

Energy-Efficient VM Scheduling in IaaS Clouds

Nguyen Quang-Hung^(✉) and Nam Thoai

Faculty of Computer Science and Engineering, HCMC University of Technology,
VNUHCM, 268 Ly Thuong Kiet Street,
Ho Chi Minh City, Vietnam
{hungnq2,nam}@cse.hcmut.edu.vn

Abstract. This paper investigates the energy-aware virtual machine (VM) scheduling problems in IaaS clouds. Each VM requires multiple resources in fixed time interval and non-preemption. Many previous researches proposed to use a minimum number of physical machines; however, this is not necessarily a good solution to minimize total energy consumption in the VM scheduling with multiple resources, fixed starting time and duration time. We observe that minimizing total energy consumption of physical machines in the scheduling problems is equivalent to minimizing the sum of total busy time of all active physical machines that are homogeneous. Based on these observations, we proposed ETRE algorithm to solve the scheduling problems. The ETRE algorithm's swapping step swaps an allocating VM with a suitable overlapped VM, which is of the same VM type and is allocated on the same physical machine, to minimize total busy time of all physical machines. The ETRE uses resource utilization during executing time period of a physical machine as the evaluation metric, and will then choose a host that minimizes the metric to allocate a new VM. In addition, this work studies some heuristics for sorting the list of virtual machines (e.g., sorting by the earliest starting time, or the longest duration time first, etc.) to allocate VM. Using log-traces in the Feitelson's Parallel Workloads Archive, our simulation results show that the ETRE algorithm could reduce total energy consumption average by 48% compared to power-aware best-fit decreasing (PABFD [6]) and 49% respectively to vector bin-packing norm-based greedy algorithms (VBP-Norm-L1/L2 [15]).

Keywords: IaaS cloud · Virtual machine scheduling · Energy efficiency · Cloud computing · Total busy time · Fixed interval

1 Introduction

Cloud computing, which enables Infrastructure-as-a-Service (IaaS), provides users with computing resources in terms of virtual machines (VMs) to run their applications [4, 5, 10, 14, 18]. Infrastructure of cloud systems are built from virtualized data centers with thousands of high-performance computing servers

[4, 5, 18]. Power consumption in these large-scale data centers requires multiple megawatts [9, 14]. Le et al. [14] estimate the energy cost of a single data center is more than \$15M per year. As these data centers scale, they will consume more energy. Therefore, advanced scheduling techniques for reducing energy consumption of these cloud systems are highly concerned for any cloud providers to reduce energy cost. Increasing energy cost and the need to environmental sustainability address energy efficiency is a hot research topic in cloud systems. Energy-aware scheduling of VMs in IaaS cloud is still challenging [10, 14, 17, 19, 20].

Much previous work [5, 6, 15] showed that the virtual machine allocation is NP-Hard. There are several studies that have been proposed to address the problem of energy-efficient scheduling of VMs in cloud data centers. A number of [5, 6, 15] current techniques for consolidating virtual machines in cloud data centers use bin-packing heuristics (such as First-Fit Decreasing [15], and/or Best-Fit Decreasing [6]). They attempt to minimize the number of running physical machines and to turn off as many idle physical machines as possible. Consider a d -dimensional resource allocation where each user requests a set of virtual machines (VMs). Each VM requires multiple resources (such as CPU, memory, and IO) and a fixed quantity of each resource at a certain time interval. Under this scenario, using a minimum of physical machines may not be a good solution. Our observations show that using a minimum number of physical machines is not necessarily a good solution to minimize total energy consumption. In a homogeneous environment where all physical servers are identical, the power consumption of each physical server is linear to its CPU utilization, i.e., a schedule with longer working time (i.e. total busy time) will consume more energy than another schedule with shorter working time (i.e. total busy time).

