An Energy-Aware and Under-SLA-Constraints VM Consolidation Strategy Based on the Optimal Matching Method

WeiLing Li, School of Software Theory and Technology Chongqing Key Lab, Chongqing University, Chongqing, China
Yongbo Wang, School of Software Theory and Technology Chongqing Key Lab, Chongqing University, Chongqing, China
Yuandou Wang, School of Software Theory and Technology Chongqing Key Lab, Chongqing University, Chongqing, China
YunNi Xia, School of Software Theory and Technology Chongqing Key Lab, Chongqing University, Chongqing, China
Xin Luo, Chinese Academy of Sciences, Chongqing Institute of Green and Intelligent Technology, Chongqing, China
Quanwang Wu, School of Software Theory and Technology Chongqing Key Lab, Chongqing University, Chongqing, China

ABSTRACT

Growing demand of computational power brings increasing scale and complexity of cloud datacenters. However, such increase also generates growing energy consumption and related cost incurred for cooling and maintenance. With concerns of cost and energy saving by both industry and academy, the reduction of energy consumption of cloud datacenters becomes a hotspot issue. Recently, virtual-machine-consolidation-based strategies are proposed as promising methods for reduction of cloud energy consumption. Virtual machine (VM) consolidation effectively increases the resource utilization rate. However, it remains a great challenge how to reduce energy consumption while maintaining the quality of service (QoS) at a satisfactory level. In this work, a comprehensive framework is presented for the above-mentioned problem, which aims at maximizing the number of physical machines (PMs) to be turned off within a consolidation period following the constraints of QoS, in terms of Service-Level-Agreement (SLA) violation rate. In comparison with most existing related works which consider invariant utilization rate of PMs in computing energy reduction of candidate migration plans, propose framework considers time-varying utilization rate and employs the number of PMs to be turned off within a consolidation period (NPTCP for simple) as the optimization objective. The proposed framework consists of a resource selection algorithm taking the predicted migration overhead (derived by the Pareto distribution) as inputs and another algorithm generating optimal matching plans based on preference scores of candidate VMs. For the model validation purpose, a case study is conducted on the CloudSim simulation platform and it shows that the proposed method achieves better energy reduction and less SLA violation.

KEYWORDS

Cloud, CloudSim Toolkit, Datacenter, Virtual Machine Consolidation

INTRODUCTION

Cloud computing, as an emerging technology, is featured by the ability of elastic provisioning of on-demand computing resources ranging from applications to storage over the Internet on a pay-per-use manner (Graubner et al. 2013). Cloud computing brings in numerous benefits for companies and end customers, e.g., end customers can invoke computational resources for almost all possible types of

DOI: 10.4018/IJWSR.2017100104

workload when resources are reachable and a large number of today's Web services are deployed on the Cloud (Deng et al.2014), and published on the Internet (Gan et al. 2017).

Cloud data-centers are key enablers for the scalability of the cloud platform (Xia et al., 2015). However, inefficient resource utilization is a common problem in today's cloud datacenters (Vogels, 2008) mainly because datacenters are usually designed and deployed to meet the peak work load but actually running at low load at most of the operational time. Datacenter owners thus have to afford huge cost for the investment into physical resources, which are often under-utilized. Both industry and academy are in a high need of methods to optimize energy efficiency of cloud datacenters. Recently, great efforts have been taken in this direction. The underlying requirement for energy-efficiency-optimization is to minimize energy consumption while meeting the performance promised to cloud users specified by the Service-Level-Agreement (SLA). Several techniques and methods, at both hardware and software level, are developed for the above-mentioned purpose. At hardware level, e.g., Dynamic Component Deactivation (DCD) and Dynamic Performance Scaling (DPS) are two such techniques, by which hardware components can enter the dormant state under the control of hardware timers. While at software level, the live migration techniques are developed to offload tasks and processes among different machines for load balancing, VM-consolidation, or risk-decentralization purposes.

Virtual machine consolidation moves virtual machines (VM) form under-utilized physical machines (PM) to better-utilized ones. Thus, PMs with non-load can be switched into dormant state and consumes less energy. Moreover, the remaining PMs alive can consequently take a more condensed workload and thus the cloud system achieves a high utilization rate. Unfortunately, VM consolidation itself consumes time, energy, and resources, especially when the destination PM is highly loaded. Performance and QoS degradation, which may further cause SLA violation, can often be observed when frequent migration activities are carried out. Hence, strategies must be designed in such a way that energy reduction and SLA commitment are both taken into account.

Note that various early contributions consider consolidation and migration plans to be independently carried out on PMs, i.e., every PM decides whether to migrate or receive VMs based on its own utilization rather than system utilization. Recently, some other works consider dependent consolidation plans but most of them assume invariant utilization rate of PMs in computing energy reduction of candidate migration plans, i.e., the gains of energy efficiency is calculated by comparing VM distributions before and after the consolidation plan is carried out but the utilization rate of every PM is assumed to be unchanged after the consolidation is accomplished. Such assumption may lead to considerable accuracy loss in calculating energy reduction of candidate consolidation plans and could further mislead the determination of optimal plan. In contrast, however, this work considers time-varying utilization rate of PMs and circumvent such accuracy loss by employing the number of PMs to be turned off within a consolidation period as the optimization objective instead of energy reduction.

In this study, a novel model is proposed to describe the performance-aware and SLA-constraint VM consolidation problem and develop a selection algorithm taking the predicted migration overhead (derived by the Pareto distribution) as inputs and another algorithm to generate the optimal matching plans based on preference scores of candidate VMs. For the model validation purpose, this study conducts a case study on the CloudSim simulation platform and show that our proposed algorithms achieve better energy saving efficiency.

RELATED WORKS

VM consolidation is a promising method using live migration to consolidate VMs periodically (Farahnakian et al., 2014) to reduce the power consumption by packing the existing VMs into fewer PMs and switching the idle PMs into a power saving mode (Farahnakian et al., 2016). In recent years, there have been numerous researches in data center energy efficiency and most of them focus on using VM consolidation methods.

