Live Migration Overhead-Aware Dynamic VM Consolidation Algorithm in Cloud Computing

Salma K. Elbay, Islam Hegazy, El-Sayed M. El-Horbaty

Computer Science Department, Faculty of computer and information sciences,
Ain Shams University, Cairo, Egypt
salma.elbay, islheg, shorbaty@cis.asu.edu.eg

Abstract

Energy Efficiency has become a crucial concern in modern data centers. Dynamic VM consolidation is one of the effective approaches endorsed to achieve energy efficiency in cloud data centers hosting thousands of servers. Live migration is a core feature enabling VM consolidation. However, live migration is a costly operation imposing energy and performance overhead.

An efficient dynamic virtual machine consolidate should consider the cost due to live migration. In this paper, we design and implement a dynamic VM consolidation algorithm based on simulated annealing that accounts for the migration cost imposed by a consolidation plan. We conduct simulation-based experiments on CloudSim using real cloud workload traces from PlanetLab to evaluate the performance of the proposed algorithm. Results show that the using the proposed algorithm, the simulated data center consumes almost the same amount of energy of that using a FF based consolidation. However, the proposed algorithm accounts for the cost due to live migration and hence, reducing SLA violations and performance degradations.

Keywords: *Cloud Computing, live migration, migration overhead, dynamic consolidation, simulated annealing.*

1. Introduction

Cloud computing has become a disruptive technology transforming businesses by providing infrastructure and resources as services. Energy efficiency is a crucial concern in cloud data centers. The amount of energy consumed by the infrastructure, including servers, network devices, cooling equipment and others, highly affects the cloud provider's profit margin. It also has a major environmental impact due to the carbon emissions contributing to global warming.

Virtual machine consolidation is one of the fundamental approaches to achieve energy efficiency in cloud data centers. It involves packing all available Virtual Machines (VMs) on an optimal number of Physical Machines (PMs). The goal is to free underutilized servers, turn them into sleep mode and hence, save their operational power. Servers consume 80% of their peak power even at 20% utilization [1]. Thus, turning a physical server into a low-power state mode can significantly reduce the total data center's energy consumption.

VM consolidation can be static or dynamic. A static VM consolidation (SVMC) algorithm's inputs are: a set of all available VMs and a set of all available PMs. The output of a SVMC algorithm is a VM-PM assignment map defining the allocation of every VM to one of the given PMs. SVMC algorithms are usually useful at the beginning to generate an initial placement for VMs on a set of empty PMs. However, A dynamic VM consolidation (DVMC) algorithm doesn't ignore the dynamic nature of a cloud environment, in terms of changing VM resource demand of assigned workloads. It redistributes active VMs to achieve a certain objective, as energy saving or load balancing. Accordingly, the VM-PM allocation can be changed in response to the varying resource requirements of a VM caused by the dynamic workload running on that VM.

To achieve energy saving in cloud data centers, both types of VM consolidation algorithms are required. A SVMC algorithm is needed at the beginning to provide an initial placement plan for the set of VMs on the set of available PMs. This algorithm should be energy-aware such that it tends to minimize the total number of active PMs. Yet, workloads assigned to VMs in a cloud environment are dynamic. Resource requirements vary over time and thus, the allocation of VMs on the servers need to be re-evaluated on periodic basis to check if further energy saving could be achieved through freeing some PMs. Hence, an online DVMC algorithm need to be running in the background providing periodic migration plans to keep the energy consumed by PMs at minimum through optimizing the utilization of active PMs. The relocation of a VM is typically accomplished through live migration operation. In this work, we focus on the dynamic VM consolidation problem.

In a live migration operation, the VM is physically moved from one server to another while still being up and running with minimal interruption. Live migration is a core feature in modern data centers. However, live migration is a costly operation that requires additional CPU cycles at both, the source host and the destination host. According to Strunk [2], the cost of migration can be broadly divided into two categories; energy cost and performance cost. The overhead due to migration should not be ignored during dynamic VM consolidation decisions. A migration that will be less costly in terms of energy and performance shall be favored on another migration introducing higher overhead.

Achieving energy efficiency through DVMC is an NP-Hard optimization problem. Exact, heuristic and metaheuristic methods have been used to solve the DVMC problem in the literature. However, metaheuristic approaches draw most of the research interest. This is because metaheuristic approaches, as Ant Colony Optimization (ACO), Artificial Bee Colony (ABC) and Firefly, provide near-optimal solutions in reasonable running time.

In this paper, we address the DVMC problem using Simulated Annealing algorithm [3]; a metaheuristic approach for solving discrete optimization problems.

