td>
<td>40.50</td>
</tr>
<tr>
<td>50</td>
<td>10.36</td>
<td>16.17</td>
<td>19.85</td>
<td>22.30</td>
<td>39.20</td>
<td>36.52</td>
</tr>
<tr>
<td>60</td>
<td>08.63</td>
<td>14.29</td>
<td>18.01</td>
<td>23.13</td>
<td>25.15</td>
<td>30.68</td>
</tr>
<tr>
<td>70</td>
<td>08.10</td>
<td>14.00</td>
<td>14.86</td>
<td>15.90</td>
<td>22.91</td>
<td>23.20</td>
</tr>
<tr>
<td>80</td>
<td>07.01</td>
<td>10.20</td>
<td>10.91</td>
<td>15.34</td>
<td>17.02</td>
<td>21.60</td>
</tr>
<tr>
<td>90</td>
<td>06.80</td>
<td>09.52</td>
<td>10.31</td>
<td>14.70</td>
<td>16.97</td>
<td>19.20</td>
</tr>
<tr>
<td>100</td>
<td>05.90</td>
<td>07.50</td>
<td>08.40</td>
<td>12.90</td>
<td>16.00</td>
<td>14.97</td>
</tr>
</tbody>
</table>

TABLE I

TABLE OF PERCENTAGE OF GAINED ENERGY WHEN MIGRATION IS USED

Energy savings depend evidently on the service rate or the lifetime of VMs or the duration of their jobs relative to the load induced by the VM resource requests. Savings in the simulation can reach as high as 41.89% for inter-arrival times of 25 seconds and job durations of 30 seconds. For less favorable ratios or loads, the savings for the scenarios tested in the evaluation are less significant but remain respectable (5.90%) for the highest loads (inter-arrivals time is equal to 5 seconds) and longer job durations (of 100 sec).

![Fig. 12. Energy savings](image)

In order to complete the analysis, the energy savings that can be achieved by the Best-Fit, the exact allocation and the exact allocation combined with migration are compared for similar scenarios with a restricted set of parameter settings ($\lambda^{-1} = 10s$). All the servers are initially considered OFF, which means that the energy saving is initialized to 100%. Figure 12 depicts the evolution of the percentage of energy saved by the three algorithms. The obvious dependence on system load is reflected by the gradual decrease of energy savings for increasing VM lifetimes (or increasing job durations). For the exact Bin-Packing allocation algorithm the energy savings remain quite high at low load; 90% of energy savings for the exact allocation only and 95% combined migration. Energy savings achieved by the Best-Fit heuristic stay below the percentages achieved by the exact allocation algorithm with or without migration at all different loads. As usual the gap will vanish for saturated systems as the servers will be always busy and permanently turned on, the exact algorithm can nevertheless provide gains if intelligent groupings and re-arrangements are applied (this is however out of scope of this paper).

V. Conclusions

This paper explored the problem of VM placement in cloud providers’ data centers, which is known as NP-hard. Our original contribution consists in using exact algorithms for placement and consolidation that minimize jointly energy consumption and migration costs in acceptable convergence times compared slightly faster heuristics such as best fit.

We have proposed a linear integer program corresponding to an exact allocation algorithm coupled with an efficient exact VM migration algorithm to reduce energy consumption via consolidation. Significant energy savings can be obtained depending on system loads. At low load the gains can be quite significant. These gains at high loads even if much lower remain sufficiently valuable using the proposed algorithms.

References