Future Generation Computer Systems 100 (2019) 380–394

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Virtual machine allocation and migration based on performance-to-power ratio in energy-efficient clouds

Xiaojun Ruan [a,*], Haiquan Chen [b,*], Yun Tian [c], Shu Yin [d]

[a] California State University, East Bay, United States
[b] California State University, Sacramento, United States
[c] California State University, Fullerton, United States
[d] ShanghaiTech University, China

ARTICLE INFO

Article history:
Received 8 September 2018
Received in revised form 24 April 2019
Accepted 14 May 2019
Available online 16 May 2019

ABSTRACT

The last decade witnessed a dramatic advance in cloud computing research and techniques. One of the key challenges in this field is reducing the massive amount of energy consumption in cloud computing data centers. Many power-aware virtual machine (VM) allocation and consolidation approaches were proposed to reduce energy consumption efficiently. However, most of the existing efficient cloud solutions save energy at the cost of significant performance degradation. In this paper, we propose a strategy to calculate the optimized working utilization levels for host computers. As the performance and power data need to be measured on real platforms, to make our design practical, we propose a strategy named "PPRGear" which is based on the sampling of utilization levels with distinct Performance-to-Power Ratios (PPR) calculated as the number of Server Side Java operations completed during a certain time period divided by the average active power consumption in that period. In addition, we present a framework for virtual machine allocation and migration which leverages the PPR for various host types. By achieving the optimal balance between host utilization and energy consumption, our framework is able to ensure that host computers run at the most power-efficient utilization levels, i.e., the levels with the highest PPR, thus tremendously reducing energy consumption with ignorable sacrifice of performance. Our extensive experiments with real world traces show that compared with three baseline energy-efficient VM allocation and selection algorithms, IqrMc, MadMmt, and ThrRs, our framework is able to reduce the energy consumption up to 69.31% for various host computer types with fewer migration times, shutdown times, and little performance degradation for cloud computing data centers.

1. Introduction

Cloud computing has been widely adopted by businesses, individuals, and large enterprises. However, energy consumption has become a big concern in the last decade since cloud data centers consumed significant power and generated giant power bills. According to the data disclosed by The New York Times in 2012, Facebook data centers consumed about 60 million watts and Google data centers consumed as much as almost 300 million watts [1]. In 2013, data centers in the United States collectively consumed 91 billion kWh of electrical energy and generated 97 million metric tons of carbon dioxide (CO_2) [2]. In 2014, more than 2% of the United State's electricity usage was consumed by data centers [3]. Furthermore, by 2020, the annual electricity usage in the United States is expected to be as much as 140 billion kWh which is the output of about 50 power plants [4]. The carbon dioxide emission generated by Information and Communication Technology (ICT) is expected to exceed 1.4 billion metric tons. It is estimated that data centers are responsible for about 18% of the total energy consumed by all ICT systems in the world [5]. Therefore, many energy-efficient approaches have been explored at facility level, in cooling systems [6,7], in data center network [8], and by using computing resource allocation strategies. Among those methods, the computing resource allocation is considered as the most achievable and cost-effective approach since it does not require any hardware modifications or upgrades. Virtualization is a key technology to achieve energy efficiency in data centers. VMs can be created, deleted, and migrated among host computers depending on power-aware decisions [9]. Energy-efficient VM management has been explored in task scheduling [10], workload consolidation [11,12], temperature-aware capping [13], request batching [14], local or remote clouds choosing [15], mobile service selection [16], etc.

* Corresponding author.
E-mail addresses: xiaojun.ruan@csueastbay.edu (X. Ruan), haiquan.chen@csus.edu (H. Chen).

https://doi.org/10.1016/j.future.2019.05.036
0167-739X/

Gelenbe et al. showed that energy consumption in ICT is related to workload, and concluded that the optimal energy consumption and processing time trade-off could be achieved by tuning workloads in computer systems [5]. Their work also indicated that computing systems should turn on more servers when the workload is sufficiently high in order to achieve energy efficiency and acceptable levels of Quality of Service (QoS) [17].

To the best of our knowledge, our work is the first to leverage the Performance-to-Power Ratio (PPR) of computing nodes in VM allocation and migration to achieve the optimal balance between host utilization and energy consumption. Performance-to-Power Ratio is calculated as the number of Server Side Java operations, or ssj_ops, completed during a certain time period divided by the average active power consumption in that period. Most of the current VM placement and migration policies are based on primitive system characteristics like power, utilization, network bandwidth, or storage space. However, in this paper, we propose an energy-efficient VM allocation and migration strategy based on PPR which is not a primitive characteristic of host computers. Our proposed framework is able to dynamically allocate VMs to and migrate VMs among hosts so that host computers can operate at the most power-efficient utilization levels, i.e., at the utilization level with the highest PPR. Specifically, this paper has the following contributions:

- We propose a novel VM allocation and migration framework which allocates and migrates virtual machines in clouds based on host performance-to-power ratios. Under this framework, host computers run at their optimal or near-optimal utilization levels so that the energy consumption can be significantly reduced without much sacrifice of cloud-end computation performance.
- We propose the exact and approximate methods to determine the ranges of gears for a specific host type. Thanks to our sampling strategy, the proposed approximate method is able to efficiently derive the range of each gear and estimate the whole system energy cost. Without loss of generality, in our verification experiments, we assume that each host computer maintains 11 gear levels (from gear 0 to gear 10) that corresponds to distinct utilization levels (from 0%, 10%, …to 100%).
- We develop the VM allocation and migration modules under our proposed PPRGear framework based on the calculation of the Performance-to-Power Ratio (PPR) on host computers. These two modules have been designed seamlessly to trigger virtual machine allocation and migration automatically when a host is overutilized or underutilized in order to achieve the optimal balance between host utilization and energy consumption.
- Our extensive experiments on CloudSim [18] with real-world traces show that compared with ThrRs, MadMmt, and IqrMc [19], our framework is able to reduce the energy consumption significantly for various host computer types. More importantly, the SLA violation rate of our framework is almost the same as that of Dynamic Voltage and Frequency Scaling (DVFS), indicating that our framework results in ignorable performance degradation.