Some studies attempted to use statistical or historical data to predict the system state and decide how to implement the VM consolidation. Bobroff et al. (2007) present a dynamic server migration and consolidation algorithm, which uses time series forecasting techniques and bin packing heuristic to minimize the number of PMs required to support a workload. However, their algorithm does not take into account the number of migrations required to a new placement.

Sandpiper (Wood et al., 2009) is designed to detect over-utilized PMs and creates a new mapping scheme of physical resources to virtual resources. To detect over- loaded PMs, Sandpiper collects VM and PM usage statistics, constructs profiles of resource usage and then uses the prediction techniques to achieve its goal. Xiao et al. (2013) presented a system, which use virtualization technology to allocate data center resources dynamically based on application demands and support green computing by optimizing the number of servers in use. They introduce a concept called SKEWNESS to measure the unevenness in the multidimensional resource utilization of a server. By minimizing SKEWNESS, they can combine different types of workloads nicely and improve the overall utilization of server resources.

Virtual Power Management (VPM), proposed by Nathuji et al. (2007), is designed to support the isolated and independent operation assumed by guest VMs running on virtualized platforms and to make it possible to control and globally coordinate the effects of the diverse power management policies applied by these VMs to virtualized resources. However, it did not take migration cost into account.

Srikantaiah et al. (2008) have applied bin-packing model to virtual machines consolidation based on CPU and disk usage. Experiments show the model can well balance energy consumption and performance in virtual environment. But the approach is application-oriented which is not suitable for a general virtual environment. Kusic et al. (2009) implemented a Limited Look-ahead Control (LLC) framework for dynamic resource provisioning and power management in virtual environments. The proposed approach accounts for the switching costs incurred while provisioning virtual machines. However, the proposed model requires simulation-based learning for the application-specific adjustments, which cannot be implemented by Infrastructure as a Service cloud providers. Moreover, it is not suitable for large-scale system for its complexity.

A. Verma et al. (2008) have presented an application placement controller pMapper, which that minimizes power and migration costs, while meeting the performance guarantees. The controller addresses the problem of power and migration cost aware application placement in heterogeneous server clusters. It can achieve minimizing power with a fixed performance requirement. A simple static and semi-static virtual machine consolidation approaches have been implemented in an enterprise environment in their recent work (Verma et al., 2009).

Wu & Fuyuki (2015) proposed an improved grouping genetic algorithm to estimate migration overhead and implement heterogeneous VM consolidation. The migration overhead is calculated according to memory size and page-dirty-ratio of the surveyed VM. A scoring system deployed the control node, based on migration overhead and the energy reduction, and is further developed to decide the optimal consolidation plan from all candidate ones. Wang et al. (2017) discussed the underlying problem of deciding necessary and sufficient Condition for effective resource mapping.

Beloglazov et al. (2011, 2013) proposed heuristic-based and statistical-measure-based approaches to dynamically decide threshold values to trigger consolidation activities. The VM Consolidation problem is divided into sub-problems. Each sub-problem, e.g., the overload detection problem and VM selection problem, has multiple candidates of heuristic algorithms, which are selected at run-time based on system status.

Monilet al. (2016) proposed a heuristic and fuzzy-logic-based migration control method based on Beloglazov's strategies. In their work, a Fuzzy VM selection algorithm dynamically decides the source PM for consolidation. The overhead to make such decision is reduced in comparison with traditional approaches with deterministic decisions. An overload detection algorithm is also proposed to decide the destination PM for the consolidation activity, based on the mean, median and standard deviation of utilization of candidate PMs.

PROPOSED METHOD

In this section, a resource selection algorithm and an optimal-matching-based VM consolidation algorithm are proposed for performance-aware and SLA-constrained VM consolidation problem. Before going in details, the datacenter configuration is discussed in part A. This part elaborates basic configuration of a datacenter and discusses the need of using *NPTCP* as the alternative optimization objective.

Datacenter Configuration

This study considers a cloud datacenter comprised of numbers of physical machines. Each physical machine, PM_i, can be characterized by its CPU capacity (CPU_i), memory size (MEM_i), and bandwidth (BW_i). Through virtualization, a single PM can support and execute no more than *n* multiple VMs. Each VM instances can be characterized by their assigned resources (cpu_j', mem_j' and bw_j') or their actual resource usage (cpu_j, mem_j and bw_j). The control node of the datacenter is response for the VM-PM mapping and energy-aware VM consolidation.

CPU utilization is the most important performance and energy-efficiency metric since CPUs consume the majority of energy. (1) captures the relationship between CPU utilization of a running PM, denoted by u_i, and its power consumption (i.e., energy consumption per unit time), $P(u_i)$. The maximal/idle power consumptions of a PM are usually considered to fixed values, as denoted by P_{peak} *and* P_{idle}. According to most existing works, $P(u_i)$ is approximately showing a linearly trend and can be obtained as (Beloglazov & Buyya, 2013):

$$P\left(u_i\right)=\begin{cases} P_{idle,i}+\left(P_{peak,i}-P_{idle,i}\right)\times u_i, & if\ u_i>0 \\ 0, & if\ u_i=0 \end{cases} \tag{1}$$

Consequently, energy consumption can be obtained as the following equation by considering time-varying power consumption:

$$E\left(u_i\left(t\right)\right)=\int_0^T P\left(u_i\left(t\right)\right)dt \tag{2}$$

CPU utilization of a PM can be calculated as the CPU usage for VMs plus Virtual-Machine-Manager (VMM):

$$\overline{u_i}\left(t\right)=\frac{\overline{cpu_{vmm}}+\overline{cpu_{i1}\left(t\right)}+\ldots+\overline{cpu_{in}\left(t\right)}}{cpu_i} \tag{3}$$

