2013 Seventh International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing

An Energy Efficient Virtual Machine Placement Algorithm with Balanced Resource Utilization

Wei Huang*, Xin Li†, Zhuzhong Qian‡,
*School of Computer Engineering, Nanjing Institute of Technology, China
†‡State Key Laboratory for Novel Software Technology, Nanjing University, China
*wweihuang@sina.com, †lixin@dislab.nju.edu.cn, ‡qzz@nju.edu.cn

***Abstract*—Virtualization is a key technology for resource sharing in IaaS/PaaS cloud infrastructures. One primary issue in virtualization is the virtual machine placement (VMP) problem, which is to choose proper physical machine (PM) to deploy virtual machines (VMs) in runtime. In this paper, we study the VMP problem with the goal of minimizing the total energy consumption. We first present a multi-dimensional space partition model to characterize the resource usage states of PMs. Based on this model, we then propose a virtual machine placement algorithm, which can balance the utilization of multi-dimensional resources, reduce the number of running PMs and thus lower down the energy consumption. We also evaluate our proposed balanced algorithm via extensive simulations. Simulation results show that our approach can save as much as 15% energy compared to the first fit algorithm over a long run.**

I. Introduction

In recent years, cloud computing has become a popular paradigm for providing scalable and cost-effective services [1][5]. In IaaS/PaaS cloud infrastructures, one key technology for resource sharing is server virtualization, where *virtual machines* (VMs), each of which acts like a real computer with an operating system, are created on underlying *physical machines* (PMs). With such virtualization, resources can be scheduled with fine-granularity which improves resource utilization.

One key issue in server virtualization is virtual machine placement (VMP), which is to select some suitable PM to deploy each new required VM in runtime. One important concern during the VMP process is to reduce the energy consumption caused by the running of PMs, so as to (i) decrease the energy cost which composes a significant fraction of the total cloud costs and keeps on growing since the unit price of energy is raising over time, and (ii) make the cloud computing more energy-sustainable and greener.

One efficient way to cut down the energy consumption is to reduce the number of running PMs [7]. As different VMs require different kinds and amounts of resources (e.g., CPU, memory, disk, bandwidth, etc.), when placing VMs on PMs to meet such requirements, a PM cannot host any more VM as long as an arbitrary dimensional resource is exhausted even when all other dimensional resources are sufficient, a phenomenon well-known as *bucket effects*. Thus, in those fully loaded PMs, all the unutilized resources, referred to as *resource fragments*, are wasted. So in order to improve the resource utilization such that the number of running PMs is minimized, we have to lower down the number of *resource fragments* and decrease their sizes as well.

Since the *resource fragments* originate from the imbalanced use of resources over different dimensions, the placement of VMs on PMs should be executed in a resource balanced manner. Generally, our main work and contribution are summarized as follows:

Our main contributions are as follows:

- We introduce a multi-dimensional space partition model to characterize the current resource usage of each PM, which can be used to guide the design of resource-balanced VM placement algorithm. In this model, each dimension of the space corresponds to one dimensional resource, and the whole space is partitioned into three domains with distinctive features, which indicate the suitability of resource utilization for each placement task.
- We propose an energy efficient VM placement algorithm. Based on the multi-dimensional space partition model, we further propose an energy efficient online VM placement algorithm in balanced manner. When a new VM placement task arrives, our algorithm checks the posterior resource usage state for each feasible PM, and then chooses the most suitable PM according to our proposed model. Compared to traditional greedy algorithms such as *first-fit* algorithm which place the VM on any arbitrary feasible PM, our algorithm can make the resource be utilized in a more balanced manner, thus introduces less and smaller resource fragments and further consumes less energy.
- We evaluate the algorithm performance via extensive simulations, which reveal that our algorithm can save significant portion of energy compared to the *first-fit* algorithm.

The reminder of the paper is organized as follows. Section II introduces the problem formulation and lower bound of the optimal solution. In Section III, we propose the multi-dimensional space partition model. Our algorithm is presented in Section IV. Experiment results are shown in Section V. Section VI discusses some related work, and Section VII concludes this paper.

II. Problem Statement

In this section, we provide a formal definition of our problem and a lower bound of optimal solution. Then, we analyze the energy consumption formally.

978-0-7695-4974-3/13 $26.00 © 2013 IEEE
DOI 10.1109/IMIS.2013.59

A. Problem Definition

We consider a cloud platform consisting of unlimited uniform PMs, the cloud provides the service of renting VMs, e.g. Amazon EC2. The cloud tenants put forward their VM requests and the cloud platform provides VMs to meet their requirements. The process of creating VMs to meet the requirements will be called as virtual machine placement (VMP) in short and it can be understood as there are some VMs to be hosted on the PMs in cloud. We will not distinguish between *VM* and *VM request* in this paper, and so does *cloud* and *cloud platform*.