In our design, each host computer maintains 11 gear levels (from gear 0 to gear 10) that correspond to distinct utilization levels (from 0%, 10%, …to 100%). The gear with the highest PPR is chosen as the *best gear*. The top n gears with the highest PPRs are chosen as *preferred gears*. When the current working gear of a host is not in the range of the preferred gears, the current host is considered as either overutilized or underutilized. Before executing any tasks energy-efficiently, we evaluate the characteristics of computing node at different utilization levels. This evaluation finds the best gear with the highest PPR and the n preferred gears with the n highest PPR values. By allocating and migrating VMs in clouds, we aim to keep computing nodes working at the best gears. When a computing node is working at a gear higher than any preferred gears, the computing node is considered overutilized. When a computing node is working at a gear lower than any preferred gears, the computing node is considered underutilized. If a computing node is overutilized, one or multiple VM(s) in this host will be selected and then migrated out. If a computing node is underutilized, the cloud will either migrate VMs from other hosts to this host or migrate out all VMs on this host then shutdown it to save energy consumption.

The remaining part of the paper is organized as follows: Section 2 introduces the motivation and observation, Section 3 presents the preferred utilization and the energy model, Section 4 presents the overview of our approach, Section 5 details the algorithmic design, Section 6 compares PPRGear with the baselines and shows our simulation results, Section 7 presents the related work, Section 8 concludes the paper.

2. Our observations

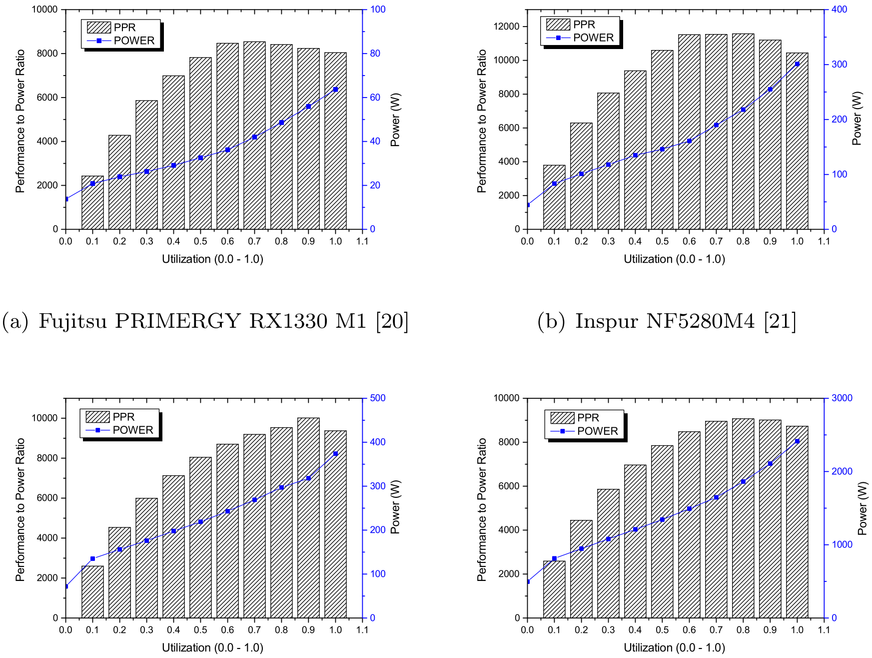
Our paper is focused on energy conservation by improving the effectiveness of energy usage, i.e., accomplishing more tasks with less energy, rather than simply reducing the energy consumption, which is vital for heavy workloads.

Standard Performance and Evaluation Corporation (SPEC) developed an energy benchmark suite *SPECpower_ssj*2008 [20]. A number of corporations have conducted experiments on their host computers by using *SPECpower_ssj*2008 and uploaded experimental results to the SPEC website. Performance-to-Power Ratio, or PPR, is calculated as the number of Server Side Java operations(*ssj_ops*) completed during a certain time period divided by the average active power consumption in that period. PPR indicates the effectiveness of power usage on a computing node.

The *SPECpower_ssj*2008 workload is controlled by system throughput and varies on different computing platforms. For instance, if the current utilization hits 100% when workload is n *ssj_ops* instructions per second, the 20% utilization's corresponding workload will be $0.2 \times n$ instructions per second. Power consumption data of the current computing platform will be collected under workloads 0, $0.1 \times n$, $0.2 \times n$, …, until n instructions per second, which represent the utilization levels from 0%, 10%, 20%, …, until 100%.

Although a few factors, including CPU, memory, hard drives, NIC, and etc., may contribute to the total energy consumption of a computing system, Google's recent research [21] reveals that even by using CPU utilization only, we are able to estimate the power cost of the whole system very accurately. This conclusion was drawn by measuring the total energy cost by running Google's most representative benchmarks and some other micro-benchmarks with a variety of loads. Therefore, in our study, we mainly base our discussion on SPECPower benchmark and we expect that the research findings we obtained also generalize to other types of workloads with different characteristics in terms of CPU, memory, and I/O consumption.

Fig. 1 presents the power consumption trends and PPR trends while CPU utilization increases in four different host models of Fujitsu Primergy RX1330 M1, Inspur NF5280M4, Dell PowerEdge R820, and IBM NeXtScale nx360 M4. According to DVFS, power consumption of CPU increases exponentially while CPU utilization increases linearly. However, the power consumption of a host computer includes CPUs, memory, secondary storage, network adaptors, etc. Möbius states that power consumption model of



(a) Fujitsu PRIMERGY RX1330 M1 [20]

(b) Inspur NF5280M4 [21]

(c) Dell PowerEdge R820 [22]

(d) IBM NeXtScale nx360 [23]

Fig. 1. PPR and power consumption with increase in CPU utilization.

Table 1
Host models [23–26].

Model	CPU	Clock	Cores	RAM	Test date
Fujitsu Primergy RX1330 M1	Intel Xeon E3-1275 8 MB L3 Cache	2.5 GHz	4	16 GB	Jul 30 2014
Inspur NF5280M4	Intel Xeon E5-2699 v3 45 MB L3 Cache	2.3 GHz	18	64 GB	Aug 29 2014
Dell PowerEdge R820	Intel Xeon E5-4650 v2 25 MB L3 Cache	2.4 GHz	40	48 GB	Apr 1 2014
IBM NeXtScale nx360 M4	Intel Xeon E5-2660 v2 25 MB L3 Cache	2.2 GHz	20	24 GB	Mar 17 2014

a host computer solely employs CPU utilization metrics [22]. According to the data collected by *SPECpower_ssj*2008 in Fig. 1, we observe that the power consumption of a host computer increases linearly while CPU utilization increases linearly even though the four host computers have different configurations as presented in Table 1.

Fig. 1(a) presents the power consumption and PPR of host model Fujitsu Primergy RX1330 M1 on utilization from 0 to 1. It is expected that the power consumption increases almost linearly while CPU utilization increases. However, the highest PPR appears at utilization level 0.7, or 70%. In other words, the computing host works the most energy-efficiently at CPU utilization 0.7. Likewise, Figs. 1(b), 1(c), 1(d) present similar PPR trends of host models Inspur NF5280M4, Dell PowerEdge R820, and IBM NeXtScale nx360 M4. Therefore, instead of trying to reduce power by simply decreasing utilization, PPRGear attempts to keep computing nodes working under the highest PPR as long as possible in order to balance utilization and power consumption.