The rest of the paper is organized as follows: In section 2, we summarize related research work. Then, we discuss the VM consolidation model with emphasis to formulating the problem, describing system architecture and we shed light on the proposed live migration overhead estimation model in section 3. In section 4, we propose our migration overhead-aware simulated annealing-based DVMC algorithm. In section 5, we discuss the experiments and results. Finally, we give concluding remarks and outline some possible future research directions in section 6.

2. Related Work

Achieving energy efficiency through DVMC is an NP-Hard optimization problem. Exact methods, such as dynamic programming and linear programming, have been used in the literature. They provide optimal solutions for the DVMC problem. However, they are expensive algorithms given the typical large number of VMs and PMs in cloud environments.

Heuristic algorithms are problem-specific methods providing approximate near-optimal solutions, such as First Fit (FF) and Best Fit (BF). Unlike metaheuristic algorithms, heuristic algorithms are problem-specific algorithms that can provide an approximate solution to the NP-Hard DVMC problem. One limitation of heuristic algorithms is that they can be trapped in local minimum due to their greedy nature, and hence fail to provide a global optimal solution. Metaheuristics can take the form of evolutionary algorithms such as Ant Colony [4], Genetic Algorithms [5], Artificial Bee Colony [6], [7]. These approaches show promising results in terms of running cost and providing near optimal solutions.

Feller et al. [4] propose the Ant Colony Optimization (ACO) algorithm to solve the Multidimensional bin-packing problem (MDBP) in the context of dynamic workload consolidation. Authors used their own java-based toolkit EnaCloud as a cloud simulation environment. ACO algorithm is compared to classical greedy FFD algorithm. Authors suggests that the workload placement is to be done initially using FFD algorithm because it has low computation time. And then, on periodic-basis, daily or weekly for example, the proposed ACO algorithm is used to optimize the placements.

Kansal et al [8] suggest using a Firefly based optimization approach. They attempt to find the best VM-Host pair where through the proposed FireFly Optimization Energy-aware Virtual Machine Migration (FFO-EVMM) approach to maximize energy efficiency through the optimum number of migrations. Farahnakian et al. [9] formulate the consolidation problem as a multi-objective combinatorial problem addressing the three conflicting objectives; reducing the energy consumption, keeping the number of VM migrations minimal and avoiding SLA violations. Ant Colony System (ACS) is used as a meta-heuristic algorithm, where the amount of pheromone produced by the ants accumulate on the preferred migration plans. Wu et al. [5] regards the consolidation problem as a grouping problem, whose objective is to divide a set of VMs into a number of disjoint subsets, representing PMs. They propose a score function for to evaluate both; a migration cost estimation method and an upper bound estimation method for maximum power saving. An improved version for the classical grouping genetic algorithm is proposed and applied to the consolidation problem.

Adhianto et al. [10] formulated the DVMC problem as an NP-hard, discrete optimization problem. They proposed an algorithm based on Ant Colony Optimization (ACO) metaheuristic approach to solve the problem. Mathematical models were introduced to estimate the energy cost due to migration, the migration overhead and SLA violation due to migration. The proposed algorithm is compared to other migration-aware DVMC algorithms and is claimed to reduce the overall power consumption by approximately 47%.

Marzolla et al. [11] propose V-Man, a DVMC algorithm based on a simple gossiping protocol. The algorithm maintains an overlay network in which all the active hosts are organized. Every host, i.e. node, on the network can only interact with some of the network hosts, its neighbors. Each node periodically broadcasts messages about the number of hosted VMs on that node to its neighbors and receives messages about the number of VMs hosed by its neighbors. The node with the greater number of VMs migrates them to its neighboring node till the destination node reaches a predefined threshold for the maximum number of hosted VMs. However, this work assumes that all the hosted VMs are identical. A host utilization can be determined given only the number of hosted VMs. This is not valid in practice since VMs vary in terms of their sizes and resource demands. Moreover, the migration overhead is ignored.

Wang et al. [12] developed a VM Placement algorithm based on Particle Swarm Optimization (PSO). The algorithm is optimized for data-intensive workloads in National Cloud Data Centers (NCDC)s. Wang et al. do not assume server homogeneity. The proposed PSO-based algorithm is compared to the modified Best Fit Decreasing (MBFD) algorithm, First Fit (FF) algorithm and Best Fit (BF) algorithm. The algorithm is shown to be energy-efficient in tree-like-topology-networked datacenters. One limitation of this work is that it assumes that every VM executes a single service.

Although energy saving is the main objective for VM consolidation, most of the literature attempts neglect the cost of live migration.