Let E_m denote the total energy consumption of m PMs in a period of time *T* and $P_m\left(x\right)$ the total energy consumption on condition that *x* PMs are already turned into dormant state at time *T*:

$$E_m=\sum_{i=1}^{m}E\left(u_i\left(t\right)\right) \tag{4}$$

$$E_m\left(x\right)=\left(m-x\right)P_{idle}T+\left(P_{peak}-P_{idle}\right)*\left(\sum_{i=1}^{m}\int_{0}^{T}\overline{u_i}dt-x*\frac{\overline{cpu_{vmm}}}{cpu_i}*T\right) \tag{5}$$

Let's consider two different consolidation plans which turn *e* and *s* PMs into the dormant state and 1<*e*<*s*<*m*. *ES(e)* and *ES(s)* denote the energy savings for the two consolidation plans, it is clear to see that:

$$ES\left(s\right)-ES\left(e\right)=\left(E_m-E_m\left(s\right)\right)-\left(E_m-E_m\left(e\right)\right)$$

$$=\left(s-e\right)P_{idle}T+\left(P_{peak}-P_{idle}\right)*\left(s-e\right)*\frac{\overline{cpu_{vmm}}}{cpu_i}*T>0 \tag{6}$$

Equation (6) indicates that during a VM consolidation period *T*, turning more PMs into dormant state leads to a better energy reduction in next period.

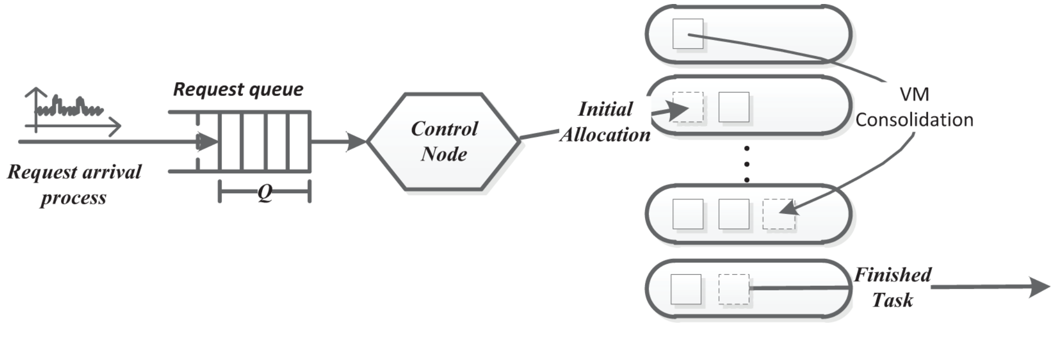
System Model

As mentioned by the previous section, we need to maximize the number of PMs to be turned off. The decision of turning off PMs and VM consolidation is made by the control node. To achieve a global optimality, the control note periodically or randomly scans the datacenter (or a part of the datacenter) to gather system information and make such decision. The control node maintains a request buffer for consecutively-arrived requests. It continuously handles requests from the request buffer on an FIFO (first-in-first-output) basis and tries to place corresponding VMs on PMs. However, the initial placement decision made by the control node may be biased and not energy-efficient. E.g., the Openstack framework can randomly select one PM from top-10 under-utilized candidates for the load-balance purpose. In doing this, PMs tend to equally share the workload and have a low utilization, which is disadvantageous to energy-efficiency. Therefore, consolidation activities may alter the initial placement scheme by moving, through the live migration techniques, VMs to towards hotspots while turning off cold one.

Based on the above discussions, an abstract control-flow model for the consolidation decision problem can be developed in Figure 1. It abstracts away implementation details cloud platforms and components while preserving the control-flow contents useful for optimal consolidation plan decision.

By repeatedly copying the refreshed dirty memory pages, live VM migration allows VMs being migrated continuing working, thereby reducing the service downtime. Note that live migration is not

Figure 1. An overview diagram of control node functions



always welcomed since it requires more pages to be copied and moved to the destination PM than the non-live one. Consequently, increasing SLA-violations, caused by insufficient available bandwidth needed to carry out page copy on the destination PM, may occur. Moreover, local computation is required on the destination PMs and thus a migration can only be carried out if the destination PM has sufficient remaining CPU resources for such computation.

It is easy to see that the algorithm for destination PM selection capable of selecting the appropriate destination PM, with adequate remaining bandwidth and CPU resources, is in high need. For this purpose, we first have to calculate the lower bound the time, *T*, required to carry out a successful migration:

$$T = \frac{E(D)}{Bi} \tag{7}$$

where *E(D)* denotes the expected amount of data to be copied within a migration, *D* is usually decided by the ratio at which memory pages of the VM are dirtied, and *Bi* the remaining bandwidth on the i^{th} candidate destination machine *PMi.*

It has been widely recognized that heavy-tailed distributions are well suited for modeling job processing and request handling activities in computer systems and networks. Thus, *D* is considered to be following such distribution. In particular, *D* is with the Pareto distribution type and it is lower-bounded by $\hat{x}$. The tail of *D* can therefore be obtained as:

$$\mathbb{P}\left\{X \geq \hat{x}\right\} = \left[\frac{x}{\hat{x}}\right]^{-v_j}, \hat{x} = mem_j \tag{8}$$

v_j denotes the Pareto index and α and ω are constants associated with the special resource configuration of VM memory:

$$v_j = \alpha - \omega \times dratio_j, \alpha > \omega > 1, \alpha - \omega > 2 \tag{9}$$

Total expected pages to be migrated can be calculated as:

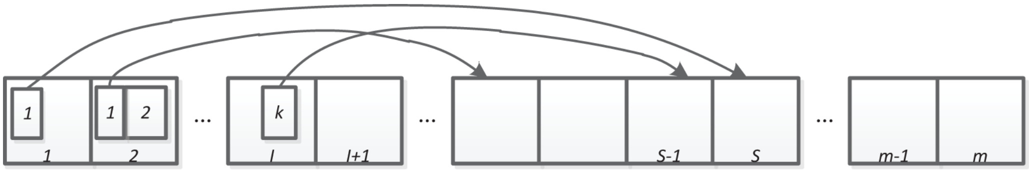
$$\mathbb{E}\left(D_j\right) = \frac{v_j * \hat{x}}{v_i - 1} \tag{10}$$

Deciding the Maximum *NPTCP*

Based on observations above, it can be seen that the consolidation plan are supposed to follow the constraints of bandwidth and CPU resources. For this purpose, we first have to decide the maximum number of PMs to be shut down in a period *T* (see Figure 2).