Table 1. Example of given six virtual machines (VMs) with their normalized (*) resource demands

VM ID	CPU*	RAM*	Network*	Start-time	Duration (hour)
VM1	0.5	0.1	0.2	0	10
VM2	0.5	0.5	0.2	0	2
VM3	0.2	0.4	0.2	0	1
VM4	0.2	0.4	0.2	0	1
VM5	0.1	0.1	0.1	0	1
VM6	0.5	0.5	0.2	1	9

To the best of our knowledge, our work is the first work that studies increasing time and resource efficiency-based approach to allocate VMs onto physical machines in order that it minimizes the total energy consumption of all physical machines. Each VM requests resource allocation in a fixed starting time and non-preemption for the duration time. We present here an example to demonstrate our ideas to minimize total energy consumption of all physical machines in the

VM placement with fixed starting time and duration time. For example, given six virtual machines (VMs) with their normalized resource demands (CPU*, RAM* and Network* are normalized demand resources to physical server's maximum total capacity resources) described in Table 1. In the example, a bin-packing-based algorithm could result in a schedule S_1 in which two physical machines are used: one for allocating VM1, VM3, VM4, and VM5; and another one for allocating VM2 and VM6. The schedule S_1 has total busy time of the six VMs is $(10 + 9) = 19$ hours. However, in another schedule S_2 in which VMs are placed on three physical machines, VM1 and VM6 on the first physical machine, VM3, VM4 and VM5 on the second physical machine, and VM2 on the third physical machine, then the schedule S_2 has total busy time of the six VMs is only $(10 + 1 + 2) = 13$ hours.

In this paper, we propose a heuristic, namely, ETRE. ETRE heuristic places VMs that request multiple resources in the fixed interval time and non-preemption into physical machines to minimize total energy consumption of physical machines while meeting all resource requirements. Using numerical simulations, we compare the ETRE with the popular modified best-fit decreasing (PABFD) [6], two vector bin-packing norm-based greedy (VBP-Norm-L1/L2) [15], and our previous algorithms (e.g. EPOBF-ST/FT [17], and MinDFT-ST/FT [16]). Using real log-trace (i.e., [1]) in the Feitelson's Parallel Workloads Archive, our simulation results show that the ETRE heuristic with its configurations could reduce total energy consumption average by 48% compared to power-aware best-fit decreasing (PABFD) [6] and 49% respectively to vector bin-packing norm-based greedy algorithms (VBP-Norm-L1/L2 [15]). Additionally, ETRE-ST/LFT/LDTF have also less total energy consumption than our previous heuristics (e.g. MinDFT-ST/FT and EPOBF-ST/FT) in the simulations.

The remainder of the paper is organized as follows. Section 2 describes the energy-aware VM allocation problem with multiple requested resources, fixed starting and duration time. We also formulate the objective of scheduling. The proposed ETRE algorithm is presented in Sect. 3. In Sect. 4 we discuss our performance evaluation using simulations. In Sect. 5 we review the related work. In Sect. 6 we conclude this paper and introduce future work.

2 Problem Description

2.1 Notations

We use the following notations in this paper:

vm_i : The i^{th} virtual machine to be scheduled.

M_j : The j^{th} physical server.

S : A feasible schedule.

P_j^{idle} : Idle power consumption of the M_j .

P_j^{max} : Maximum power consumption of the M_j .

$P_j(t)$: Power consumption of the (M_j) at a time point t .

ts_i : Fixed starting time of vm_i .

dur_i : Duration time of vm_i .

T : Maximum schedule length, which is the time that the last virtual machine will be finished.

$n_j(t)$: Set of indexes of all virtual machines that are assigned to the M_j at time t .

T_j : Total busy time (working time) of the M_j .

e_i : Energy consumption for running the vm_i in the physical machine that the vm_i is allocated.

2.2 Power Consumption Model

In this paper, we use the following energy consumption model proposed in [9] for a physical machine. The power consumption of the M_j , denoted as $P_j(\cdot)$, is formulated as follow:

$$P_j(t) = P_j^{idle} + (P_j^{max} - P_j^{idle})U_j(t) \quad (1)$$

The CPU utilization of the physical server at time t , denoted as $U_j(t)$, is defined as the average percentage of total of allocated computing powers of $n_j(t)$ VMs that is allocated to the M_j . We assume that all cores in CPU are homogeneous, i.e. $\forall c = 1, 2, \dots, PE_j : MIPS_{j,c} = MIPS_{j,1}$, The CPU utilization is formulated as follow:

$$U_j(t) = \left(\frac{1}{PE_j \times MIPS_{j,1}} \right) \sum_{c=1}^{PE_j} \sum_{i \in n_j(t)} mips_{i,c} \quad (2)$$

The energy consumption of the server in a period of $[t_1, t_2]$ is formulated as follow:

$$E_j = \int_{t_1}^{t_2} P_j(U_j(t))dt \quad (3)$$

where:

$U_j(t)$: CPU utilization of the M_j at time t and $0 \leq U_j(t) \leq 1$.