Rybina et al. [13] propose a mathematical model to estimate the energy cost of migration. This is achieved through investigating the additional power drawn by the two hosts participating in the migration process as well as the time duration of the process. Authors suggest that the VM size, available network bandwidth and the type of workload are the parameters contributing to the energy cost of live migration. During migration, the power overhead is increasing for the source and destination servers as the network bandwidth is increased. The type of workload executing on the migrated VM is affecting the migration power consumption. Migrating VM size and available network bandwidth are found to be the parameters contributing to the total time of migration. A large VM migrates in more time than a smaller VM and providing a larger network bandwidth guarantees faster migration.

Strunk [14] similarly states that the energy overhead due to live migration varies according to two parameters: the RAM size of the VM to be migrated and the available network bandwidth. A series of experiments were carried out and the obtained data was used to build a model using linear regression. In this work, a mathematical model is proposed to estimate the energy cost of the live migration operation of a VM. The model is tested experimentally and is proved to achieve more than 90% estimation accuracy. A limitation of this work is that the developed model is only valid for idle VMs.

Rybina [15] et al. considered five resource utilization parameters: number of last level cache line misses during migration, number of instructions retired, CPU utilization, ratio of active memory utilized by the source server to data transmission rate and number of “dirty” pages observed in the source server during migration. This work approaches the problem differently. Rybina et al. used Multiple Linear Regression (MLR) techniques to model the dependence of the energy consumption during migration on the five resource parameters under test. First, it is shown that the energy consumption is linearly dependent on each of the considered parameters. Then, all subset regression method is used to find the best MLR model. The proposed experiment shows that the CPU instructions retired ,last level cache line misses and “dirty” memory pages observed on the source server during migration are the parameters having the highest significance in the energy consumption for memory intensive workloads.

3. VM Consolidation Model

A. Problem Formulation

A typical cloud data center hosts a number of heterogeneous PMs with varying configurations. The resource capabilities of each PM is represented by its CPU, bandwidth and RAM. A CPU can be multi-core and its performance is measured in Million Instructions Per Second (MIPS), the network bandwidth provided by the PM is measured in Kbit/sec and the RAM size is measured in MBytes. In a virtualized cloud environment, users’ workloads are assigned to VMs. VMs are allocated to PMs according to the provider's pre-defined allocation policy. An allocation policy should perform the VM placement on a PM with sufficient resource capabilities. Also, the allocation policy may optimize some objective, such as load balancing or power saving. However, due to the dynamic nature of cloud workloads, to maintain an optimal/near-optimal placement at all times, a monitoring algorithm should be running in the background to track the statuses of VMs and carry out migrations necessary to maintain optimality/near-optimality. A dynamic VM consolidation algorithm is used periodically to pack running VMs on an optimal number of PMs to save energy and improve PMs' utilizations.

An input to a DVMC algorithm is a PM-VM allocation map at a given time instance. The output shall be a new VM-PM migration plan providing a better placement in terms of the objective function to be optimized. The ultimate goal of a DVMC algorithm is to minimize the data center's energy consumption. The total energy consumed by a data center after applying a given migration plan is referred to as *E(MP)*. A given VM can be relocated to another PM through a live migration operation.

The overhead due to migration is a quantified cost estimated as *OM*, whose estimation approach is discussed later in the next subsection. The optimization should be for the weighted product of (*E* and *OM*) for a given migration plan MP, that is, minimize the total energy consumed by PMs while keeping the overhead due to migration at minimum.

Let *AllPMs* represent the set of available PMs. Each PM $p \in AllPMs$ has a resource capacity vector $RC = \{RC^{r}_{i}\}$, where each element in the set represents the resource capacity for PM i for the resource r. The considered resources include CPU, memory and network bandwidth. Each VM $vm \in AllVMs$ has a resource demand vector $RD = \{ RD^{r}_{j}\}$where each element in the set represents the resource demand of VM j for the resource r. The set of VMs hosted by PM i is given by $HVMs_i$. The resource utilization vector for each PM $P \in AllPMs$ is given by $RU = \{RU^{r}_{i}\}$, where each $RU_i \in RU$ represents the resource utilization of the ith PM for the resource r, and is calculated as the summation of resource demand of all the hosted VMs.

$$RU_p^r = \sum_{v \in HVMs} RD_v^r \quad (1)$$

The objective is to find a VM migration plan that minimizes energy consumption and minimizes the total migration overhead. This should be achieved without violating the PM resource capacity constraints. Given the above, the objective function can be formulated as:

$$\textbf{minimize} f(MP) = E(MP) * OM(MP) \quad (2)$$

where *MP* is the Migration Plan binary matrix whose cells are indexed with the VM id and PM id. The matrix cell value specifies the VM-PM allocation status, defined as:

$$MP_{vm,p} = \begin{cases} 1, & \text{if if vm is assigned to p} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

Overhead due to Migration (OM) is calculated using the mathematical model proposed in the next section. The proposed objective function is subject to the following constraints:

$$\sum_{v \in HVMs} RD_v^r \; MP_{vm,p} \leq RC_p^r \; \forall p \in AllPMs \; \forall r \in RC \tag{4}$$

$$\sum_{P=0}^{AllPMs} MP_{vm,p} = 1 \;\; for \;\; i = 0, ..., AllVMs \tag{5}$$

The first constraint (4) ensures that the summation of resource requirements for a given resource for all hosted VMs do not exceed the PM’s available resource capacity. The second constraint (5) guarantees that a VM can be assigned to one PM.