To formalize our problem, we make the following assumptions and settings:

- Time is divided into time-slots with equal length Δt. There are $K(\tau)$ VMs to be placed at the τ^{th} time-slot.
- PM consists of D-dimensional resource with the same capacity $\mathcal{C}$.
- We use p to identify PM and v to identify VM respectively.
- $N(\tau)$ and $M(\tau)$ are used to represent the total number of PMs and VMs at the τ^{th} time-slot.
- We use $S(T)$ to represent a VM (request) sequence with T time-slots.

Learn from the assumptions and settings, we have

$$M(\tau) = \sum_{t=1}^{\tau} K(t)$$

In τ^{th} time-slot, PMs are selected to host the $K(\tau)$ VMs, the D-dimensional resource should subject the primary resource constraint.

$$\mathcal{C} \geq \sum_{v=1}^{M(\tau)} A_{vp} * R_v^d, \forall 1 \leq p \leq N(\tau), 1 \leq d \leq D \qquad (1)$$

where $R_v^d (1 \leq d \leq D)$ represents the amount of the d^{th} dimensional resource of the v^{th} VM and

$$A_{vp} = \begin{cases} 1, & \text{if } v^{th} \text{ VM is placed on } p^{th} \text{ PM;} \\ 0, & \text{otherwise.} \end{cases}$$

The objective is minimizing energy consumption. It is believed that reduce the number of PMs can cut down energy consumption [7]. We will use *the number of PMs* as the indication of *energy consumption* in this paper. Section II-B gives the formalized interpretation.

We introduce a function $\boldsymbol{\pi}$ to indicate the placement schema. We let $\boldsymbol{\pi}(v) = p$ if the v^{th} VM is placed on the p^{th} PM. The online VMP problem can be formalized as:

Definition 1 (Online VMP Problem). Given VM sequence $S(T)$ and unlimited uniform PMs with capacity $\mathcal{C}$, find a placement function $\boldsymbol{\pi}$, subject to the primary resource constraint (shown in equation 1), such that $N(\tau)(\forall \tau \leq T)$ is minimized.

For the given $S(T)$, minimizing $N(T)$ is a typical multi-dimensional vector bin packing problem, which has been proved to be NP-complete. This indicates that the online VMP problem is an NP-complete problem.

Though it is hard to get optimal solution, we can easily give a lower bound for the placement.

$$N(\tau) \geq \max_{1 \leq d \leq D} \{\frac{1}{\mathcal{C}} * \sum_{v=1}^{M(\tau)} R_v^d\}$$

B. Energy Analysis

Since *energy* can be represented as $E = P * T$, where E and P refer to *energy* and *power*, while T means time period. We have the basic equation:

$$E = \int_1^T P(t)dt = \Delta t \sum_{\tau=1}^{T} P(\tau)$$

Experiments (Section V-A) show that *power* is mainly determined by the *CPU utilization*, it can be formulated as:

$$P = a * r + b$$

where r is *CPU utilization ratio*, a and b are constants associated with the special resource configuration of PM. Hence, E can be represented as:

$$\begin{aligned} E &= \Delta t \sum_{\tau=1}^{T} P(\tau) \\ &= \Delta t \sum_{\tau=1}^{T} \sum_{j=1}^{N(\tau)} P_j \\ &= \Delta t \sum_{\tau=1}^{T} \sum_{j=1}^{N(\tau)} a * r_j + b \\ &= \Delta t \sum_{\tau=1}^{T} \sum_{j=1}^{N(\tau)} a * \sum_{i=1}^{M(j)} R_i * \frac{1}{\mathcal{C}} + b \\ &= \frac{a * \Delta t}{\mathcal{C}} \sum_{\tau=1}^{T} \sum_{j=1}^{N(\tau)} \sum_{i=1}^{M(j)} R_i + b * \Delta t \sum_{\tau=1}^{T} N(\tau) \end{aligned}$$

where $M(j)$ is the current number of VMs hosted on the j^{th} PM. For the given vM request sequence, the number of VMs at τ^{th} time-slot can be represented as:

$$M(\tau) = \sum_{j=1}^{M(\tau)} M(j)$$

So, E can be represented as:

$$E == \frac{a * \Delta t}{\mathcal{C}} \sum_{\tau=1}^{T} \sum_{v=1}^{M(\tau)} R_v + b * \Delta t \sum_{\tau=1}^{T} N(\tau)$$

where R_v refers to the volume of CPU of the v^{th} VM.