3. Problem formulation

The objective of our work is to achieve the optimal balance between the host utilization and the energy consumption for cloud data centers. Inspired by the aforementioned observations, our proposed mechanism achieves the goal by allocating and migrating VMs so that computing nodes are able to operate at their best gears, i.e., the utilization levels that will result in the highest performance-to-power ratios. In this section, we first focus on the definition and calculation of the best gear and then we present the energy consumption model of PPRGear.

3.1. Discovery of best gear

In this subsection, we are presenting the definition of the best gear using mathematical models and definition. Here the gear with the highest PPR value is considered the best gear, or to say, the most energy-efficient gear. The best gear is named as G and the utilization range is represented as $[\alpha, \alpha + I]$. In our proposed PPRGear framework, when a new host type is added into the cloud, its power consumption pattern at different CPU utilization levels must be learned in advance to identify its best gear to facilitate the subsequent VM allocation and migration.

Definition 1. The *Gear G* of a computing node is a certain utilization level $[x, y]$ where $0 \leq x \leq y \leq 1$. If the current utilization rate of a host is in the range of x to y, we say that this host is operating at gear G.

Definition 2. The *Best Gear* of a computing node is a gear that yields the highest performance-to-power ratio of that computing node, which can be represented as $[\alpha, \alpha+I]$. Here I is a predefined range for each gear and α can be calculated using Eq. (1), where the function $f_{PPR}(u)$ describes the performance-to-power ratio at the utilization rate u $(0 \leq u \leq 1)$.

$$\alpha = \operatorname{argmax} \int_{\alpha}^{\alpha+I} f_{PPR}(u)\, du, \qquad \text{subject to } 0 \leq \alpha \leq \alpha + I \leq 1 \tag{1}$$

The PPR and power values need to be calculated and collected by conducting experiments. Since it is computationally expensive to identify the PPR and power values at all possible utilization levels, the best gear can be approximated efficiently by calculating the PPR values at a certain number of utilization levels. In Fig. 1, the function $f_{PPR}(u)$ for each host type is estimated by measuring the performance-to-power ratios at 11 uniformly distributed utilization rates, i.e., from 0, 0.1, 0.2, 0.3, …, to 1 (here we assume that the interval for each utilization level is 0.1). Take Fig. 1(a) as an example, according to Fig. 1(a), the best gear of a Fujitsu PRIMERGY RX1330 M1 machine should be [0.65, 0.75], which is the gear with the highest performance-to-power ratio on this machine.

Since the PPR and power values are only measured at 11 utilization levels, it is important to accurately calculate the power values at each utilization level. $f_{Power(u)}$ can also be approximated efficiently by sampling over utilization levels. In Fig. 1, $f_{Power(u)}$ is calculated by (1) sampling at the utilization rates from 0, 0.1, 0.2, 0.3, …, to 1 and (2) applying linear interpolation to every two adjacent sampled results. Formally, given any utilization rate u ($U_l \leq u \leq U_h$), the power consumption at u can be estimated efficiently using Eq. (2), where P_h means the power consumption at U_h and P_l is the power consumption at U_l.

$$f_{Power}(u) = \frac{P_h - P_l}{U_h - U_l} U - \frac{P_h U_l - P_l U_h}{U_h - U_l} \tag{2}$$

3.2. Energy cost calculation

Energy consumption at a host depends on the time and the power. Since the power can be represented as a function of utilization rate, as shown in Fig. 1, we may formulate the energy consumption E_{h_i} in the host h_i in terms of utilization rate and time using Eq. (3). In Eq. (3), t_{h_i} is the total working time of host i, $f_{Utilization}(t)$ describes how its utilization rate changes over time t, and $f_{Power(u)}$ means the power at the utilization rate u.

$$E_{h_i} = \int_0^{t_{h_i}} f_{Power}^{h_i}(f_{Utilization}^{h_i}(t))\,\mathrm{d}t \tag{3}$$

Suppose we use N to represent the total number of the hosts in a cloud computing system, the total energy cost E_{All} in the cloud computing system can be estimated using Eq. (4).

$$E_{All} = \sum_{i=1}^{N} \int_0^{t_{h_i}} f_{Power}^{h_i}(f_{Utilization}^{h_i}(t))\,\mathrm{d}t \tag{4}$$

4. System design

Most of current VM placement and migration policies are based on the factors that are primitive system characteristics like power, utilization, network bandwidth, and storage space. We propose an energy-efficient VM allocation and migration strategy based on Performance-to-Power Ratio (PPR) which is not a primitive characteristic of host computers. Before executing any tasks energy-efficiently, we evaluate the characteristics of computing nodes at different utilization levels, called gears. The evaluation indicates the best gear with the highest PPR and the preferred gears with the highest n PPR values. By allocating and migrating VMs in clouds, we attempt to keep computing nodes working at the best gear (or as close as possible). When a computing node is working at a gear that is higher than the preferred gears, the computing node is overutilized; when a computing node is working at a gear lower than the preferred gears, the computing node is underutilized. If a computing node is overutilized due to VM utilization increasing, one or multiple VM(s) will be selected then migrated out. If a computing node is underutilized, the cloud will try to migrate out all VMs on the node then shut the computing node down. This scheme is called PPRGear.

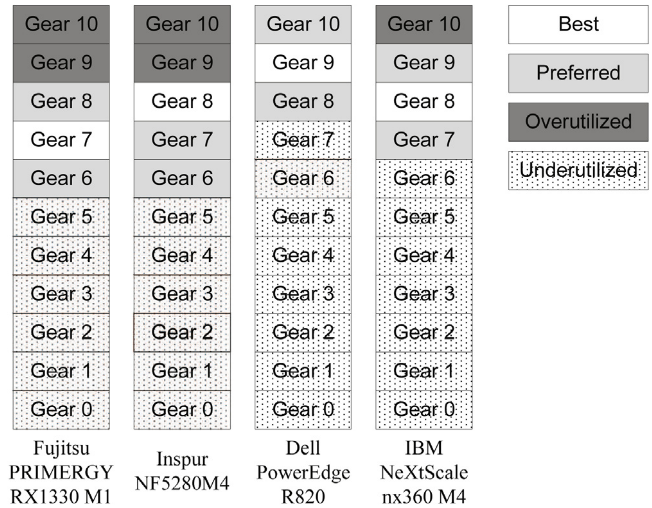


Fig. 2. An illustration of gear selection (3 preferred gears and 1 best gear).