B. System Architecture

A DVMC problem can be divided into three main steps:

1) Determining the underloaded PMs
2) Selecting VMs on the chosen PM to be migrated out
3) Determining the migration destination host for each VM

First, the underloaded hosts are detected as possible candidates for consolidation. Host under-loading detection is usually carried out by comparing a host’s resource utilization to some threshold. This threshold can be fixed as proposed by Beloglazov et al. in [16], or dynamic as suggested by Deng et al. in [17]. When a host is selected as a possible source host for consolidation, VMs on that host are selected to be migrated to another host(s). Some approaches select one VM to be migrated at a time, based on least migration time for example, as proposed by Beloglazov in [18]. Finally, the set of VMs selected to be migrated are re-distributed among other hosts based on the destination host selection algorithm.

In this work, we use a static threshold-based approach for detecting underloaded hosts. For the VM selection, an all-or-none approach is adopted. Once a host is selected as a source PM, all its hosted VMs are selected to be migrated out. Finally, simulated annealing is used as a metaheuristic algorithm for destination host selection.

C. Live Migration Overhead Estimation

Live migration of VMs is the backbone of dynamic VM consolidation. However, it doesn’t come free of charge and hence shouldn’t be carried out excessively. Given a running VM currently allocated to a given source host A. It is required to migrate the VM to a target host B, that is ready to provide the VM with its required cpu, memory and bandwidth resources. Pre-copy algorithm is the most widely implemented algorithm for live migration. The algorithm runs in several rounds. In the first round, the entire memory contents of the VM at the source server is copied to the destination server. Since live migration involves moving the VM while it is still operating, further memory pages’ modifications for the VM at the source host are expected to occur during the first round execution time. These modified memory pages are tracked. In the next rounds, memory pages are re-copied from the source host to the destination host, but now only for the modified memory pages. A termination condition will be enforced to stop those rounds, this may be a predefined threshold for the maximum number of rounds or a minimum threshold for the memory volume difference between rounds. At this stage, called stop-and-copy phase, the VM is stopped and the dirtied memory pages are copied for one last time. Finally, the VM’s execution is resumed on the target host. In this work, we attempt to quantify the live migration cost. The following factors are assumed to be the primary factors contributing to migration cost:

- Total transferred data volume
- Total migration time
- Migration downtime

The total transferred data volume is defined to be the total amount of memory transferred during the entire migration process. Total migration time is the time taken starting from initiating the migration operation till the VM is resumed at the target host. Migration downtime is the duration of the stop-and-copy phase when the VM's execution is stopped.

To estimate the above factors, we define the key parameters that affect the migration cost factors to be:

- VM RAM size
- Available network bandwidth
- Page dirtying rate

The above parameters directly impact the factors contributing to the overhead due to migration. An effective parameter is the VM memory size. A vm with a larger RAM size is expected to take longer time to be migrated than a smaller one, given that they have same available network bandwidth. The available network bandwidth is inversely proportional to the total migration time and migration downtime. As the available network link speed increases, the faster a given vm can be transferred from source host to target host over the network. The number of rounds of a pre-copy algorithm varies according to rate at which the running workload attempts to modify the memory contents of the vm, defined with the page dirtying rate.

We propose an algorithm for estimating the migration overhead. This is important to be considered in a DVMC algorithm. Many of the literature researches attempt to minimize the migration overhead by choosing migration plans having the least number of migrations. This is not always a valid assumption since VMs are usually heterogeneous in a typical cloud environment. Thus, a consolidation plan having a fewer number of migrations doesn't necessarily cost less than another consolidation plan with a greater number of migrations. For example, a consolidation plan for three VMs having a small memory sizes may be less costly than another plan for two VMs with larger memory sizes.