Let x_{ij} denote that VM_i is on PM_j, *m* the total number of current alive PMs, *NPTCP* the number of PMs which can be turned off within a period. PMs are sorted by their assigned resources, i.e., PM_i has more rest resources than PM_{i+1}. The problem of deciding the maximum number of PMs to be shut down is formulated as follows:

Figure 2. Migration activities on PMs



$$Max.NPTCP \tag{11}$$

$$s.t.\ \sum_{i=1}^{NPTCP}\sum_{j=1}^{m*n} x_{ij} \leq m - NPTCP \tag{12}$$

$$\frac{\sum_{i=1}^{NPTCP}\sum_{j=1}^{m*n} x_{ij} \times \mathbb{E}(j)}{\sum_{i=NPTCP+1}^{S}\left(B_i - \sum_{j=1}^{m*n} x_{ij} \times b_j\right)} \leq T \tag{13}$$

$$\sum_{j=1}^{m*n} x_{ij} \leq dn, i = S, 0 < d \leq 1 \tag{14}$$

$$\sum_{i=1}^{m} x_{ij} = 1, \forall j \tag{15}$$

The first three constraints guarantee that there is sufficient resource to support the migration activity within time *T*. The last equation guarantees that a VM can reside in one and only one PM.

Maximum *NPTCP* can be decided using Algorithm 1. Note that the complexity of this algorithm is *O*(*mlogm*) since the sorting time is *O(mlogm)* and it takes *O(m)* time to decide s when the sorting is done.

Algorithm 1. MAX NPTCP evaluation

```
Input Memory, CPU and Bandwidth configuration of each PM; Input assigned Memory, CPU and Bandwidth of each VM; Input Memory, CPU, Bandwidth usage and dirty ratio of each VM;
Input Period time T;
PMlist←Sort PMs increasingly based on PM assigned resource;
Integer VMN←0, NPTCP←0, td←0, tfbw←0;//VMN:a sum of VM on counted PMs; td: total data needs to be transmit; tfbw: total free bandwidth on destination PM;
While NPTCP ≤ 1/2m && VMN ≤ m-NPTCP&&# of VMs in PMs(NPTCP+VMN+1) < n && td/tfbw < T
    NPTCP←NPTCP+1;
    VMN←VMN +# of VM on PMlist (NPTCP);
    td←td+ totaldata(PMlist(NPTCP));
    tfbw←tfbw+ freebandwidth(PMlist (NPTCP+1:NPTCP+1+VMN));
return NPTCP;
```

Deciding Destination PMs

The candidate virtual machines ready for migration is denoted by *V* while the destination PMs selected by Algorithm 1can be described as *H*. The sizes of *V* and *H* are denoted by *m* and *n*. The problem of finding a strategy to migrate every candidate VM with minimum migration cost can be formulated as a matching problem on a weighted bipartite graph, $\{G=(V,H,E), w\}$.Networking plays a crucial role in Cloud computing (J. Huang et al. 2015&2016). VM consolidation costs networking resource and does negative effects to the performance of data centers. To take these effects into account, in this paper, we use the weight $w(j,i)$ of edge $e_{ji} \in E$denotes the cost of migrating VM_j to PM_i:

$$w(j,i) = \frac{\mathbb{E}(D_j) * MTD_{ji}}{B_i - \sum_{r=1}^{m} x_{ir} \times b_r} \tag{16}$$

MTD denotes the time-delay matrix and MTD_{ji} the delay required by migrating VM_j to PM_i. Such delay time depends on the datacenter network topology and relative position of both source PM (i.e., the PM which VM_j was deployed before migration) and the destination PM, i.e., PM_i. *MTD* can be measured and obtained from real tests.

A matching scheme *M* is assumed to be a subset of $V \times H$ such that $(v_j, h_i) \in M$ implies that *v* finds *h* acceptable, each VM is assigned to at most one PM and vice versa. $M(e)$ denotes a function over edges such that $M(e_{ij}) = 1$ if e_{ij} is accepted by the matching scheme M and $M(e_{ij}) = 0$ otherwise.

The resulting problem can be formulated as:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{m} w(j,i) * M(e_{ij}) \tag{17}$$

$$s.t. \sum_{j=1}^{m} M(e_{ij}) = 1, \forall i \tag{18}$$

$$\sum_{i=1}^{n} M(e_{ij}) = 1, \forall j \tag{19}$$

$$M(e_{ij}) \geq 0, \forall i, \forall j \tag{20}$$

To solve the problem, a preference list matrix, *PLM*, is generated. *PLM* is an m*n matrix and every row of the matrix represents the preference list of one of the candidate VMs. Elements of each row is a permutation of the set *H* with an increasing order by their weights. Let Tuples $p = (r, k)$ denote positions of PLM generated by the above algorithm, *r* is some row and *k* is some natural number. A matching corresponds to a set of positions, *P*, if there is exactly one position for each row and no two positions contain the same PM and *P=M.*

PLM can be generated by Algorithm 2 with the time complexity of n^2logn.

PLM serves as an input into Algorithm 3. This algorithm first explores the PLM column by column and considers each un-matched VM in turn. If there exists a non-overriding PM in this position, the algorithm generates a matching and removes the VM and PM from *V* and *H*, respectively.