For given VM sequence $S(T)$, $N(\tau)$ is the only one variable. Learn from the equation, we know that minimizing $N(\tau)$ can significantly reduce energy consumption.

III. Multi-dimensional Space Partition Model

As mentioned above, due to the dimensionality of resource, *resource leak* occurs in the PM. To represent the utilization of multi-dimensional resource and judge the status of PM, we introduce vector space partition model to measure and guide the placement procedure. PM will be fully loaded if an arbitrary dimensional resource is exhausted. Then, new PM starts up to host other VMs.

There may be resource fragment in the fully loaded PM. Obviously, the resource fragment is wasted, and the phenomenon of resource wasting is named *resource leak*. In order to minimize the number of PMs under given VM sequence, it is necessary to reduce *resource fragment*.

To represent the D-dimensional resource utilization of PM, we propose a multi-dimensional space partition model, which can also judge the suitability of resource utilization for the VM placement and thereby guide VM placement. The basic idea of multi-dimensional space partition model is characterizing *resource leak* quantitatively. At the first, we formally define the *state vector* of PM.

Definition 2 (*state vector*)**.** Given a PM p, the d^{th} dimensional resource utilization ratio is $\gamma_d (1 \leq d \leq D)$. The *state vector* of PM can be represented as $us(p) = (\gamma_1, \gamma_2, ..., \gamma_D)$.

For each PM contains D-dimensional resource, its state corresponds to one point (one-to-one) in the vector space partition model. There are three domains in the partition model. The domain, to which the PM belongs, is determined by the position of the corresponding point, and the domain indicates whether *resource leak* occurs in the PM.

Figure 1 shows an example when $D = 2$. The point $\boldsymbol{E}$ with coordinate $(1.0, 1.0)$ refers to the *state vector* that all dimensional resources are exhausted, while the point $\boldsymbol{O}$ is initial point which means the PM is idle. There are three domains in the partition model.

Acceptance Domain (AD): The acceptance domain implies that the D-dimensional resource are all nearly exhausted and there is few resource fragment in the PM. It is an ideal case for all PMs.

Forbidden Domain (FD): The forbidden domain indicates that there is obvious disequilibrium of D-dimensional resource utilization and there may be excessive *resource fragment* in the PM. This case should be avoided.

Safety Domain (SD): The safety domain means that there is no obvious disequilibrium of D-dimensional resource utilization. This is the balanced case.

The unit square is partitioned into three parts by three quarter circles. The acceptance domain part is one quarter circle with center $\boldsymbol{E}$ and radius $r_0 (r_0 \in [0, 1.0])$. The forbidden domain part is divided into two subparts by two symmetrical quarter circles with the same radius $R_0 (R_0 \in [0, 1.0])$. $\boldsymbol{O_1}$ (red point) is the center of the red quarter circle and $\boldsymbol{O_2}$ (green point) is the center of the green quarter circle. Safety domain lies in the part between the three quarter circles and square edges. The acceptance domain has higher priority if there exists overlapped zone.

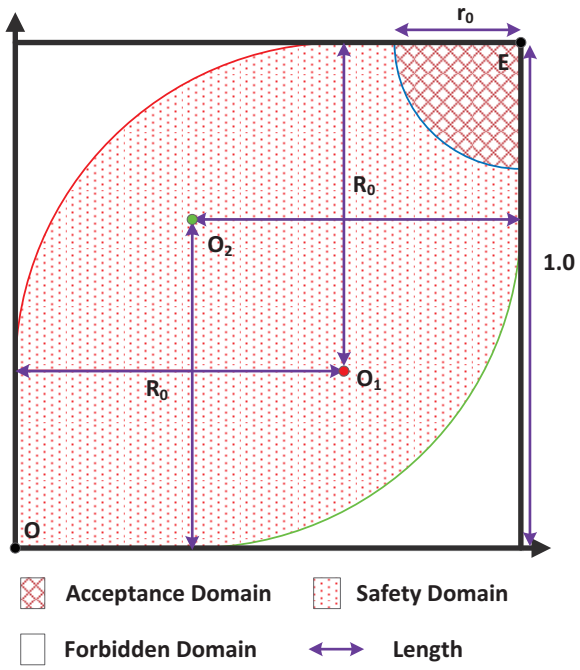


Fig. 1. Multi-dimensional space partition model

Generally, for the D-dimensional space partition model, given point $\boldsymbol{S}(\gamma_1, \gamma_2, ..., \gamma_D)$ and the parameters r_0 and R_0, the domain determination function $\boldsymbol{f}$ can be defined as:

$$\boldsymbol{f}(S) = \begin{cases} AD, & \text{if } distance(S, E) \leq r_0; \\ SD, & \text{else if } ((\forall d, distance(S, O_d) \leq R_0) \| \\ & (\forall d, \gamma_d \leq (1 - R_0)) \| \\ & (\forall d, \gamma_d \geq R_0)); \\ FD, & \text{otherwise.} \end{cases}$$

where $distance(\boldsymbol{S}, \boldsymbol{E})$ means the distance between the two points $\boldsymbol{S}$ and $\boldsymbol{E}(1, 1, ..., 1)$, the coordinate of point $\boldsymbol{O_d}$ is represented as $(\alpha_1, \alpha_2, ..., \alpha_D)$, where

$$\alpha_k = \begin{cases} 1 - R_0, & \text{if } k = d; \\ R_0, & \text{otherwise.} \end{cases}$$

Given the parameters r_0 and R_0, it is easy to determine which domain the point (γ_1, γ_2) belongs to. We can judge the suitability of PM for hosting some VM according to the domain feature of PM.

IV. An Energy Efficient Virtual Machine Placement Algorithm

In this section, we propose an online algorithm based on the partition model. The basic idea of our balanced algorithm is making a tradeoff between balancing multi-dimensional resource utilization and minimizing the *local* number of PMs when placing VMs at each time-slot. The basic operation is starting up a new PM to avoid excessive *resource fragment*. It is efficiency to decrease *resource fragment* over the long run and further reduce the number of PMs. Algorithm 1 shows the framework of our approach.

A. PM Selection

At the τ^{th} time-slot, the cloud accepts $K(\tau)$ VM requests. Some PM(s) will be selected to host the VM(s). According to the *state vector* of PM and VM *resource volume*, we define the *potential state vector* as follows:

Definition 3 (*potential state vector*)**.** Given a PM p, which has sufficient resource to host a VM v. The D-dimensional resource of VM v is $(R_1, R_2, ..., R_D)$. The *state vector* of PM is $(\gamma_1, \gamma_2, ..., \gamma_D)$ and the capacity is $\mathcal{C}$. The *potential state vector* refers to the new *state vector* when the VM is placed on the PM and the *potential state vector* can be represented as $pus(p, v) = (\gamma'_1, \gamma'_2, ..., \gamma'_D)$, where $\gamma'_d = (\mathcal{C} * \gamma_d + R_d)/\mathcal{C}$.

Potential state vector indicates the suitability of the PM for hosting the VM. The suitability is corresponding to the domain in which the *potential state vector* lies. The PM, whose *potential state vector* lies in *acceptance domain (AD)*, has the priority to be selected, while lies in *safety domain (SD)* has the secondary priority. The *potential state vector* lies in *forbidden domain (FD)* implies that excessive *resource leak* will occur in the PM and the PM will not be selected. A new PM will start up if there is no PM is selected.

B. Metric Determination

There may be many PMs with the same priority, the one with optimal suitability will be selected. We provide the definitions of the *metric* $\mathfrak{R}$ and $\mathfrak{D}$ as follows:

Definition 4 ($\mathfrak{R}$: *resource utilization ratio*)**.** Given PM with *(potential) state vector* $(\gamma_1, \gamma_2, ..., \gamma_D)$, $\mathfrak{R}$ is defined as:

$$\mathfrak{R} = \frac{1}{D} \sum_{d=1}^{D} \gamma_d$$

The *metric* $\mathfrak{R}$ represents the mean value of the D-dimensional resource utilization ratios, and it indicates the overall resource utilization of the PM.

Definition 5 ($\mathfrak{D}$: *distance*)**.** Given PM with *(potential) state vector* $(\gamma_1, \gamma_2, ..., \gamma_D)$, $\mathfrak{D}$ is defined as:

$$\mathfrak{D} = \sqrt{\sum_{d=1}^{D} (\gamma_d - \frac{1}{D} \sum_{d=1}^{D} \gamma_d)}$$

The *metric* $\mathfrak{D}$ represents the distance from the *state vector* point to the line $x_1 = x_2 = ... = x_D$, and it indicates the equilibrium of D-dimensional resource utilization.

According to the *metrics*, some PM is selected or new PM starts up. It is unavoidable for PMs that there exists *resource fragment* during the onlie placement process. However, excessive *resource leak* can be avoided by balancing the D-dimensional resource utilization.

V. Performance Evaluation

In this section, we first give the power model of PM, then evaluate the online balanced algorithm. We present comparative results between our algorithm and other algorithms and we evaluate the impact of *balance factor* R_0 and *satisfaction factor* r_0.