Table 1: Migration Overhead Estimation - Notations Summary

Symbol	Description
V_{mem}	VM RAM size
V_{BW}	VM available Bandwidth
V_{cpu}	VM CPU demand
PDR	Page Dirtying Rate
$resuming\_thd$	Time to resume a VM on a destination host
$MinTDV\_thd$	Minimum threshold for transferred data volume
$MaxNumRounds$	Maximum number of pre-copy iterations
Est_MO	Total calculated Migration Overhead
MDV_PerR	List for Migrated Data Volume per round
$TMVD$	Total Migrated Data Volume
$RDuration$	List for Rounds' Duration
TMT	Total Migration Time
DT	Migration Downtime

Algorithm 1 Calculate VM Migration Cost

INPUTS: V_{mem}, V_{BW}, V_{cpu}, PDR, $resuming\_thd$, $MinTDV\_thd$, α_1, α_2, α_3, $MaxNumRounds$
OUTPUTS: Est_MO

1: $MDV\_PerR[0] = V_{mem}$
2: $TMVD = V_{mem}$
{Iterative copying starts}
3: **for** $i = 0, ..., MaxNumR$ **do**
4: $RDuration[i] = MDVperR[i]/V_{BW}$
5: $TMT+ = RDuration[i]$
6: $MDVperR[i+1] = RDuration[i] * PDR$
7: $TMDV+ = MDVperR[i+1]$
8: **if** $MDVperR[i+1] < MinTDV\_thd$ **then**
9: $RDuration[i+1] = MDVperR[i+1]/V_{BW}$
10: $TMT+ = RDuration[i+1]$
11: $DT = RDuration[i+1] + resuming\_thd$
12: **break**
13: **end if**
{Stop and Copy phase}
14: $RDuration[i+1] = MDVperR[i+1]/V_{BW}$
15: $TMT+ = RDuration[i+1]$
16: $DT = RDuration[i+1] + resuming\_thd$
17: **end for**
18: $Est\_MO = \alpha_1 * TMDV + \alpha_2 * TMT + \alpha_3 * DT$

In Algorithm 1, live migration overhead estimation approach is shown. A summary of the used symbols is shown in Table 1.

The overhead estimation algorithm receives the vm memory size, bandwidth and cpu demand as inputs. For real applications, the page dirtying rate of a given workload can be monitored and estimated using toolkits, as libvirt [19]. For simulation purposes, the page dirtying rate can be assumed to be a random percentage from the vm memory size. The expected outputs of the algorithm is a total estimated cost for the live migration operation. The pre-copy algorithm starts with migrating all the vm memory contents [line 1]. Then, for pre-defined number of rounds, the algorithm proceeds with re-copying the dirtied

pages. In each round, the round duration is calculated as the quotient of the dirtied memory volume in the previous round and the total available bandwidth [line 4]. The migrated data volume of the next round is estimated to be equal to the product of the current round duration and the page dirtying rate. The stopping condition of the algorithm is either reaching a maximum threshold for the number of rounds or when the amount of migrated memory volume falls under a given threshold. When the maximum number of rounds is reached, the stop-and-copy phase [line 9 - 11] and [line 14 - 16] proceeds with the last iteration of dirty pages copying and then estimates the migration downtime.

The total cost is calculated as the weighted sum of the migration cost factors. Weights α_1 , α_2 , α_3 represent each factor significance in the contribution to the total cost such that:

$$\alpha_1 + \alpha_2 + \alpha_3 = 1 \qquad (6)$$

4. Simulated Annealing-based DVMC Algorithm

Simulated annealing, inspired by annealing in metallurgy, is a metaheuristic used to provide near-optimal solutions for global optimization problems. In metallurgy, annealing involves heating metals and then controllably cooling them to reduce their defects. In the simulated algorithmic approach, the resemblance of slow cooling is represented by the decreasing probability of accepting bad solutions during the exploration of the search space. At the beginning, i.e. at high temperature, the search space is explored and worse solutions may be accepted to avoid being trapped in local optima. As the temperature approaches zero, a simulated annealing-based algorithm is less probable to accept worse solutions and will only take better transitions as in a downhill/uphill greedy approach. In this work, the initial VM placement is performed by the First Fit (FF) heuristic. Afterwards, the migration overhead-aware simulated annealing DVMC is used periodically in the background to dynamically optimize the VM-PM placement.

Algorithm 2 Simulated Annealing-based DVMC algorithm

INPUTS: $AllPMs, AllVMs, prob\_threshold$
OUTPUTS: MM_{best}

1: $MM_{old} = FF(AllPMs, AllVMs)$
2: $MM_{curr} = MM_{best} = MM_{old}$
3: $E_{best} = Energy(MM_{best})$
4: **while** $Current\_Temp > 0$ **do**
5: $\quad E_{curr} = Energy(MM_{curr})$
6: $\quad MM_{neigh} = GetNeighbor(MM_{curr})$
7: $\quad E_{neigh} = Energy(MM_{neigh})$
8: $\quad delta = E_{old} - E_{neigh}$
9: $\quad acc\_prob = e^{-delta/Current_{T}emp}$
10: $\quad$ **if** $acc\_prob > prob\_threshold$ **or** $delta > 0$ **then**
11: $\qquad MM_{curr} = MM_{neigh}$
12: $\qquad E_{curr} = E_{neigh}$
13: $\quad$ **end if**
14: $\quad$ **if** $E_{neigh} < E_{best}$ **then**
15: $\qquad MM_{best} = MM_{neigh}$
16: $\qquad E_{best} = E_{neigh}$
17: $\quad$ **end if**
18: $\quad$ Reduce temperature
19: **end while**