Algorithm 2. PLMGENERATOR

```
input V: a set of candidate VM, H: a set of destination PM and w
input preference list matrix $PLM \leftarrow null$;
for j in V
    for i in H
        $T(key, value) \leftarrow \{i, w(j,i)\}$;
    endfor
    sort T increasingly based on value ;
    $PLM(j) \leftarrow T$;
    $T \leftarrow null$;
endfor
return PLM
```

Algorithm 3. POM (Pareto-Optimal-Matching) swap

```
input preference list matrix PLM, set V, set H
input list $L \leftarrow null$, set $VE \leftarrow \varnothing$,set $HE \leftarrow \varnothing$;
for $i \leftarrow 1$ to n
    for j in V
        if $PLM(i, j) \in H$
            $L(j) \leftarrow PLM(i, j)$;
            $VE := VE \cup j$;
            $HE := HE \cup PLM(i, j)$;
    endfor
    $H := H/HE$;
    $V := V/VE$;
endfor
for $k \leftarrow 1$ to s
    randomly pick x, y from V;
    $S \leftarrow w(x, L(x))+w(y, L(y))$;
    $S'$ $w(y, L(x))+w(x, L(y))$;
    if $S' < S$
        exchange(L(x),L(y))
endfor
return L
```

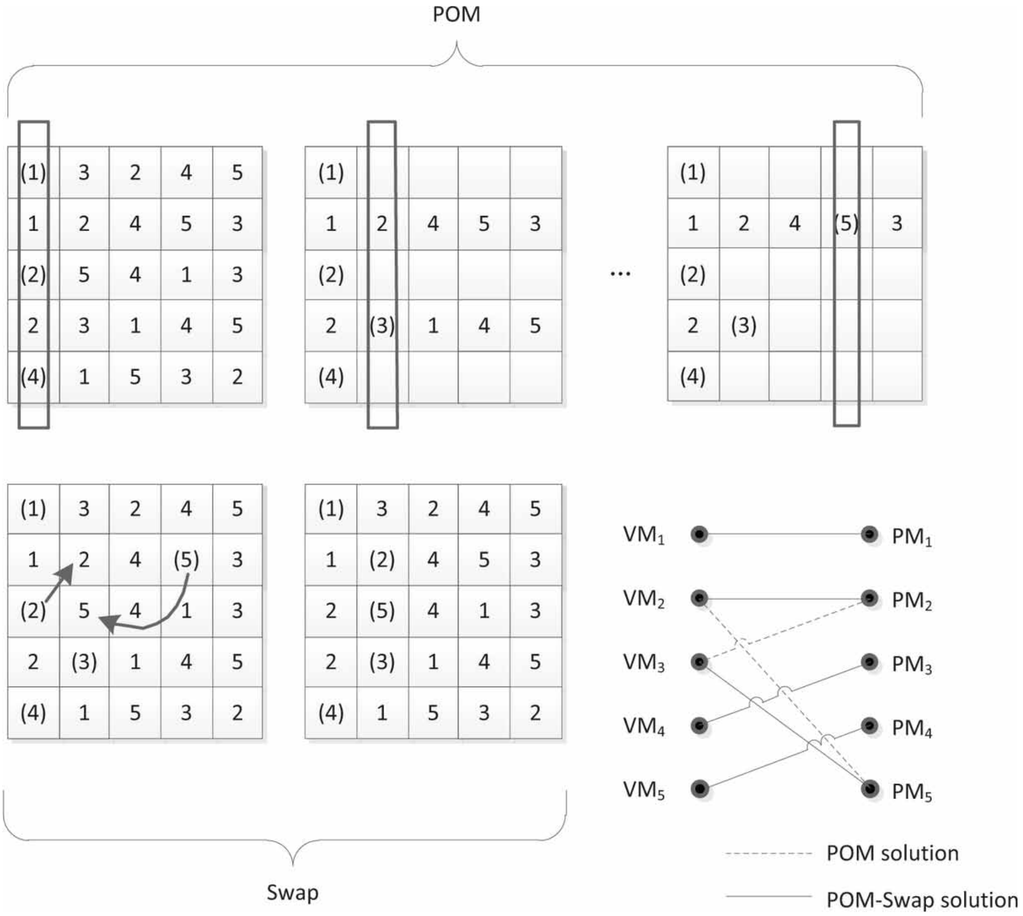
This operation makes sure that most selected elements are on the left of the *PLM* and it generated a solution called Pareto optimal matching (POM) which means no two VMs could both improve by swapping the PMS that they are assigned to.

The second loop of this algorithm randomly selects two matches specified by PLM and swaps the destination PMs on condition that such swap leads to a reduced aggregated weight of the corresponding bipartite graph. Parameter *s* denotes the cycle index of this loop.

A good example is presented in Figure 3 to show the effectiveness of the proposed algorithm. Each row of the *PLM* example given in this figure denotes a candidate VM of *V*, and elements of each row are a permutation of the destination PM set *H*. The algorithm first investigates the first column of PLM and decides that elements*1, 2* and *4* are available for VM *1, 3* and *5*, respectively. Matches of *{1, 1}*, *{3, 2}* and *{5, 4}* are generated and therefore selected VMs and PMs are removed from *V* and *H*. In the following, the algorithm traverses remaining columns. Thus, by the process above, matches of *{1, 1}*, *{2, 5}*, *{3, 2}*, *{4, 3}* and *{5, 4}* are finally decided.

Swap process randomly picks two matches out, *e.g.*, *{2, 5}* and *{3, 2}*. It then exchanges the PMs to generate two new matches, *i.e.*, *{2, 2}* and *{3, 5}*. Finally, it accepts the swap if $w(2,2) + w(3,5) < w(2, 5)+w(3, 2)$. The process repeats Algorithm 3 for *s* times. As shown by Figure 3, the consolidation plan illustrated in the lower right part is generated.

Figure 3. An example of POM-swap



EVALUATION

Experimental Setup

Our proposed algorithms are implemented in CloudSim 3.0.3 and the cloud system simulated is based on the HP ProLiant G4 (1860MIPS for each core) and HP ProLiant G5 (2660MIPS for each core) servers. Network bandwidth is 1GB/s. VMs are of 4 types, namely, High-CPU Medium Instance (2500 MIPS, 0.85 GB); Extra Large Instance (2000 MIPS, 3.75 GB); Small Instance (1000 MIPS, 1.7 GB); and Micro Instance (500 MIPS,613 MB). The consolidation period *T*is 2 seconds and the parameters of pare to index α and ω are 4 and 1.5, respectively. The consolidation period *T* is 2 seconds and the number of Swap loops is equal to *m*.