Algorithm 1 Framework of Balanced Algorithm

Input: VM sequence *S*
Output: The number of PMs

initial running PM number N, initial time-slot τ
fetch the VM requests $R(\tau)$
while $R(\tau)! = NULL$ **do**
  $K(\tau) \leftarrow R(\tau).size$
  for $k = 1$ to $K(\tau)$ **do**
    $R \leftarrow 0$
    $D \leftarrow \infty$
    $pm = \emptyset$
    for $eachPMp$ **do**
      if $pus(p, k) \in AD$ **then**
        if $R < \mathfrak{R}(p)$ **then**
          $R = \mathfrak{R}(p)$
          $pm = p$
        end if
      else if $pus(p, k) \in SD$ **then**
        if $PM = \emptyset$ **then**
          if $D > \mathfrak{D}(p)$ **then**
            $D = \mathfrak{D}(p)$
            $pm = p$
          end if
        end if
      else
        start new PM, $pm = newPM()$
        $N \leftarrow N + 1$
      end if
    end for
    place the VM on pm
  end for
  record $N(\tau)$
  $\tau \leftarrow \tau + 1$
  fetch $R(\tau)$
end while
return $N(\tau)sequence$

A. Power Model

We measure the power of PM via a power monitor device, and gathere a lot of real data including power, CPU utilization ratio, memory occupation. We analyze the data and establish a power model of power consumption. We found that power is mainly determined by the utilization of CPU, figure 2 shown some sample of the data. According to plentiful analysis and verification, we confirm that the conclusion is:

$$P = a * r + b$$

where r is *CPU utilization ratio* and $r \in (0, 1]$, a and b are constants associated with the special resource configuration of PM. For example, for the sample in figure 2, the values are: $a = 61.24, b = 35.70$.

B. Simulation Setup

In our simulations, the scenario is set as: there are $K(\tau)$ VM requests in the τ^{th} time-slot and $K(\tau)$ is a random

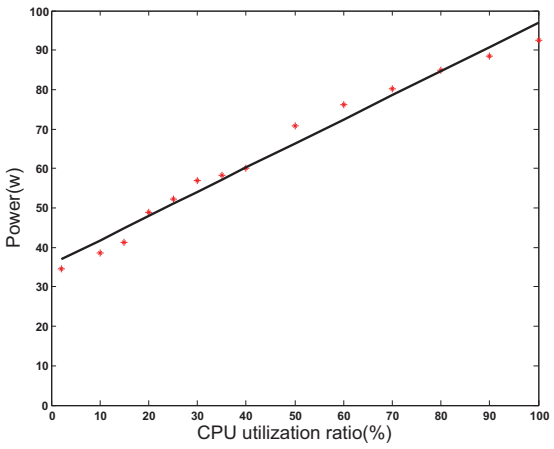


Fig. 2. Relationship between power and CPU utilization ratio.

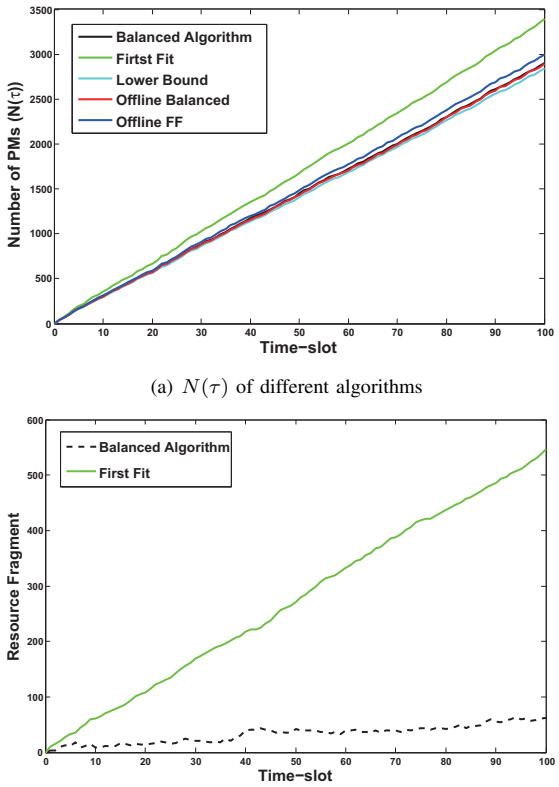
value between 60 and 100. We use the D-dimensional space partition model in our algorithm, and make $D = 2$, it can be considered as *CPU* and *memory*. The capacity of resource is 12, it is consistent with the real situation that physical server owns 12GB (or 24GB) memory and 12 (or 24) CPU cores. The required resource is a random value between 1 and 8. It is reasonable that different tasks have variant resource requirements, e.g., some computational tasks may be sensitive to CPU while some web services need more memory. Exactly because of this differentiation, the disequilibrium of multi-dimensional resource utilization exists and then it results in *resource leak*.