PE_j : Number of processing elements (i.e. cores) of the M_j .

$mips_{i,c}$: Allocated MIPS of the c^{th} processing element to the vm_i by the M_j .

$MIPS_{j,c}$: Maximum capacity computing power (Unit: MIPS) of the c^{th} processing element on the M_j .

2.3 Problem Formulation

Given a set of virtual machines vm_i ($i = 1, 2, \dots, n$) to be scheduled on a set of physical servers M_j ($j = 1, 2, \dots, m$). Each VM is represented as a d-dimensional vector of demand resources, i.e. $vm_i = (x_{i,1}, x_{i,2}, \dots, x_{i,d})$. Similarly, each physical machine is denoted as a d-dimensional vector of capacity

resources, i.e. $M_j = (y_{j,1}, y_{j,2}, \dots, y_{j,d})$. We consider types of resources such as processing element (core), computing power (Million instruction per seconds-MIPS), physical memory (RAM), network bandwidth (BW), and storage. Each vm_i is started at a fixed starting time (ts_i) and is non-preemptive during its duration time (dur_i).

We assume that the power consumption model is linear to CPU utilization. Even if all physical servers are identical and all VMs are identical too, the scheduling is still NP-hard with $d \geq 1$ [15]. With the problem considered in this paper, all physical servers are identical and their power consumption models are linear to their CPU utilization as can be seen in the two Eqs. (1) and (3). The energy consumption of a physical server in a time unit is denoted as E_0 and is the same for all physical servers since the servers are identical. The objective is to find out a feasible schedule S that minimizes the total energy consumption in the Eq. (4) with $i \in \{1, 2, \dots, n\}$, $j \in \{1, 2, \dots, m\}$, $t \in [0; T]$ as following:

$$\text{Minimize } (E_0 \times \sum_{j=1}^m T_j + \sum_{i=1}^n e_i) \quad (4)$$

$$\text{Minimize } (E_0 \times \sum_{j=1}^m T_j + \sum_{i=1}^n e_i) \sim \text{Minimize } (\sum_{j=1}^m T_j) \quad (5)$$

where the total busy time (working time) of a physical server, denoted as T_j , is defined as union of interval times of all VMs that are allocated to a physical machine M_j at time T .

The scheduling problem has the following hard constraints that are described in our previous work [16].

3 Heuristic Based Scheduling Algorithm

In this section, we present our energy-aware scheduling algorithm, namely, ETRE (Energy-aware using increasing Time and Resource Efficiency metric). ETRE presents a performance metric to unify the increasing time and estimated resource efficiency when mapping a new VM onto a physical machine. Then, ETRE will choose a host that has the minimum of the metric. Our previous MinDFT-ST/FT [16] only focused on minimizing the increasing time when mapping a new VM onto a physical machine. The ETRE additionally considers resource efficiency during an execution period of a physical machine in order to fully utilize resources in a physical machine. Furthermore, the core ETRE algorithm can swap an overlapped VM, which has already been assigned to an active physical machine before, with a new VM to minimize total busy time of the physical machine. In this paper, two VMs are overlapped if $ts_1 < ts_2 < (ts_1 + dur_1) < (ts_2 + dur_2)$, where ts_1 , ts_2 , dur_1 , dur_2 are starting times and duration times of two VMs. The core ETRE algorithm will swap a new VM and its overlapped VM together if two VMs meet these conditions: (i) both VMs are of the same VM type (i.e. the same amount of requested resources such

as number of CPU core, physical memory, network bandwidth, storage, etc.); (ii) the new VM has duration time longer than its overlapped VM. Neither our previous MinDFT-ST/FT [16] and the EPOBF-ST/FT [17] have these swapping steps.

Based on Eq. 2, the utilization of a resource r (resource r can be CPU, physical memory, network bandwidth, storage, etc.) of a PM j -th, denoted as $U_{j,r}$, is formulated as follow:

$$U_{j,r} = \sum_{s \in n_j} \frac{V_{s,r}}{H_{j,r}}. \quad (6)$$

where n_j is the list of VMs that are assigned to the physical machine j , $V_{s,r}$ is the amount of the requested resource r of the virtual machine s (note that in our study the value of $V_{s,r}$ is fixed for each user request), and $H_{j,r}$ is the maximum capacity of the resource r in the physical machine j .