In algorithm 2, the proposed migration overhead-aware simulated annealing DVMC algorithm is shown. Initially, the set of VMs selected from underutilized hosts are placed on the set of available PMs. Available PMs exclude over-utilized hosts and hosts previously turned to a low-power state. This is achieved using a heuristic algorithm which is first fit algorithm in our case [Line 1]. Simulated annealing based approach starts in line 4 and continues till the temperature reaches 0. In each iteration, the objective function value for the current migration map, i.e. the energy of the solution, is calculated [line 5] as E_{curr}. Then, a neighboring migration map MM_{neigh} is obtained through *GetNeighbor* function. The energy of the new migration map is computed [Line 7] and compared with the current migration map. If the neighboring solution provides a better solution, i.e. less energy then the new solution is accepted. Otherwise, the probability of acceptance is computed using Metropolis Criterion [20] that varies according to the temperature. At high temperatures, i.e. early iterations, the acceptance probability evaluates to a large value, implying that it is likely to accept new solutions even if they are not good enough (exploration phase). At low temperatures, the acceptance probability evaluates to small values and it is more likely to reject bad solutions (exploitation phase). MM_{best} and E_{best} keep track of the migration map providing the least objective function value and its corresponding objective function value respectively [lines 15 - 16]. It is important to reduce temperature at slow rate to ensure convergence to global optimum. Temperature is reduced by 0.01 for every iteration in our experiments.

Neighbor Function Neighbor function is a method adding the sense of randomness to this simulated annealing-based approach. The function simply chooses a number of VMs from the given migration map at random and assigns them to other random PMs. The overloaded, switched off and PMs Underutilized hosts that are currently being consolidated are excluded from the list of possible destination PM candidates. The number of VMs to be shuffled is fixed and whose value is best determined through empirical experiments.

5. Experiments and Results

Conducting large-scale experiments on real environments repeatably is very expensive. Thereby, experiments are conducted by using Cloudsim [21]. Cloudsim is an open-source cloud infrastructure simulation environment written in Java. We extended cloudsim for the implementation and evaluation of the proposed algorithm. Workload data are obtained from real cloud traces from CoMon project, a monitoring infrastructure for PlanetLab [22]. A workload file of a given day contain data of CPU usage of thousands of VMs collected every five minutes on this day. In this section, we introduce migration overhead-aware simulated annealing based DVMC algorithm experiments and results. Xen is used as the hypervisor. Experiments are carried out offline on a Lenovo workstation (Intel Core i7 2.5 GHz CPU (4 cores), 16 GB of RAM and 2 TB storage). The number of created VMs is double the number of PMs. There are 3 VM categories according to their RAM size; Large-sized VMs (4 GB), Medium-sized VMs (2 GB) and Small-sized VMs (500 MB). All VMs having single-core 1000 MIPS CPU, 100 GB of storage and 1GBit/sec of network bandwidth. The power consumed by a host is computed based on a Linear model according to its CPU utilization given by equation 7.

$$P(util) = (P_{max} - P_{idle}) * util + P_{idle} \tag{7}$$

where P_{max} is the maximum power consumed by the host, P_{idle} is the power consumed by the host when in an idle state. P_{max} and P_{idle} are fixed during the experiments.

The experiment's objective is to assess the quality of the VM consolidation decisions for each of Best Fit (BF), First Fit (FF), Worst Fit (WF) and Simulated Annealing-based algorithms.

Table 2: Experiments Details

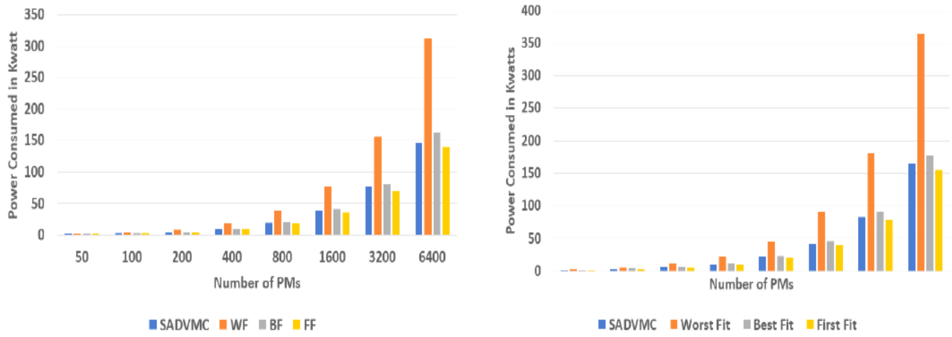
Experiment #	# PMs	#VMs	Workload
Exp 1	50	100	20110322
Exp 2	100	200	
Exp 3	200	400	
Exp 4	400	800	
Exp 5	800	1600	20110303
Exp 6	1600	3200	
Exp 7	3200	6400	
Exp 8	6400	12800	