Energy consumption and SLA violation rate are considered as the major metrics for optimal consolidation determination:

1. **Energy Consumption (kWh):** The main metric as the target of VM consolidation is to reduce energy consumption. Energy consumption is computed by taking into account all PMs throughout the simulation by mapping of CPU and energy consumption from Table 1. The CPU utilization is measured and power consumption is calculated from Table 1 and at the end of the simulation energy consumption is measured by accumulating all PMs' energy consumption.

Table 1. Characteristics of servers

Machine Type	Power Consumption Based on CPU Utilization					
	0	20%	40%	60%	80%	100%
HP G4 (Watt)	86	92.6	99	106	112	117
HP G5 (Watt)	93.7	101	110	121	129	135

To compare the energy saving efficiency of datacenters of different sizes, ESE is introduced. ESE donates the energy consumption of a datacenter employed a consolidation strategy versus the energy consumption of a same size datacenter employed the basis energy saving method DVFS:

$$ESE\left(POM - Swap\right) = \frac{E_{POM-Swap}}{E_{DVFS}} \tag{21}$$

2. **Service level agreement violation (SLAV):** Combined impact of Overload time fraction and Performance degradation due to migration. This metric can be directly got from the Cloudsim.

Note that saving energy and reducing SLA violation rate is usually conflicting goals. The product of *ESE* and *SLAV*, *ESV*, is therefore considered as a metric to decide the optimality of consolidation plans.

Experimental Result

To study the effect of VM consolidation on energy consumption, the simulation runs using Random workloads for 24 hours. Proposed method is designed to compare with traditional consolidation strategies used in CloudSim. Traditional consolidation strategies are combinations of two Overload detection algorithms (LR and THR) and four VM selection (MC, MMT, MU and RS) methods. Consequently, proposed method is compared with 8 different consolidation configurations. As can be seen from tables and figures in this article, proposed method outperforms most other consolidation strategies in terms of energy efficiency and SLA violation rate. By considering ESV as the metric of optimality, proposed method outperforms all other consolidation strategies.

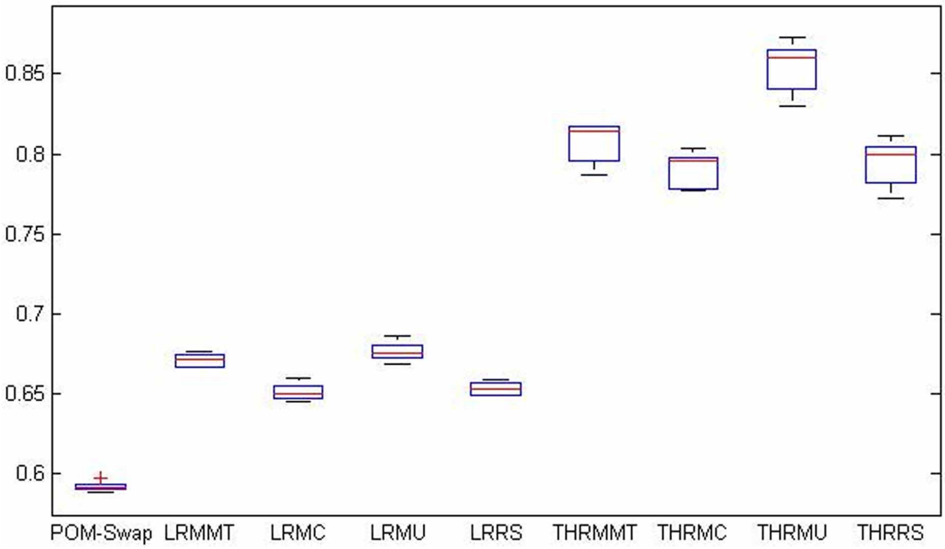
Energy consumptions of datacenters of five different sizes are shown in Table 2.

By comparing the ESE results of proposed and traditional methods in the Figure 4, it is obviously to see that energy consumption is significantly reduced in proposed method. Average ESE of the proposed method is 0.59 while the minimum average result of all other methods is 0.65. These results confirm that resource

Table 2. Energy consumption of different energy-saving strategies(kWh)

Number of Hosts	Energy Consumption of Different Energy-Saving Strategies(kWh)									
	DVFS	POM-Swap	Lr-mmt	Lr-mc	Lr-mu	Lr-rs	Thr-mmt	Thr-mc	Thr-mu	Thr-rs
25	27.86	16.49	18.79	18.23	18.82	18.29	21.92	21.66	23.01	21.5
50	53.11	31.26	35.4	34.5	35.55	34.52	42.45	41.41	44.81	41.76
75	76.3	45.14	51.64	50.38	52.34	50.29	62.37	61.34	66.62	61.88
100	101.56	60.01	68.21	65.84	68.93	66.35	82.97	80.8	87.67	81.44
150	152.63	91.33	101.73	98.52	102.84	99	124.24	121.42	131.26	122.05

Figure 4. Energy saving efficiency



selection strategy of proposed method, maximum *NPTCP*, performs well on saving energy. Therefore, it can be inferred that the energy saving strategy of proposed method owns an advantage over traditional methods.

The overall SLA violations of all consolidation methods are shown in Table 3 and Figure 5. The average SLA violation of our proposed method, 2.76%, outperforms Lr-mmt, Lr-mc, Lr-mu, Lr-rs, Thr-mmt, Thr-mc, and Thr-rs methods. Although our approach achieves similar violation to Thr-mu method, Thr-mu allows merely a single VM to be migrated at a time and while our approach allows multiple VMs to be migrated and definitely achieves better energy efficiency. As indicated by Figure 6, our proposed approach achieves lowest *ESV* as well.