Inspired by the work from Microsoft research team [8, 15], resource efficiency of a physical machine j -th, denoted by RE_j , is Norm-based distant [15] of two vectors: normalized resource utilization vector and unit vector $\mathbf{1}$, the resource efficiency is formulated as follow:

$$RE_j = \sum_{r \in \mathcal{R}} ((1 - U_{j,r}) \times w_r)^2 \quad (7)$$

where $\mathcal{R} = \{\text{cpu, ram, netbw, io, storage}\}$: set of resource types in a host, w_r is the weight of resource r in a physical machine.

In this paper, we propose a unified metric for increasing time and resource efficiency that is calculated as:

$$TRE = (t^{diff} \times w_{r=time})^2 + \sum_{r \in \mathcal{R}} ((1 - U_{j,r}) \times w_r)^2 \quad (8)$$

where: t^{diff} is the increasing time (Unit: hours) of total busy time of all physical machines before and after allocating a new VM to this host; and $w_{r=time}$ is weight of time in the TRE metric.

ETRE chooses a physical host that has a minimum value of the TRE metric to allocate a VM. The ETRE can sort the list of VMs by earliest starting time first, or earliest finishing time first, or longest duration time first, etc. The ETRE solves the scheduling problem in the time complexity of $\mathcal{O}(n \times m \times q)$ where n is the number of VMs to be scheduled, m is the number of physical machines, and q is the maximum number of allocated VMs in the physical machines $M_j, \forall j = 1, 2, \dots, m$.

4 Performance Evaluation

4.1 Algorithms

In this section, we study the following VM allocation algorithms:

- PABFD, a power-aware and modified best-fit decreasing heuristic [5,6]. The PABFD sorts the list of VM_i ($i=1, 2, \dots, n$) by their total requested CPU utilization and assigns new VM to any host that has a minimum increase in power consumption.
- VBP-Norm-LX, a family of vector packing heuristics that is presented as Norm-based Greedy with degree $X=1, 2$ [15]. Weights of these Norm-based Greedy heuristics use FFDAvgSum which are $\exp(x)$, which is the value of the exponential function at the point x , where x is the average sum of demand resources (e.g. CPU, memory, storage, network bandwidth, etc.). VBP-Norm-LX assigns a new VM to any host that has minimum of these norm values.
- EPOBF-ST and EPOBF-FT, which is presented in [17], sort the list of VM_i ($i=1, 2, \dots, n$) by their starting time (ts_i) and respectively by their finished time ($ts_i + dur_i$). Both EPOBF-ST and EPOBF-FT choose a host that has the maximum performance-per-watt to assign a new VM. The performance-per-watt is the ratio of the total maximum capacity MIPS and the maximum host's power consumption.
- MinDFT-ST and MinDFT-FT, which is presented in [16], sort the list of VM_i ($i=1, 2, \dots, n$) by their starting time (ts_i) and respectively by their finished time ($ts_i + dur_i$). Both MinDFT-ST and MinDFT-FT allocate each VM (in a given set of VMs) to a host that has a minimum increase in the total completion time of hosts.
- ETRE, our proposed algorithm discussed in Sect. 3. We evaluate the ETRE with some of its configurations: The ETRE-ST sorts the list of virtual machines by VM's earliest starting time first and host's allocated VMs by its finishing times. The finishing time of a virtual machine, which is sum of its starting time and its duration time, is calculated by ($ts_i + dur_i$). The ETRE-LDTF sorts the list of virtual machines by VM's longest duration time first and host's allocated VMs by its finishing time. The ETRE-LFT sorts the list of virtual machines by VM's latest finishing time first and host's allocated VMs by its finishing time.

4.2 Simulated Simulations

We evaluate these algorithms by simulations using the CloudSim [7] to create a simulated cloud data center system that has identical physical machines, heterogeneous VMs, and with thousands of CloudSim's cloudlets [7] (we assume that each HPC job's task is modeled as a cloudlet that is run on a single VM). The information of VMs (and also cloudlets) in these simulated workloads is extracted from a real log-trace (HPC2N Seth log-trace [1]) in Feitelson's Parallel Workloads Archive (PWA) [2] to model HPC jobs. When converted from the log-trace, each cloudlet's length is a product of the system's processing time and CPU rating (we set the CPU rating which is equal to included VM's MIPS). We convert job's submission time, job's start time (if the start time is missing, then the start time is equal to the sum of job's submission time and job's waiting time), job's request run-time, and job's number of processors in job data from the log-trace in the PWA to VM's submission time, starting time and duration