C. Performance Analysis

Besides our balanced algorithm, we implement other three algorithms. Among algorithms for bin packing problem, *first fit* is a well known algorithm and its performance has been proved well. We implement an online *first fit (FF)* algorithm without considering *resource leak*. Furthermore, we implement two offline algorithms named *offline-balanced* and *offline-FF*. The two algorithms give the results of offline VMP. At the same time, we give the value of lower bound.

We use the metric $N(\tau)$, number of running PMs at τ^{th} time-slot, to evaluate the performance of the algorithms. Meanwhile, we illustrate the ratio of other algorithms to balanced algorithm and lower bound, which reflects the difference of algorithm performance clearer.

Learn from Figure 3(a), which shows the result of $N(\tau)$, we know that the performance of *balanced algorithm* (black line) is nearly same as *offline-balanced* (cyan line) and it is very close to the *lower bound* (red line). As time-slot increases, there is visible gap between *balanced algorithm* and *first fit (FF)* algorithm (green line). Balanced algorithm performs even better than *offline-FF* (blue line). Figure 3(b) shows the result of *resource fragment* of *balanced algorithm* and *first fit*. The ordinate axis represents the accumulated value of resource fragments distributed in all of the $N(\tau)$ PMs. It is clear that



(a) $N(\tau)$ of different algorithms

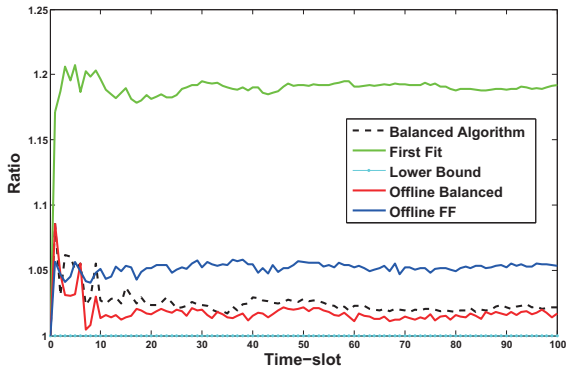
(b) Resource fragment: balanced algorithm vs. first fit

Fig. 3. Performance comparison of different algorithms

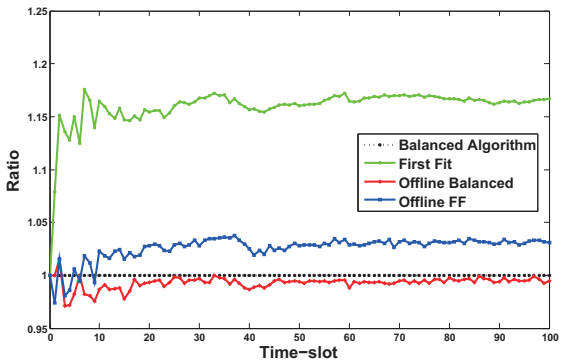
there exists more *resource fragments* while adopting *first fit* algorithm.

To illustrate the performance of *balanced algorithm* directly, figure 4 shows the comparative results, the ordinate axis denotes the ratio of $N(\tau)$ associated with other algorithms to the $N(\tau)$ determined by *balanced algorithm* or *lower bound*, in figure 4(b) and figure 4(a) respectively. At the primary stage of placement, the performance fluctuates. Then, the lines flatten out after the period of fluctuation. Compared to *balanced algorithm*, *first fit* needs 1.15 times PMs. This indicates that we can save near 15% energy cost via adopting *balanced algorithm* algorithm to accomplish the placement task.

Impact of balancing resource utilization. The comparative results of various R_0 with fixed $r_0 = 0.0, R_0 = 0.1$ are shown in figure 5. $r_0 = 0.0$ implies that only the idea of balancing multi-dimensional resource works. The PM selection result is affected by R_0 directly. In the Fig. 5, The ordinary axis is the ratio of the corresponding $N(\tau)$ of various R_0 to the $N(\tau)$ associated with $R_0 = 1.0$, which is adopted in the related simulations of figure 3 and 4. Learn from the result, the performance improves when the value of R_0 varies from 0.0 to 0.8. The result indicates that it is better to enhance the balance restriction, i.e. increase *balance factor* R_0, we can benefit from balancing resource utilization. However, if the balance



(a) Ratio of $N(\tau)$: other-algorithms/lower-bound



(b) Ratio of $N(\tau)$: other-algorithms/balanced-algorithm

Fig. 4. Performance comparison

restriction is immoderate, e.g. the case when $R_0 = 1.0$, there will be an negative effect. Generally, the idea of *balancing* resource utilization shows remarkable improvement on energy saving.