CONCLUSION

VM consolidation is a promising method to reduce the energy consumption of the data centers. In this study, a novel consolidation strategy consists of a resource selection method and resource matching method is proposed. A Pareto distribution is employed for estimating the total memory required in migrating VMs and the overall optimality of consolidation can be measured by the maximum number of servers to be turned off in a given period. An algorithm called POM-Swap is presented in order to

Table 3. Overall SLA violation of different energy-saving strategies

Number of Hosts	POM-Swap	Lr-mmt	Lr-mc	Lr-mu	Lr-rs	Thr-mmt	Thr-mc	Thr-mu	Thr-rs
25	2.26%	2.56%	2.97%	3.18%	2.60%	2.92%	3.19%	2.67%	3.47%
50	2.67%	3.50%	3.14%	3.55%	3.55%	3.15%	3.34%	2.80%	3.16%
75	2.83%	4.23%	3.65%	3.74%	3.85%	3.07%	3.22%	2.61%	3.31%
100	2.86%	4.16%	4.04%	4.00%	3.75%	2.90%	3.29%	2.54%	3.20%
150	3.18%	4.44%	4.34%	4.18%	4.25%	2.87%	3.28%	2.80%	3.12%

Figure 5. Overall SLA violation

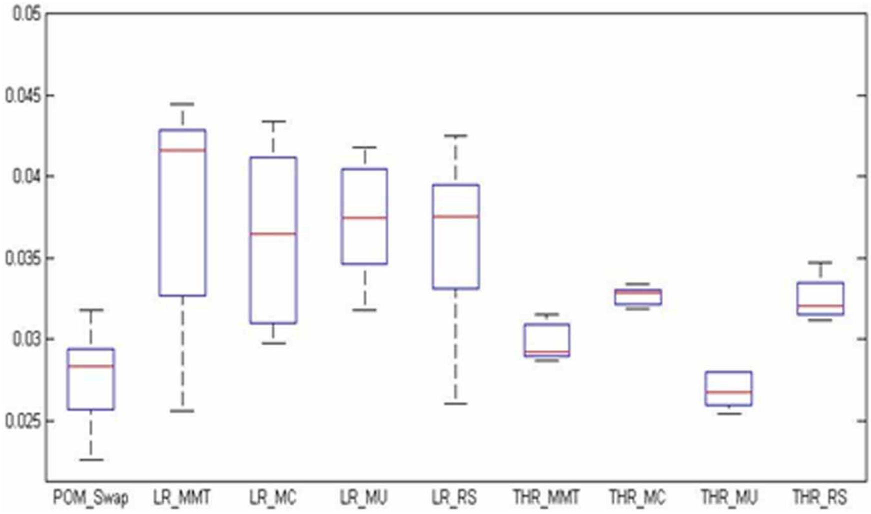
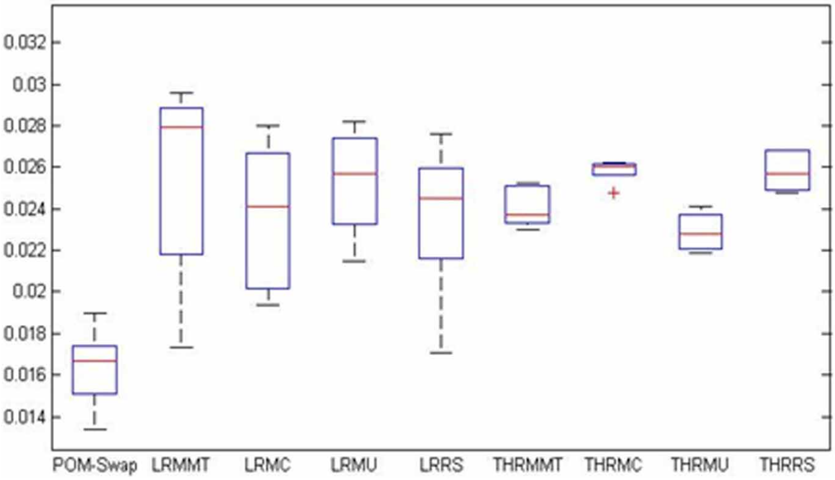


Figure 6. ESV



generate the optimal consolidation plan with minimal SLA violation. A simulative case study suggests that the proposed strategy outperforms traditional methods in terms of energy saving and SLA violation.

ACKNOWLEDGMENT

This work is supported by NSFC under Grant No. 61472051; Fundamental Research Funds for the Central Universities under project Nos. 106112014CDJZR185503 and CDJZR12180012;Science foundation of Chongqing Nos. cstc2014jcyjA40010 and cstc2014jcyjA90027;Chongqing Social Undertakings and Livelihood Security Science and Technology Innovation Project Special Program No. cstc2016shmszx90002; China Postdoctoral Science Foundation No.2015M570770; Chongqing Postdoctoral Science special Foundation No. Xm2015078; Universities' Sci-tech Achievements Transformation Project of Chongqing No. KJZH17104.

REFERENCES

Beloglazov, A., & Buyya, R. (2012). Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers. *Concurrency and Computation*, *24*(13), 1397–1420. doi:10.1002/cpe.1867

Beloglazov, A., & Buyya, R. (2013). Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service Constraints. *IEEE Transactions on Parallel and Distributed Systems*, *24*(7), 1366–1379. doi:10.1109/TPDS.2012.240

Bobroff, N., Kochut, A., & Beaty, K. (2007). Dynamic placement of virtual machines for managing SLA violations. In *Proceedings of the IFIP/IEEE International Symposium on Integrated Network Management* (pp. 119–128). doi:10.1109/INM.2007.374776

Deng, S. G., Huang, L. T., & Xu, G. D. (2014). Social network-based service recommendation with trust enhancement. *Expert Systems with Applications*, *41*(18), 8075–8084. doi:10.1016/j.eswa.2014.07.012

Farahnakian, F., Ashraf, A., Liljeberg, P., Pahikkala, T., Plosila, J., Porres, I., & Tenhunen, H. (2014). Energy-aware dynamic VM consolidation in cloud data centers using ant colony system. In *Proceedings of the 2014 IEEE 7th International Conference on Cloud Computing (CLOUD 2014)* (pp. 104–111).