The total data center power consumption during a full-day simulation is the metric proposed as an indicator for energy consumption. It is worth mentioning that the total power consumption of a data center includes the power consumed by servers, storage units, cooling and network equipment. For simplicity, we only focus on the power consumed by servers in this work. Eight experiments were conducted with varying the number of running PMs. Experiments are repeated twice running two different workload files of PlanetLab featuring two days. Details are shown in Table 2.

In figure 1 we show the total power consumption for March 3rd and 22nd from CoMon workloads for each of the compared algorithms with increasing the number of PMs. Worst Fit achieves the least power saving followed by Best Fit, Overhead-aware Simulated Annealing then First Fit.

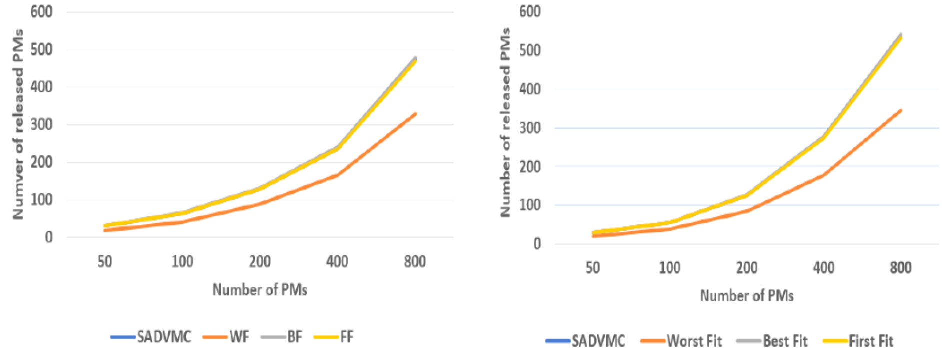
In figure 2, the number of released PMs are shown with increasing the number of PMs. Overhead-aware SADVMC, Best Fit and First Fit provide very close results and hence, the curve representing their number of released PMs are overlapping.

Results show that first fit consolidation approach outperforms best fit, worst fit and our proposed simulated annealing approach in terms of amount of power saved and number of released PMs. Being migration overhead-unaware, first fit performs consolidation regardless current VM-PM placement and hence saves more power and frees up more PMs. However, our proposed overhead-aware simulated annealing approach performs consolidation with consideration to the current VM placement and the imposed cost due to migration. This introduces higher computation time compared to the other algorithms. Computation time greatly rises with the increase of the cluster size being consolidated.



(a) (b)

Figure 1: Power consumption of PMs for different algorithms with increasing number of PMs for (a) 20110303 (b) 20110322



(a) (b)

Figure 2: Number of released PMs for different algorithms with increasing number of PMs for (a) 20110303 (b) 20110322

6. Conclusion and Future Work

In this research, we proposed an algorithm for dynamic virtual machine consolidation based on simulated annealing. The devised algorithm accounts for the overhead due to live migration based on a proposed estimation model. The algorithm has two objectives: minimizing the total data center energy consumption and minimizing the total migration overhead. We conducted experiments comparing our migration overhead-aware simulated annealing-based algorithm to several heuristic algorithms: best fit, worst fit and first fit. From the perspective of the total amount of saved power, SADVMC provided results close to first fit algorithm. However, the difference is first fit does not consider migration overhead and only performs bin-packing for VM instances onto available PMs. From the computation time point of view, the time elapsed to consolidate PMs rises exponentially with increasing PMs size and is around double that of First Fit.

In the future, we plan to integrate the migration cost estimation model with other DVMC algorithms and inspect the results. The migration cost estimation model can also be flexible to handle different cost aspects. In the future, we intend to consider the affinity of a VM to other related VMs before migration as part of the migration cost. Sometimes multiple VMs have dependencies and their physical proximity highly affect their performance. Moreover, it has been noticed that SLA violations may arise from aggressive consolidation. Hence, we intend to propose different consolidation operation modes providing a trade off between the amount of energy preserved and posed SLA-violations.

References

[1]. M. Telford, "The Case for Energy-Proportional Computing," no. March, pp. 36–43, 2004.