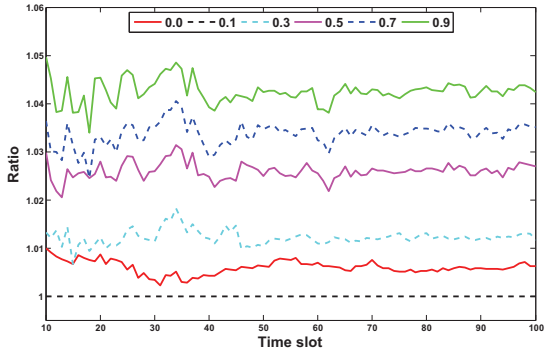


Fig. 5. Impact of balancing resource utilization

VI. Related Work

The virtual machine placement problem under multiple constraints is highly complicated. The constraints includes capacity, network, availability, and performance. [22] optimizes the placement of VMs in a traffic-aware way. Traffic patterns among VMs were aligned with the communication distance between them, VMs with mutual bandwidth usage are assigned to PMs in close proximity. [24] takes network condition into account to minimize the data transfer time consumption while maintain the application performance. Similarly, [8] formulates the VM consolidation with network bandwidth constrain into a stochastic bin packing problem. The paper shown that the number of PMs is within $(1+\epsilon)(\sqrt{2}+1) - OPT$, for any fixed $\epsilon > 0$. [21] defines a new high-availability property for a VM. It is guaranteed that a VM can be relocated to a non-failed PM without relocating other VMs, when a VM is marked as *k-resilient*. Application profiling technique was employed is [19] to improve resource utilization in the process of VM placement. There is a tradeoff between application performance and resource utilization.

Previous works deal with the various VMP problems with the emphasis on system properties, e.g. scalability, availability, etc. However, these works are founded on a primary assumption, the set of VMs is given beforehand, which always violate the reality. Essentially, the online VMP problem is a process of meeting resource requirements from cloud tenants, which is similar with multi-dimensional vector bin packing problem due to the dimensionality of resource.

For $d \geq 2$, the d-dimensional vector bin packing problem has no APTAS (unless P=NP). [13] analyzes the basic feature of multi-dimensional vector bin packing and gives the lower bound of FFD-based variants for some special cases. [15] shows a $d^{\frac{1}{2}-\epsilon}$ hardness of approximation. [12] shows that one can get $(d+\epsilon)$-approximation in linear time, for any $\epsilon > 0$. [14] proposes a polynomial time approximation scheme for randomized instance of the multi-dimensional vector bin packing. [3] presents a framework for analyzing online bin packing algorithms and propose an algorithm within the framework, the algorithm has asymptotic performance ratio at most 1.58889. Yao [11] shows that no algorithm running in time $o(n \log n)$ can give better than a d-approximation.

We present an online placement algorithm which improves resource utilization via reducing the number of resource fragments and cutting down their sizes. Experiment results confirm the efficiency of our algorithm, the algorithm performance is close to the theoretical lower bound of optimal solution.

VII. Conclusion

In this paper, we address the problem of virtual machine placement in clouds with virtualization. Targeted at minimizing energy consumption. We formulated the online virtual machine placement problem as multi-dimensional resource allocating problem. A resource-oriented online algorithm is proposed to provide heuristic solution of the problem. In order to increase resource utilization, we present a vector space partition model to represent multi-dimensional resource utilization, which ensures balanced usage of multi-dimensional resource.

High resource utilization lowers the number of running physical machines and furthermore cuts down energy consumption. The performance is evaluated by extensive simulations. The experiment results show a significant improvement of energy conservation, more than 15% energy can be saved.

ACKNOWLEDGMENT

This work is partially supported by the National Natural Foundation of China under Grant No. 61073028, 61021062, 61202113; the National Basic Research Program of China (973) under Grant No. 2009CB320705; the Jiangsu Natural Science Foundation under Grant No. BK2011510.

REFERENCES

[1] P. Rainer, M. Erich, and T. Simon. "Evidence and Cloud Computing: The Virtual Machine Introspection Approach," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 4, no. 1: pp. 135-152, 2013.

[2] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, "Virtual Infrastructure Management in Private and Hybrid Clouds," *IEEE Internet Computing*, vol. 13, no. 5, pp. 14-22, 2009.