Farahnakian, F., Liljeberg, P., & Plosila, J. (2014). Energy-efficient virtual machines consolidation in cloud data centers using reinforcement learning. In *Proceedings of the 22nd Euromicro International Conference on Parallel, Distributed and Network-based Processing (PDP)* (pp. 500–507). doi:10.1109/PDP.2014.109

F. Farahnakian, T. Pahikkala, P. Liljeberg, J. Plosila, Nguyen T. H., & H. Tenhunen (2016). Energy-aware VM Consolidation in Cloud Data Centers Using Utilization Prediction Model. *IEEE Transaction on Cloud Computing*.

Graubner, P., Schmidt, M., & Freisleben, B. (2013). Energy-efficient virtual machine consolidation. *IT Professional*, *15*(2), 28–34. doi:10.1109/MITP.2012.48

Huang, J., Duan, Q., Guo, S., Yan, Y., & Yu, S. (2015). Converged Network-Cloud Service Composition with End-to-End Performance Guarantee. *IEEE Transactions on Cloud Computing*.

Huang, J., Li, S., Duan, Q., Yu, R., & Yu, S. (2016). QoS Correlation-Aware Service Composition for Unified Network-Cloud Service Provisioning. In *Proceedings of the 2016 IEEE Global Communications Conference (GLOBECOM)* (pp. 1-6). doi:10.1109/GLOCOM.2016.7842186

Kusic, D., Kephart, J. O., Hanson, J. E., Kandasamy, N., & Jiang, G. (2009). Power and performance management of virtualized computing environments via lookahead control. *Cluster Computing*, *12*(1), 1–15. doi:10.1007/s10586-008-0070-y

Monil, M., & Rahman, R. M. (2016). VM consolidation approach based on heuristics, fuzzy logic, and migration control, Journal of Cloud Computing: Advances. *Systems and Applications*, *5*(1), 8.

Nathuji, R., & Schwan, K. (2007). VirtualPower: Coordinated power management in virtualized server systems. *Operating Systems Review*, *41*(6), 265–278. doi:10.1145/1323293.1294287

Srikantaiah, S., Kansal, A., & Zhao, F. (2008). Energy aware consolidation for cloud computing. *Proceedings of the 2008 conference on Power aware computing and systems HotPower'08* (Vol. 12, p. 10).

Tian, G., Wang, J., He, K. Q., Sun, C. G., & Tian, Y. (2017). Integrating Implicit Feedbacks for Time-aware Web Service Recommendations. *Information Systems Frontiers*, *19*(1), 75–89. doi:10.1007/s10796-015-9590-1

Verma, A., Ahuja, P., & Neogi, A. (2008). pMapper: Power and migration cost aware application placement in virtualized systems. In *Proceedings of the ACM/IFIP/USENIX International Conference on Distributed Systems Platforms and Open Distributed Processing* (*Vol. 12,* pp. 243-264). doi:10.1007/978-3-540-89856-6_13

Verma, A., Dasgupta, G., De, P., & Kothari, R. (2009). Server workload analysis for power minimization using consolidation. In *Proceedings of the 2009 conference on USENIX Annual technical conference* (pp. 28-28).

Vogels, W. (2008). Beyond Server Consolidation. *ACM Queue; Tomorrows Computing Today*, *6*(1), 20–26. doi:10.1145/1348583.1348590

Wang, S. G., You, D., & Zhou, M. C. (2017). A Necessary and Sufficient Condition for a Resource Subset to Generate a Strict Minimal Siphon in S4PR. *IEEE Trans. Automatic Control*.

Wood, T., Shenoy, P., Venkataramani, A., & Yousif, M. (2009). Sandpiper: Black-box and gray-box resource management for virtual machines. *Computer Networks*, *53*(7), 2923–2938. doi:10.1016/j.comnet.2009.04.014

Wu, Q. W., & Fuyuki, I. (2015). Heterogeneous VM consolidation using an improved grouping genetic algorithm. In *Proceedings of the IEEE 17th International Conference on High Performance Computing and Communications* (pp. 397-404).

Xia, Y. N., Zhou, M. C., Luo, X., Zhu, Q. S., Li, J., & Huang, Y. (2015). Stochastic Modeling and Quality Evaluation of Infrastructure-as-a-Service Clouds. *IEEE Transactions on Automation Science and Engineering*, *12*(1), 162–170. doi:10.1109/TASE.2013.2276477

Xiao, Z., Song, W., & Chen, Q. (2013). Dynamic Resource Allocation Using Virtual Machines for Cloud Computing Environment. *IEEE Transactions on Parallel and Distributed Systems*, *24*(6), 1107–1117. doi:10.1109/TPDS.2012.283

Wei Ling Li received the BS degree and MEng degree from Chongqing University, Chongqing, China, in 2010, 2014, respectively. He is currently a PhD candidate in software engineering.

YunNi Xia received the BS degree in computer science from Chongqing University, Chongqing, China, 2003, and the PhD degrees in computer science from Peking University, Beijing, China, 2008. He is currently a Professor with the College of Computer Science and Technology, Chongqing University. He has authored or coauthored over 50 research publications. His current research interests include Petri nets, software quality, performance evaluation, and cloud computing system.

Xin Luo received the BS degree in computer science from the University of Electronic Science and Technology of China, Chengdu, China, in 2005, and the PhD degree in computer science from Beihang University, Beijing, China, in 2011. He joined the Chongqing Institute of Green and Intelligent Technology, Chinese Academy of Sciences, Chongqing, China in 2016, and is currently a Professor of computer science and engineering. His research interests include big data analysis and intelligent control. He has published 60+ papers in his related areas.

Quanwang Wu is currently a lecturer of Computer College in Chongqing University, Chongqing, China. He received his BS, MS, and PhD degrees in computer science from Chongqing University in 2007, 2010, and 2013. He was a special researcher at the digital content and media sciences research division of the National Institute of Informatics (NII) in Tokyo, Japan from 2014 to 2015. His main research interests include service- oriented computing and cloud computing.