Before a cloud can be utilized energy-efficiently, each computing node in the cloud hardware platform should be evaluated in the following steps to find out the best gear and n preferred gears. First, the performance and the power data is collected by using benchmark suit SPECpower_ssj2008 [20]. In this paper, the performance and the power data was collected by the following vendors: Fujitsu, Inspur, Dell, and IBM. The Table 1 presents the configurations of the hosts used in later experiments. Power is the average power consumption in a certain time interval which is 300 s in this paper. We define performance as *ssj_ops* over time. Second, PPR is calculated simply by performance over power in our experiments for the eleven gears ($n = 11$) on CPU utilization from 0%, 10% … 100%. Third, sort gears on PPR in descending order to find the best gear and n preferred gears.

Fig. 1 indicates that although power consumption increases linearly while utilization increases, the highest PPR values may not appear with the highest power consumption values. Therefore, even though the host computer offers the highest performance at the highest utilization, the CPU is not working as efficiently as working under a lower utilization due to the lower PPR. Based on PPR values, PPRGear decides how to allocate new VMs and how to migrate running VMs.

Fig. 2 demonstrates an example of gear selection for the four host types. In this example, the number of preferred gears is set to 3. According to the PPR values in Fig. 1, the best gear is selected from each group and is marked in white; three preferred gears (including the best gear) are selected and are marked in light gray; all gears higher than preferred gears are selected as overutilized gears and are marked in dark gray; all gears lower than preferred gears are selected as underutilized gears and are marked in dot pattern. Each gear has a corresponding CPU utilization. For example, gear 7 means the current host computer is working at CPU utilization 70%. PPRGear attempts to keep each host working at the preferred gears as long as possible. If the utilization goes below or over the preferred gears, PPRGear will conduct migrations to migrate VMs either out or in. When PPRGear is migrating VMs to a destination host computer, PPRGear attempts to allocate VMs in order to make the destination host working at the CPU utilization level which is the closest to that of the best gear. Fig. 2 indicates that different host models may have different selections for the best and preferred gears based on PPR values. Therefore, allocation and migration may vary due to different gear selections.

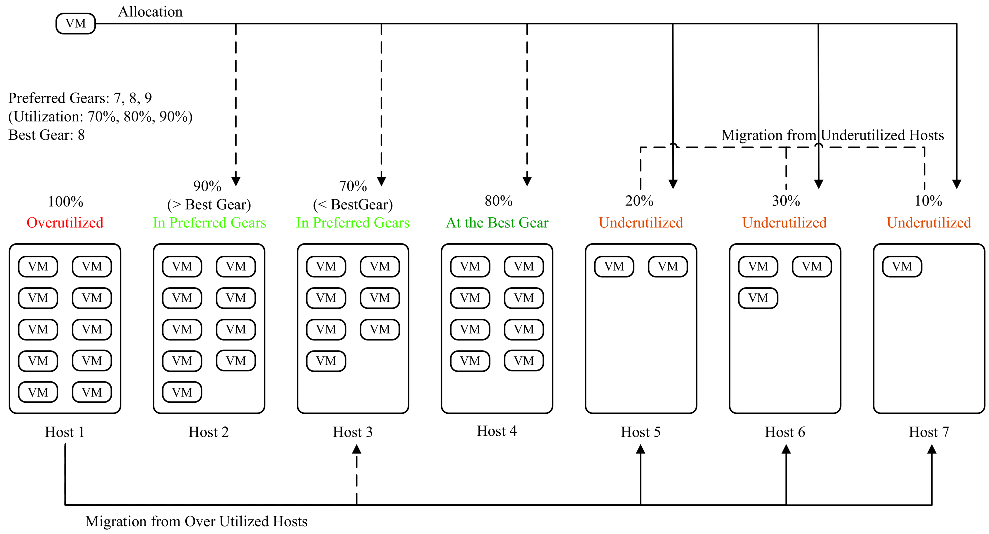


Fig. 3. A possible snapshot of VM allocation and migration (3 machines (Hosts 5, 6, and 7) underutilized, 1 (Host 1) overutilized, 1 (Host 4) at the best gear, 2 (Hosts 2 and 3) at preferred gears). For the sake of presentation, we assume every VM consumes the same amount of host utilization—10%. At this point of time, Algorithm 3 will select two VMs on Host 1 to migrate out to Host 6, which is selected by Algorithm 2 as the destination host. That will reduce Host 1's utilization to the Best Gear range and will make Host 6 closer to its Best Gear. Afterwards, all VMs on Host 5 and Host 7 will migrate out by Algorithm 4 to Host 6. At last, Host 5 and 7 will be shut down due to no VMs on them.

Notice that in Fig. 2, three preferred gears of Dell Power Edge R820 are the exact three gears with the highest utilizations. Therefore, there are no overutilized gears for this host type. But migrations will still be conducted if current VMs require more MIPS that exceeds current host's capability. And it is unlikely to have VMs migrate in since a host would be the last choice of migration destination when it is working at the best gear.

Fig. 3 serves as an example of the VM allocation and migration strategy based on PPRGear framework. Assuming there are 7 hosts in the cloud, Host 1 is overutilized; Host 5, 6, and 7 are underutilized; Host 2 and 3 are working at preferred gears; and Host 4 is working at the best gear. In this case, Host 1 needs to migrate VMs out to lower utilization, Host 5, 6, and 7 need to either migrate VMs in, to increase utilization, or migrate all VMs out to shut down. Underutilized hosts and the hosts at preferred gears are the possible destination hosts.

5. Algorithmic design

In this section, we first provide our algorithmic design of PPRGear and then elaborate on its two modules: VM Allocation and VM Migration.

5.1. Overall design of PPRGear

Algorithm 1 presents the overall algorithm of PPRGear. PPRGear is a daemon process that monitors the cloud computing environment and balances the workload among hosts by conducting migration based on PPR.