[3] S.S. Seiden, "On the Online Bin Packing Problem," *Journal of the ACM*, Vol. 49, No. 5, September 2002, pp. 640-671.

[4] M. R. Garey, R. L. Graham and D. S. Johnson, "Resource Constrained Scheduling as Generalized Bin Packing," *Journal of Combinatorical Theory*, vol. 21, no. 3, pp. 257-298, 1976.

[5] S. Yoshiaki, M. Masami, and F. Youji. "A Server-Aided Computation Protocol Revisited for Confidentiality of Cloud Service," *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, vol. 2, no. 2: pp. 83-94, 2011.

[6] G. J.Woeginger, "There is no asymptotic PTAS for two-dimensional vector packing," *Information Processing Letters*, vol. 64, no. 6, pp. 293-297, 1997.

[7] R. Bianchini and R. Rajamony, "Power and Energy Management for Server Systems," *IEEE Computer*, vol. 37, 2004.

[8] M. Wang, X. Meng, and L. Zhang, "Consolidating Virtual Machines with Dynamic Bandwidth Demand in Data Center," *In Porc. of INFOCOM 2011 (Mini-Conference)*.

[9] H. Liu, H. Jin, X. Liao, C. Yu, and C.-Z. Xu, "Live Virtual Machine Migration via Asynchronous Replication and State Synchronization," *IEEE TPDS*, vol. 3, no. 22, pp. 426-438, 2011.

[10] M.R. Garey and D.S. "Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness," *W.H. Freeman*, 1979.

[11] A.C. Yao. "New Algorithms for Bin Packing", *Journal of the ACM (JACM)*, vol. 27, no. 2, pp. 207-227, 1980.

[12] W.F. de la Vega and G.S. Lueker. "Bin Packing can be solved with $1+\epsilon$ in Linear Time", *Combinatorica*, vol. 1, no. 4, pp. 349-355, 1981.

[13] J. Csirik, J.B. Frenk, M. abbe, and S. Zhang. "On the Multidimensional Vector Bin Packing", *Acta Cybernetica (ACTAC)*, vol. 9, no. 4, pp. 361-369, 1990.

[14] D. Karger and K. Onak. "Polynomial Approximation Schemas for smoothed and random Instances of Multi-dimensional Packing Problem", In *Proceeding of the 8th annual ACM-SIAM symposium (SODA)*, USA, 2007, pp.1207-1216.

[15] C. Chekuri and S, Khanna. "On multi-dimensional packing problems," In *ACM-SIAM Symposium on Discrete Algorithms*, pp. 185-194, 1999.

[16] U. Sharma, P. Shenoy, S. Sahu, and A. Shaikh, "A Cost-aware Elasticity Provisioning System for the Cloud," *In Proc. of ICDCS 2011*.

[17] Z. Gong, X. Gu, and J. Wilkes, "PRESS: PRedictive Elastic ReSource Scaling for cloud systems," *In Proc. of CNSM 2010*.

[18] Z. Shen, S. Subbiah, X. Gu, and J. Wilkes, "CloudScale: Elastic Resource Scaling for Multi-Tenant Cloud Systems," *In Proc. of SOCC 2011*.

[19] A. V. Do, J. Chen, C. Wang, Y. C. Lee, A. Y. Zomaya, and B. B. Zhou, "Profiling Applications for Virtual Machine Placement in Clouds," *In Proc. of Cloud Computing 2011*.

[20] J. Rao, X. Bu, K. Wang, and C.-Z. Xu, "Selt-adaptive Provisioning of Virtualized Resource in Cloud Computinng," *In proc. of SIGMETRICS 2011*.

[21] E. Bin, O. Biran, O. Boni, E. Hadad, E. K. Kolodner, Y. Moatti, and D. H. Lorenz, "Guaranteeing High Availability Goals for Virtual Machine Placement," *In Proc. of ICDCS 2011*.

[22] X. Meng, V. Pappas, and L. Zhang, "Improving the Scalability of Date Center Networks with Traffic-aware Virtual Machine Placement," *In Proc. of INFOCOM 2010*.

[23] X. Li, Z. Qian, R. Chi, B. Zhang, and S. Lu, "Balancing Resource Utilization for Continuous Virtual Machine Requests in Clouds," *In Proc. of IMIS 2012*.

[24] J. T. Piao, and J. Yan, "A Network-aware Virtual Machine Placement and Migration Approach in Cloud Computing," *In Proc. of GCC 2010*.

[25] M. Lin, A. Wierman, L. H. Andrew, and E. Thereska, "Dynamic Right-sizing for Power-proportional Data Center," *In Proc. of INFOCOM 2011*.