Table 2. Eight (08) VM types in simulations

VM Type	MIPS	Cores	Memory (MBytes)	Net. Bw (Mbits/s)	Storage (GBytes)
Type 1	2500	8	6800	100	1000
Type 2	2500	2	1700	100	422.5
Type 3	3250	8	68400	100	1000
Type 4	3250	4	34200	100	845
Type 5	3250	2	17100	100	422.5
Type 6	2000	4	15000	100	1690
Type 7	2000	2	7500	100	845
Type 8	1000	1	1875	100	211.25

time, and the number of VMs (each VM is created in a round-robin manner with the 8 types of VMs, which can be seen in Table 2 on the number of VMs). Eight types of VMs as presented in the Table 2 are similar to categories in Amazon EC2's VM instances: high-CPU VM, high-memory VM, etc. Fig. 1 shows the chart of starting times and finishing times of the VMs in a simulation (the simulations have the same starting times and duration times of VMs). All physical machines are identical machines. Each physical machine has system information and its power consumption as in Table 3. In the simulations, we use weights as following: (i) the weight of increasing time of mapping a VM to a host: {0.001, 0.01, 1, 100, 3600}; (ii) weights of computing resources such as the number of MIPS per CPU core, physical memory (RAM), network bandwidth, and storage are 940, 24414, 1, 0.0001 respectively. We simulate on the combination of these weights. The total energy consumption of ETRE-ST, ETRE-LFT and ETRE-LDTF are the average of five times simulation with various weights of increasing time (e.g. 0.001, 0.01, 1, 100, or 3600) (Fig. 2).

Table 3. System information of a typical physical machine and its maximum and idle power consumption (P^{max} and P^{idle}) (ratio of P^{idle} and P^{max} is 0.7).

Host Type	MIPS	Cores	Memory (MB)	Network (Mb/s)	Storage (GB)	P^{max}	P^{idle}
M1	3250	16	140084	10000	10000	600	420

We choose PABFD [6] as the baseline algorithm because the PABFD is a famous power-aware best-fit decreasing in the energy-aware scheduling research community. We also compare our proposed VM allocation algorithms with two vector bin-packing algorithms (VBP-Norm-L1/L2) to show the importance of with/without considering VM's starting time and finish time in reducing the total energy consumption of VM placement problem.

Table 4. Result of simulations using the first 400 jobs of the HPC2N Seth log-trace [1]. The normalized energy (Nor. Energy) column is the normalized energy. The energy saving is percentage of energy consumption of an algorithm which reduces in comparison with the energy consumption of PABFD.

Algorithm	#Hosts	#VMs	Energy (KWh)	Nor. Energy	Saving (%)
PABFD	5000	7495	7071.50	1.00	0 %
VBP-Norm-L1	5000	7495	7114.74	1.01	-1 %
VBP-Norm-L2	5000	7495	7114.74	1.01	-1 %
EPOBF-ST	5000	7495	4454.77	0.63	37 %
EPOBF-FT	5000	7495	4409.15	0.62	38 %
MinDFT-ST	5000	7495	4534.36	0.64	36 %
MinDFT-FT	5000	7495	4409.15	0.62	38 %
ETRE-ST	5000	7495	4070.07	0.58	42 %
ETRE-LFT	5000	7495	3418.07	0.48	52 %
ETRE-LDTF	5000	7495	3451.53	0.49	51 %

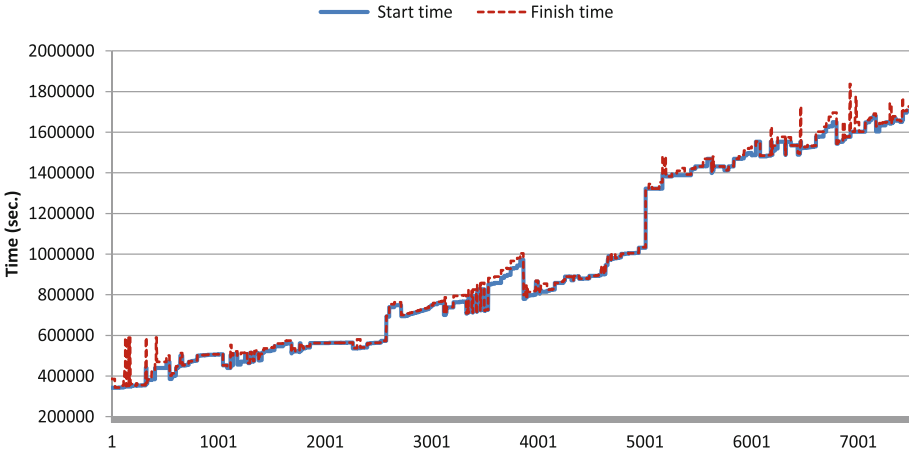


Fig. 1. Starting time (blue line) and finishing time (dotted red line) of VMs in simulations with HPC2N Seth log-trace [1] (Color figure online).