Algorithm 1 PPRGear Overall Algorithm

while *True* **do**
  run Algorithm 3: VM Selection on Overutilized Hosts for Migration
  if H_{over} is not empty **then**
    run Algorithm 2: VM Placement Algorithm
  end if
  run Algorithm 4: VM Selection on Underutilized Hosts for Migration
end while

5.2. VM allocation

VM Allocation algorithm is used to place a VM. The VM to be placed is either newly created or migrated from an overutilized host or an underutilized host. VM allocation chooses an appropriate destination host with both computing capability and power consumption concerns. Algorithm 2 presents how PPRGear allocates hosts for VMs. First, PPRGear traverses all hosts to collect current CPU utilization of each host. Current CPU utilization is summed by all the VMs running on or migrating to the host. Second, PPRGear skips all the overutilized hosts even though some of them may still have enough resource to run the VM due to energy-efficient concerns. Third, PPRGear calculates the expected host utilization by summing the current host utilization and the predicted utilization of the VM on the host. If the expected host CPU utilization is at the best gear or at the CPU utilization closest to the best gear (3% tolerance), the current host is considered to be chosen as the destination host for the VM. If no hosts' expected utilization is within the tolerable range, then PPRGear attempts to find the host whose expected utilization is the closest to the best gear utilization.

5.3. VM migration

VMs migrate to other hosts when the current host is either overutilized or underutilized. VMs on overutilized and underutilized hosts will be selected for migration with power consumption concerns. Then VM Allocation algorithm will be in charge of allocating appropriate hosts for the selected VMs. Therefore, VM Migration contains three steps: 1. Detect overutilized hosts and underutilized hosts; 2. Select VMs for migration; 3. Choose appropriate destination hosts. When VMs are migrating from either underutilized or overutilized hosts, PPRGear considers the underutilized hosts as possible destination hosts first. The use of hosts at preferred gears will be the second choice. We concluded two cases to describe PPRGear VM Migration.

Algorithm 2 VM Placement Algorithm

input: H_{list}: all hosts, vm: a new virtual machine, utilizationDiff: MAXVALUE
output: h_{chosen}: the chosen destination host
$h_{chosen} \leftarrow null$
for each h in H_{list} **do**
  utilization $\leftarrow$ 0
  $VM_h \leftarrow$ All Virtual Machines on h
  for each v in VM_h **do**
    utilization $\leftarrow$ utilization + v.utilization
  end for
  if utilization $>$ highest preferred gear utilization $\vee$ utilization + vm.utilization $>$ 1 **then**
    continue
  else if utilization + vm.utilization $>$ best gear Utilization - 0.03 $\wedge$ utilization + vm.utilization $<$ best gear Utilization + 0.03 **then**
    return h
  else if abs(utilization + vm.utilization - best gear utilization) $<$ utilizationDiff) **then**
    utilizationDiff $\leftarrow$ abs(utilization + vm.utilization - best gear utilization)
    $h_{chosen} \leftarrow$ h
  end if
end for
return h_{chosen}

Algorithm 3 VM Selection on Overutilized Hosts for Migration

input: H_{list}: all hosts
output: $V_{migrate}$: selected VMs for migration
$H_{over} \leftarrow \phi$
$V_{migrate} \leftarrow \phi$
for each h in H_{list} **do**
  utilization $\leftarrow$ 0
  $VM_h \leftarrow$ All Virtual Machines on h
  for each vm in VM_h **do**
    utilization $\leftarrow$ utilization + vm.utilization
  end for
  if utilization $>$ utilization of highest preferred gear **then**
    $H_{over} \leftarrow$ h
  end if
end for
for each h in H_{over} **do**
  $VM_h \leftarrow$ Migratable Virtual Machines on h
  Sort VM_h in descending order on utilization
  utilization $\leftarrow$ 0
  $V \leftarrow null$
  while h is overutilized **do**
    for each vm in VM_h **do**
      if migrating out vm can make host utilization closer to the best gear utilization **then**
        V $\leftarrow$ vm
      end if
    end for
  end while
  $V_{migrate} \leftarrow$ V
end for
return $V_{migrate}$

5.3.1. Overutilized hosts detection and VM selection

The utilization on a host varies even though there is no VMs migrate in or out. The reason is that the utilization of each VM also varies as time goes. In other words, a host currently working at the best gear may exceed the highest preferred gear after a while with no VMs migrating in, and vice versa.

Algorithm 3 shows Overutilized Host Detection and VM Selection to migrate VMs out from overutilized hosts. First, PPRGear traverses all hosts to find overutilized hosts and put them in overutilized host list *H_over*. Whether a host is overutilized depends on if the current host CPU utilization is higher than that of the highest preferred gear or not. Therefore, if the number of preferred gears is set large enough, the gear 10 (the highest gear) may be included in the preferred gears. Then the PPRGear will not migrate VMs for the overutilization case since there will be no hosts to be considered as overutilized. Second, for each host h in H_{over}, PPRGear puts all migratable VMs in to list VM_h then sort the list in descending order on utilization. Third, find the VM(s) to migrate out in order to reduce the host utilization as close to the best gear utilization as possible. Then add the VM(s) in the list $V_{migrate}$ for migration. When Algorithm 3 finishes, $V_{migrate}$ contains the VMs to migrate out. Then, PPRGear calls Algorithm 2 to find appropriate destination hosts for VMs in the list $V_{migrate}$.

5.3.2. Underutilized hosts detection and VM selection

Underutilized Hosts Detection checks whether a host's current utilization level is lower than that of the lowest preferred gear. If the number of preferred gears is set large enough, PPRGear may not be able to find any underutilized hosts for migration due to the wide range of preferred gears.

Unlike the case of overutilized hosts, when migrating VMs from underutilized hosts, the destination hosts, which were also underutilized hosts, may turn in to the hosts that work at the preferred gears. In other words, some underutilized hosts could turn in to well utilized hosts after migrating VMs in from previous underutilized hosts. This type of hosts will be removed from the underutilized host list. Therefore, underutilized host list H_{under} needs to be updated after each VM migration operation. Furthermore, the order of underutilized hosts to migrate VMs does affect the results since the destination hosts status may change during migration. VM selection for underutilized hosts is simpler than that of overutilized hosts. Once PPRGear decides to migrate VMs from an underutilized host, all VMs will migrate out. Then the host will be shut down after migration.

Algorithm 4 presents the algorithm to look for VMs to migrate out from the underutilized hosts. First, if the utilization of a host is lower than the lowest preferred gear utilization, then the host is put in the list H_{under}. Second, PPRGear sorts H_{under} in ascending order on utilization. Last, PPRGear migrates all VMs out from the host with the lowest utilization. Destination hosts utilization must be updated before migrating VMs from next host in H_{under}.