[2]. A. Strunk, "Costs of virtual machine live migration: A survey," Proceedings - 2012 IEEE 8th World Congress on Services, SERVICES 2012, pp. 323–329, 2012.

[3]. S. Kirkpatrick, "Optimization by Simulated Annealing Optimization by Simulated Annealing," no. January 1983, 1983.

[4]. E. Feller, L. Rilling, and C. Morin, "Energy-Aware Ant Colony Based Workload Placement in Clouds," 2011 IEEE/ACM 12th International Conference on Grid Computing, 2011. [Online]. Available: http://ieeexplore.ieee.org/document/6076495/

[5]. Q. Wu, F. Ishikawa, Q. Zhu, and Y. Xia, "Energy and migration cost-aware dynamic virtual machine consolidation in heterogeneous cloud datacenters," IEEE Transactions on Services Computing, vol. 1374, no. c, pp. 1–1, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7589069/

[6]. N. Ajith Singh and M. Hemalatha, "Cluster based bee algorithm for virtual machine placement in cloud data centre," Journal of Theoretical and Applied Information Technology, vol. 57, no. 3, pp. 1–10, 2013.

[7]. G. Xu, Y. Ding, J. Zhao, L. Hu, and X. Fu, "A Novel Artificial Bee Colony Approach of Live Virtual Machine Migration Policy Using Bayes Theorem," Scientific World Journal, vol. 2013, 2013.

[8]. N. J. Kansal and I. Chana, "Energy-aware Virtual Machine Migration for Cloud Computing - A Firefly Optimization Approach," Journal of Grid Computing, vol. 14, no. 2, pp. 327–345, 2016. [Online]. Available: http://dx.doi.org/10.1007/s10723-016-9364-0

[9]. F. Farahnakian, A. Ashraf, T. Pahikkala, P. Liljeberg, J. Plosila, I. Porres, and H. Tenhunen, "Using Ant Colony System to Consolidate VMs for Green Cloud Computing," IEEE Transactions on Services Computing, vol. 8, no. 2, pp. 187–198, 2015.

[10]. L. Adhianto, S. Banerjee, M. Fagan, M. Krentel, G. Marin, J. Mellor-Crummey, and N. R. Tallent, "Multi-objective, Decentralized Dynamic Virtual Machine Consolidation using ACO Metaheuristic in Computing Clouds," Concurrency Computation Practice and Experience, vol. 22, no. 6, pp. 685–701, 2017.

[11]. M. Marzolla, O. Babaoglu, and F. Panzieri, "Server consolidation in clouds through gossiping," 2011 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, WoWMoM 2011 - Digital Proceedings, 2011.

[12]. S. Wang, A. Zhou, C. H. Hsu, X. Xiao, and F. Yang, "Provision of Data-Intensive Services Through Energy-and QoS-Aware Virtual Machine Placement in National Cloud Data Centers," IEEE Transactions on Emerging Topics in Computing, vol. 4, no. 2, pp. 290–300, 2016.

[13]. K. Rybina, W. Dargie, A. Strunk, and A. Schill, "Investigation into the energy cost of live migration of virtual machines," The Third IFIP Conference on Sustainable Internet and ICT for Sustainability (SustainIT 2013), pp. 1–8, 2013.

[14]. A. Strunk, "A lightweight model for estimating energy cost of live migration of virtual machines," IEEE International Conference on Cloud Computing, CLOUD, pp. 510–517, 2013.

[15]. K. Rybina and A. Schill, "Estimating Energy Consumption during Live Migration of Virtual Machines," pp. 1–5, 2016.

[16]. A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-aware resource allocation heuristics for efficient management of data centers for Cloud computing," Future Generation Computer Systems, vol. 28, no. 5, pp. 755–768, 2012. [Online]. Available: http://dx.doi.org/10.1016/j.future.2011.04.017

[17]. D. Deng, K. He, and Y. Chen, "DYNAMIC VIRTUAL MACHINE CONSOLIDATION FOR IMPROVING ENERGY EFFICIENCY IN CLOUD DATA CENTERS," 2016.

[18]. A. Beloglazov and R. Buyya, "Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers," pp. 1397–1420, 2012.

[19]. (2016) Libvirt toolkit. [Online]. Available: http://libvirt.org/

[20]. N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," The Journal of Chemical Physics, vol. 21, no. 6, pp. 1087–1092, 1953.

[21]. R. N. Calheiros, R. Ranjan, A. Beloglazov, C. De Rose, and R. Buyya, "Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," vol. 41, pp. 23–50, 01 2011.

[22]. K. Park and V. S. Pai, "CoMon : A Mostly-Scalable Monitoring System for PlanetLab," no. August, 2004.