4.3 Results and Discussions

Table 4 shows simulation results of scheduling algorithms solving scheduling problems with 7,495 VMs and 5,000 physical machines (hosts), in which VM's data is converted from the first 400 jobs in the HPC2N Seth log-trace [1]. None of the algorithms use VM migration techniques, and all of them satisfy the Quality of Service (i.e. allocates all resources that user VM requested on-time). We use total energy consumption as the performance metrics for evaluating these VM allocation algorithms. The energy saving shown in both Table 4 is the reduction

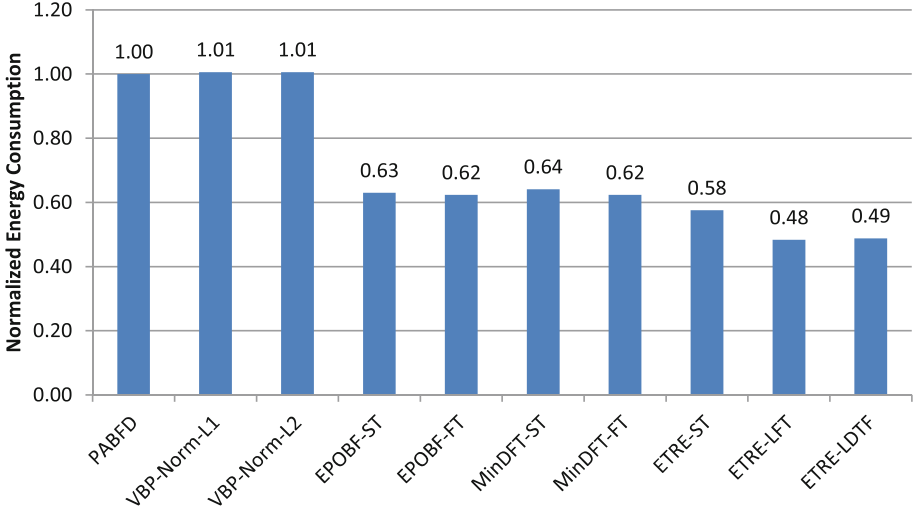


Fig. 2. Normalized energy. Result of simulations with HPC2N Seth log-trace.

of total energy consumption of the corresponding algorithm compared to the baseline PABFD [6] algorithm.

Table 4 shows that our ETRE-ST/LFT/LDTF compared to PABFD [6], can reduce the total energy consumption on the average by 48 %, and 49 % respectively compared to norm-based vector bin-packing algorithms (VBP-Norm-L1/L2) in simulations with the first 400 jobs of the HPC2N Seth log-trace.

The PABFD generates a schedule that uses higher energy consumption than the ETRE-ST/LFT/LDTF because of the following main reasons. First, our hypothesis in this paper is that each VM consumes the same amount of energy in any physical server (e_i) and all physical servers are identical. As a consequence, the PABFD will choose a random physical server to map a new VM. The PABFD sorts the list of VMs by decreasing the requested computing power (e.g. MIPS), therefore the PABFD allocates VMs that firstly have the most requested computing power. In Table 2, all type-3 VMs have the highest requested computing power in the list, the next is a type-1 VM, etc. Instead, our proposed ETRE-ST/LFT/LDTF algorithms assign a new VM to a physical server in such a way that has minimum increase of total busy time of all physical machines and use fully all resources in physical machines.

These ETRE-ST, ETRE-LFT and ETRE-LDTF algorithms perform better than our previous algorithms such as MinDFT-ST/FT and EPOBF-ST/FT in the simulations. Compared to EPOBF-ST and EPOBF-FT, the ETRE-ST, ETRE-LFT and ETRE-LDTF have less total energy consumption on the average by 18 % and 17 % respectively. The ETRE-ST, ETRE-LFT and ETRE-LDTF have also less total energy consumption than the MinDFT-ST and MinDFT-FT on the average by 20 % and 17 %, respectively. In the simulations, swapping between a new VM and its overlapped VM that is allocated to a host reduce

total busy time on the host. For input as in Table 1, the VM2 is removed from the first host, the VM6 will be allocated to the first host.