5.3.3. Complexity analysis

Suppose N and M represent the total number of VMs and hosts, respectively, in the cloud and n_i is the number of VMs on the ith host. The time complexity of Algorithm 2, VM Placement Algorithm, is $O(\sum_{i=1}^{M} n_i) = O(N)$ since the Algorithm 2 traverses all VMs on each host. Time complexity of Algorithm 3 is $O(\sum_{i=1}^{M} n_i) + O(\sum_{i=1}^{M} n_i log_2(n_i)) + O(\sum_{i=1}^{M'} n_i) = O(Nlog_2(N))$ where M' is the number of overutilized hosts. Time complexity of Algorithm 4 is $O(\sum_{i=1}^{M} n_i) + O(M'' log_2 M'') + O(M'') = O(Nlog_2(N))$ where M'' denotes the number of underutilized hosts. Note that since the utilization monitoring and sorting jobs can be done locally on hosts, the efficiency of the proposed algorithms can be further improved by distributing the utilization monitoring job to host computers.

Algorithm 4 VM Selection on Underutilized Hosts for Migration

```
input: H_list: all hosts
output: null
H_under ← ϕ
V_migrate ← ϕ
for each h in H_list do
    utilization ← 0
    VM_h ← All Virtual Machines on h
    for each vm in VM_h do
        utilization ← utilization + vm.utilization
    end for
    if utilization < utilization of lowest preferred gear then
        H_under ←
    end if
end for
Sort H_under in ascending order on utilization
for each h in H_under do
    if h is underutilized then
        V_migrate ← all virtual machines on h
    end if
    call Algorithm 2 to allocate V_migrate
end for
```

6. Performance evaluation

To demonstrate the performance and energy-efficiency of PPRGear, we evaluated the performance of PPRGear on four different host models under different workloads in terms of energy consumption, Service-Level Agreement (SLA) violation, shutdown times, and migration times by using CloudSim 3.0.3 [18]. CloudSim 3.0.3 is an event-driven simulator that is used to simulate infrastructures and application services in cloud computing with customizable policies of virtual machine selection, allocation, migration, and provisioning on configurable host models. We implemented PPRGear in CloudSim 3.0.3 with real-world host models.

Based on our simulation results, compared with energy-efficient VM migration baseline algorithms ThrRs, MadMmt, and IqrMc [19], PPRGear reduced energy consumption up to 69.31% with fewer migration and shutdown times. Compared with DVFS, a non-migration power efficient strategy for processors, PPRGear significantly reduced power consumption up to 95.3% under light workloads with little Service-Level Agreement violation. When the workload was extremely high, the SLA violation of PPRGear was almost the same as that of baseline algorithms and DVFS.

6.1. Experimental setup

We simulated a cloud computing center with 800 homogeneous host computers in four different models: Fujitsu Primergy RX1330 M1, Inspur NF5280M4, Dell PowerEdge R820, and IBM NeXtScale nx360 M4. Table 1 presents the specifications of four models among which Inspur NF5280M4 is equipped with the largest capacity of memory (64 GB), while Dell PowerEdge R820 is equipped with the most computing cores (40). The power consumption and performance data of all four host models were collected in the middle of year 2014 by host manufactures using benchmark suite SPEC *power_ssj*2008 [20].

We assume that all virtual machines are configured in the same specification as presented in Table 2. VM's MIPS are mapped from host computers' CPU frequency to quantitatively evaluate CPU utilization [19]. MIPS, processing element amount, memory, and VM size decide the sources requested from the host computer. Bandwidth and VM size decide the VM migration cost as $Migration\_time = VM\_size/Bandwidth$. Notice that although the default MIPS of each VM is 2000, the actual amount of requested MIPS varies depending on the utilization specified in the workload. The actual MIPS required by a VM in a time interval can be estimated as $2000 \times utilization$ and varies over time.

Table 2
Virtual machine configuration in our experiments.

Configuration parameters	Default value
MIPS	2000
Processing element	2
Memory	1 GB
Bandwidth	100 Mbit/s
VM size	2.5 GB
VM CPU utilization	0%–100%
Workload coefficient	0.1x–4x

Note that the performance degradation and service downtime were also considered in our experiments in CloudSim by adding 10% of the CPU utilization. Voorsluys et al. found that both performance degradation and downtime depend on application behaviors [27]. When the cloud is running applications with variable workloads, the average performance degradation and downtime can be approximately estimated as 10% of the CPU utilization [19].

6.2. Workloads

We adopted PlanetLab [28] workload in our study. In order to investigate the performance of PPRGear, all VMs were evenly allocated to 800 host computers at the beginning of our experiments. Then, PPRGear starts to manage VMs by migrating VMs and allocating host computer resource. The PlanetLab workload in our experiments is a list of VM CPU utilization percentage values collected on March 3rd, 2011. Each workload is 24-hour long and the interval of utilization measurement is 300 s.

In addition to using the original PlanetLab workload, we varied the volume of the original workload from 10% ($0.1\times$) to 400% ($4\times$) to investigate its impact on our proposed framework.

6.3. The baseline algorithms

In order to investigate the energy-efficiency improvement and performance impact, PPRGear is compared with three energy-efficient VM allocation and selection algorithms: IqrMc, MadMmt, and ThrRs [19] which use different policies to select hosts and VMs for migration. Each of those three algorithms has two core steps. First step is deciding whether the current host is overutilized based on utilization threshold. The utilization threshold is either configured as a constant value before experiments or is generated dynamically based on historical utilization information. Second, the algorithm selects an appropriate VM from a host for migration if the host is currently overutilized. VM selection algorithm will repeat until the host is not overutilized.

- Static Threshold VM allocation policy and Random Selection VM selection policy, or ThrRs, uses a static utilization threshold instead of generating one in real time. In our experiments, utilization threshold is constantly set to 0.8. When a host is overutilized, ThrRs randomly selects a VM for migration regardless of the VM's performance impact on the host. ThrRs is a primitive energy-efficient VM allocation and selection algorithm.
- Median Absolute Deviation VM allocation policy and Minimum Migration Time VM selection policy, or MadMmt, uses a robust statistic Median Absolute Deviation to calculate historical data's median for utilization threshold. According

to this dynamic utilization threshold, if a host is overutilized, MadMmt selects a VM that takes the least time for migration.

- Interquartile Range VM allocation policy and Maximum Correlation VM selection policy, or IqrMc, uses another robust statistic Interquartile Range to analyze historical utilization. IqrMc selects the most correlated VM to CPU for migration [29].

Note that we also compared PPRGear with a popular non-VM-Migration energy-efficient algorithm DVFS (Dynamic Voltage and Frequency Scaling) [30] for SLA violations. Since DVFS is designed to adapt to CPU utilization, it does not delay execution. Hence, DVFS was used as the performance baseline in our experiments for SLA violations.

6.4. Preferred number of gears and best gear

In Section 3, we present exact and efficient calculations for PPRGear. Although exact calculation is the most accurate way to apply PPRGear, it requires a large amount of work to collect performance and power data and conclude them as a function corresponding to utilization. Therefore, we use efficient calculation in our experiments. According to the experimental results, the efficient calculation based PPRGear performs effectively on both energy and SLA. And it is only required to collect performance and power data for 11 utilization levels on each type of host computer.

The preferred gears are the top n highest PPR gears. Number of preferred gears is used to judge whether a host is overutilized or underutilized. If current host utilization is higher than the highest utilization of preferred gears, then the current host is overutilized. If the current host utilization is lower than the lowest utilization of preferred gears, then the current host is underutilized.

The best gear is the gear with the highest PPR value. In other words, the host computer achieves the most power-efficient utilization level while working at the best gear. Best gear is used when migrating and allocating VMs. When migrating VMs from other hosts or allocating newly created VMs, PPRGear attempts to make the targeted host computing work as close to the best gear utilization as possible.

Note that there is always one best gear in one host computer and the best gear is also one of the preferred gears. The number of preferred gears is set by cloud administrators and has significant performance impact on PPRGear. When the number of preferred gears is set small, the cloud works energy-efficiently but it could also lead to highly frequent VM migrations and host shutdowns which will be harmful to host computers' reliability. If the number of preferred gears is too large, PPRGear does not work energy-efficiently since there are too many preferred gears. Therefore, overutilized hosts and underutilized hosts will be very rare and few migrations will be triggered in PPRGear.

6.5. Impact of workloads

6.5.1. Impact of workloads on energy consumption

Fig. 4 presents the energy consumption of various host types under different workloads. Host models of Figs. 4(a), 4(b), 4(c), 4(d) are Fujitsu Primergy RX1330 M1, Inspur NF5280M4, Dell PowerEdge R820, and IBM NeXtScale nx360 M4, respectively. The corresponding performance-to-power ratios and average active power are presented in Table 3. Server Side Java Operations (*ssj_ops*) are presented in Table 4. There are various metrics that could be used to assess a host computer's performance. Since both Hadoop and Spark depend on Java Virtual Machine, we use Server Side Java Operations to evaluate a host computer's computing performance in this paper.

In our experiments, a host's utilization is calculated based on all VMs allocated on the host. VM utilization is stated in workload trace file with 300 s interval between each measurement. In order to test PPRGear under different workloads, the original VM utilization is also manipulated by multiplying a control factor between 0.1 to 4 in experiments. In all four subfigures of Fig. 4, energy consumption increases as workload increases.

According to Figs. 4(a), 4(b), 4(c), and 4(d), the two PPRGear algorithms, with number of preferred gears 1 and 2, consume significantly less energy. The figures also indicate that PPRGear's energy conservation rate is higher when workload is lower. When workload is extremely high (4x), the total energy consumption of PPRGear is very close to the other algorithms. When workload increases, the energy conservation rate decreases in all host models.

6.5.2. Impact of workloads on SLA

Service-Level Agreement (SLA) describes the Quality of Service (QoS) of cloud services between cloud service providers and customers. In other words, SLA is the promise made by cloud service providers about the computing resource offered to customers. If SLA is violated, cloud service providers usually refund some money back to customers. Our method was proposed as a live migration strategy and therefore we use SLA violation to quantitatively measure the negative impact on performance.

In our experiments, we adopted CloudSim's default SLA [18]. We use *overall SLA violation rate*, or $SLAVR_{overall}$, to evaluate the quality of the services provided at the cloud end, which can be calculated using Eq. (5). In Eq. (5), m_i^r means the MIPS requested by Virtual Machine i during the whole process while m_i^a denotes the total MIPS that are actually allocated to Virtual Machine i, assuming that there are n virtual machines in total. Eq. (6) indicates that the violation rate is positive when m_i^r is greater than m_i^a since the SLA is not met due to lack of available MIPS on the host computer; and the violation rate is zero as long as m_i^a is greater or equal to m_i^r.

$$SLAVR_{overall} = \frac{\sum_{i=1}^{n} d(i)}{\sum_{i=1}^{n} m_i^r} \tag{5}$$

where

$$d(i) = \begin{cases} m_i^r - m_i^a & \text{if } m_i^r > m_i^a \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

As shown in Fig. 5, when the workload exceeded the capacity of the cloud computing system (i.e., the workload was higher than 1x), the performance of all the investigated methods started to deteriorate accordingly in terms of SLA violation.

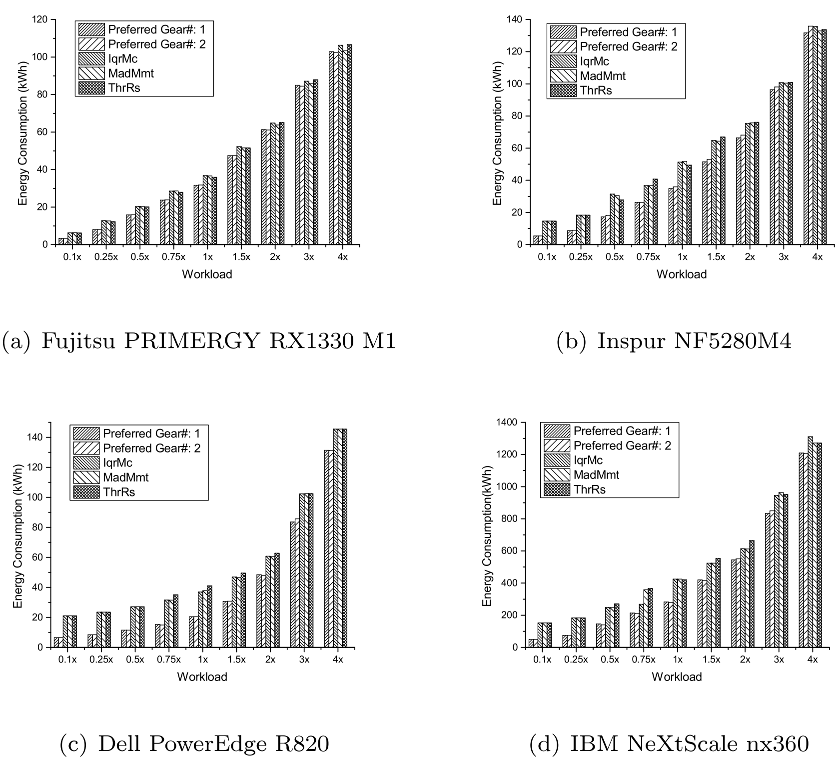
According to Figs. 5(a), 5(b), 5(c), and 5(d), the SLAs of PPRGear were either very close or almost the same as that of ThrRs, MadMmt, IqrMc, and DVFS under different workloads and number of preferred gears. In other words, PPRGear does not compromise system performance or capacity of the cloud for higher energy efficiency.