5 Related Work

Many previous researches [5,6,8,12,19] proposed algorithms that consolidate VMs onto a small set of physical machines (PMs) in virtualized datacenters to minimize energy/power consumption of PMs. Much work has considered the VM placement problem as a bin-packing problem, and have used bin-packing heuristics to place VMs onto a minimum number of PMs to minimize the energy consumption [5,6]. Beloglazov et al. [5,6] have proposed VM allocation problem as bin-packing problem and presented a power-aware best-fit decreasing (denoted as PABFD) heuristic. PABFD sorts all VMs in a decreasing order of CPU utilization and tends to allocate a VM to an active physical server that would take the minimum increase of power consumption. A group in Microsoft Research [15] has studied first-fit decreasing (FFD) based heuristics for vector bin-packing to minimize number of physical servers in the VM allocation problem. Some other work also proposed meta-heuristic algorithms to minimize the number of physical machines. A hill-climbing based allocation of each independent VM is studied in [11]. In the VM allocation problem, however, minimizing the number of used physical machines is not equal to minimizing the total energy consumption of all physical machines.

Takouna et al. [19] presented power-aware multicore scheduling and their VM allocation algorithm selects a host which has the minimum increasing power consumption to assign a new VM. The VM allocation algorithm, however, is similar to the PABFDs [6] except that it concerns memory usage in a period of estimated runtime for estimating the host's energy. The work also presented a method to select optimal operating frequency for a (DVFS-enabled) host and configure the number of virtual cores for VMs. Our proposed ETRE algorithm that is different from these previous work. Our ETRE algorithm uses the VM's fixed starting time and duration time to minimize the total working time on physical servers, and consequently minimize the total energy consumption in all physical servers.

In 2007, Kovalyov et al. [13] presented a work to describe characteristics of a fixed interval scheduling problem in which each job has fixed starting time, fixed processing time, and is only processed in the fixed duration time on a available machine. The scheduling problem can be applied in other domains. Angelelli et al. [3] considered interval scheduling with a resource constraint in parallel identical machines. The authors proved the decision problem is NP-complete if the number of constraint resources in each parallel machine is a fixed number greater than two.

6 Conclusions and Future Work

In this paper, we formulated an energy-aware VM allocation problem with fixed starting time and non-preemption. We also discussed our two key observations

in the VM allocation problem. First, minimizing total energy consumption is equivalent to minimizing the sum of total busy time of all physical machines (PMs). For some possible schedules, which have same ETRE-ST/LFT/LDTF of all PMs, the TRE metric decides a schedule that has higher resource efficiency. Second, swapping between an unallocated VM and its overlapped VM, which has already been allocated to a PM, can reduce the ETRE-ST/LFT/LDTF of all PMs. Based on these observations, we proposed ETRE algorithm to solve the energy-aware VM allocation with fixed starting time and duration time.

Our proposed ETRE can reduce the total energy consumption of the physical servers compared with that of other algorithms in simulation results on the HPC2N Seth [1] in the Feitelson's PWA [2]. The combination of ETRE with its sorting list of virtual machines by latest finishing time first (ETRE-LFT) has the least total energy consumption in these simulations.

In future, we are developing ETRE into a cloud resource management software (e.g. OpenStack Nova Scheduler). We are also working on IaaS cloud systems with heterogeneous physical servers and job requests consisting of multiple VMs. Moreover, we will study how to choose the right weights of time and resources (e.g. computing power, physical memory, network bandwidth, etc.) in another paper.

Acknowledgment. This research was conducted within the “Studying and developing practical heuristics for energy-aware virtual machine-based lease scheduling problems in cloud virtualized data centers” sponsored by TIS, and a fund by HCMUT (under the grant number T-KHMT-2015-33). As an Erasmus Mundus Gate project’s PhD student at The Johannes Kepler University (JKU) Linz, I am thankful to Prof. Dr. Josef Kueng as supervisor. I am also thankful to all reviewers.

References

1. The HPC2N Seth log-trace (HPC2N-2002-2.2-cln.swf.gz file). http://www.cs.huji.ac.il/labs/parallel/workload/1_hpc2n/HPC2N-2002-2.2-cln.swf.gz. Accessed 1 May 2015
2. Feitelson’s Parallel Workloads Archive. <http://www.cs.huji.ac.il/labs/parallel/workload/>. Accessed 31 Januray 2014
3. Angelelli, E., Filippi, C.: On the complexity of interval scheduling with a resource constraint. *Theoret. Comput. Sci.* **412**(29), 3650–3657 (2011). <http://www.sciencedirect.com/science/article/pii/S0304397511002623>
4. Barroso, L.A., Clidaras, J., Hölzle, U.: The datacenter as a computer: an introduction to the design of warehouse-scale machines. *Synth. Lect. Comput. Architect.* **8**(3), 1–154 (2013)
5. Beloglazov, A., Abawajy, J., Buyya, R.: Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Gener. Comput. Syst.* **28**(5), 755–768 (2012)
6. Beloglazov, A., Buyya, R.: Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurrency Comput. Pract. Exper.* **24**(13), 1397–1420 (2012)

7. Calheiros, R.N., Ranjan, R., Beloglazov, A., De Rose, C.A.F., Buyya, R.: Cloudsim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. *Softw. Pract. Exper.* **41**(1), 23–50 (2011)
8. Chen, L., Shen, H.: Consolidating complementary VMs with spatial/temporal-awareness in cloud datacenters. In: *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 1033–1041. IEEE, April 2014. <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6848033>
9. Fan, X., Weber, W.D., Barroso, L.: Power provisioning for a warehouse-sized computer. In: *ISCA*, pp. 13–23 (2007)
10. Garg, S.K., Yeo, C.S., Anandasivam, A., Buyya, R.: Energy-efficient scheduling of HPC applications in cloud computing environments. CoRR abs/0909.1146 (2009)
11. Goiri, I., Julia, F., Nou, R., Berral, J.L., Guitart, J., Torres, J.: Energy-aware scheduling in virtualized datacenters. In: *2010 IEEE International Conference on Cluster Computing*, pp. 58–67. IEEE, September 2010. <http://doi.ieeecomputersociety.org/10.1109/CLUSTER.2010.15>, <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5600320>
12. Knauth, T., Fetzer, C.: Energy-aware scheduling for infrastructure clouds. In: *4th IEEE International Conference on Cloud Computing Technology and Science Proceedings*, pp. 58–65. IEEE, December 2012, <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6427569>
13. Kovalyov, M.Y., Ng, C., Cheng, T.E.: Fixed interval scheduling: models, applications, computational complexity and algorithms. *Eur. J. Oper. Res.* **178**(2), 331–342 (2007)
14. Le, K., Bianchini, R., Zhang, J., Jaluria, Y., Meng, J., Nguyen, T.D.: Reducing electricity cost through virtual machine placement in high performance computing clouds. In: *SC*, p. 22 (2011)
15. Panigrahy, R., Talwar, K., Uyeda, L., Wieder, U.: Heuristics for vector bin packing. Technical report, Microsoft Research (2011)
16. Quang-Hung, N., Le, D.-K., Thoai, N., Son, N.T.: Heuristics for energy-aware VM allocation in HPC clouds. In: Dang, T.K., Wagner, R., Neuhold, E., Takizawa, M., Küng, J., Thoai, N. (eds.) *FDSE 2014*. LNCS, vol. 8860, pp. 248–261. Springer, Heidelberg (2014)
17. Quang-Hung, N., Thoai, N., Son, N.T.: EPOBF: energy efficient allocation of virtual machines in high performance computing cloud. In: Hameurlain, A., Küng, J., Wagner, R., Thoai, N., Dang, T.K. (eds.) *TLDKS XVI*, LNCS 8960. LNCS, vol. 8960, pp. 71–86. Springer, Heidelberg (2015). http://link.springer.com/10.1007/978-3-662-45947-8_6
18. Sotomayor, B.: Provisioning computational resources using virtual machines and leases. Ph.D. thesis, University of Chicago (2010)
19. Takouna, I., Dawoud, W., Meinel, C.: Energy efficient scheduling of HPC-jobs on virtualize clusters using host and VM dynamic configuration. *Oper. Syst. Rev.* **46**(2), 19–27 (2012)
20. Viswanathan, H., Lee, E.K., Rodero, I., Pompili, D., Parashar, M., Gamell, M.: Energy-aware application-centric VM allocation for HPC workloads. In: *IPDPS Workshops*, pp. 890–897 (2011)