6.5.3. Impact of workloads on migration times

When workload gets heavier, a smaller number of preferred gears may cause more migration and shutdown times. Fig. 7 presents the impact of workloads on migration for all four types of hosts. Fig. 7(a) shows that PPRGear (the numbers of preferred gears is 1 and 2, respectively) causes fewer migration times compared with other baseline algorithms when workload is less than or equal to 1.5x. Yet, when the workload is higher, the migration times of PPRGear will surpass baseline algorithms in Fig. 7(a) (Fujitsu Primergy RX1330 M1). However, Fig. 7(c) (Dell PowerEdge R820) shows that the number of migration is always less when using PPRGear. The reason is that Dell PowerEdge

Table 3
Performance-to-power ratio of gears [23–26].

Gear level:	Gear0	Gear1	Gear2	Gear3	Gear4	Gear5	Gear6	Gear7	Gear8	Gear9	Gear10
Utilization:	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
Host model	Performance-to-power ratio values										
Fujitsu Primergy RX1330 M1	0	2425	4281	5857	6991	7821	8467	8540	8410	8231	8041
Inspur NF5280M4	0	3796	6295	8063	9385	10 590	11 519	11 536	11 570	11 198	10 441
Dell PowerEdge R820	0	2599	4538	5995	7130	8050	8705	9194	9533	10 013	9372
IBM NeXtScale nx360 M4	0	2589	4445	5858	6965	7849	8477	8952	9070	9012	8731
Host model	Average active power (Watt)										
Fujitsu Primergy RX1330 M1	13.8	20.8	23.9	26.3	29.1	32.6	36.2	42.0	48.6	55.9	63.7
Inspur NF5280M4	44.4	83.3	101	118	135	146	161	190	218	255	301
Dell PowerEdge R820	71.8	135	156	176	198	219	243	269	297	318	374
IBM NeXtScale nx360 M4	497	814	947	1079	1211	1344	1493	1648	1863	2108	2414



(a) Fujitsu PRIMERGY RX1330 M1
(b) Inspur NF5280M4
(c) Dell PowerEdge R820
(d) IBM NeXtScale nx360

Fig. 4. Energy consumption of various host types under varying workloads.

R820 has 40 cores which is 10 times more than Fujitsu Primergy RX1330 M1's 4 cores. Therefore, the higher capacity of the host computer, the heavier workload it is able to handle with fewer migrations.

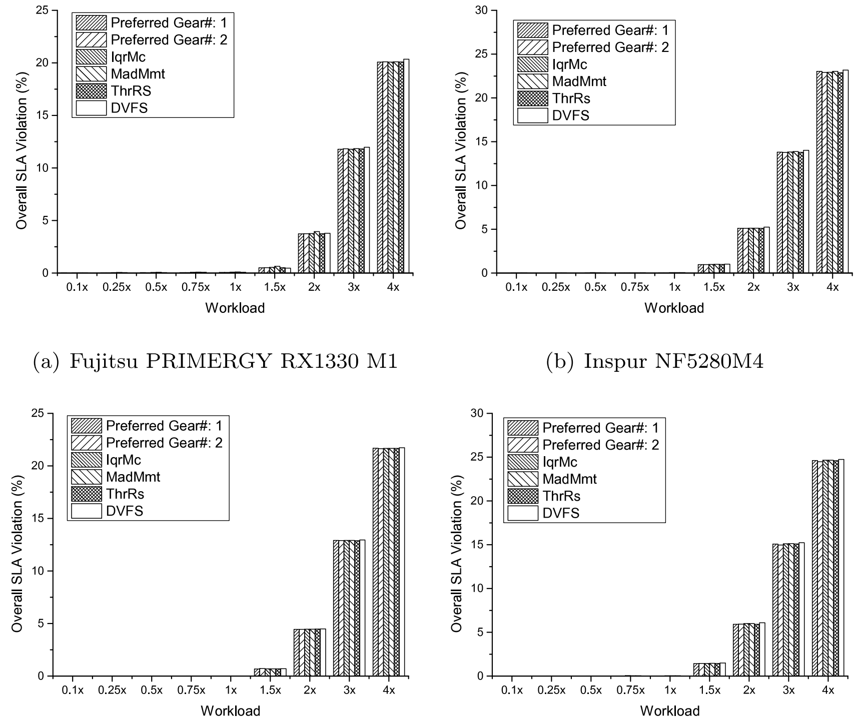
It may seem at the first glance that having a higher number of the preferred gears will result in less migration. However, it is not always the case. The actual number of migration occurrences can be also affected by the workload. For example, given a very high workload, having fewer preferred gears may lead to more overutilized hosts. That will make it more difficult to locate an underutilized host as the target machine to receive the "migrate-out" VMs. If an available host cannot be found, the migration does not occur actually. In Fig. 7, under a high workload, setting the number of the preferred gears to 1 resulted in most of hosts being overutilized. Consequently, fewer migrations can be made, due to the failure of finding a target host for a migration attempt.

6.5.4. Impact of workloads on shutdown times

Fig. 8 presents the numbers of shutdowns for all four hosts. In our experiments, when a host is working at utilization level 0%, we say it is shutdown. Similarly to the number of migration, the fewer cores of a host computer, the more sensitive the host is to workloads. In most cases, PPRGear performs with fewer or similar number of shutdowns.

6.6. Impact of the number of preferred gears

Fig. 6 show the impact of number of preferred gears on energy consumption, migration and shutdown times under various workloads. Figs. 6(a), 6(d), and 6(g) show the impact of the number of preferred gears on energy cost under workloads of 0.5x, 1x, and 2x the original workload, respectively. Energy conservation rate is significant when the number of preferred gears is small. When the number of preferred gears is large enough, the energy consumption will be the same as that of DVFS. The reason is that when the number of preferred gears is set to be large, the host computer will be mostly working at the utilization levels of preferred gears. The migration times will reduce, and so will the effectiveness of PPRGear. PRRGear also uses DVFS on individual



(a) Fujitsu PRIMERGY RX1330 M1

(b) Inspur NF5280M4

(c) Dell PowerEdge R820

(d) IBM NeXtScale nx360

Fig. 5. SLA violation rate.

host computers. Therefore, if the number of preferred gears is large enough, PPRGear will work exactly the same as DVFS since DVFS does not migrate virtual machines.

Figs. 6(b), 6(e), and 6(h) show the impact of the number of preferred gears on the number of migrations under workloads of 0.5x, 1x, and 2x the original workload, respectively. They indicate that when the number of preferred gears increases, the number of migrations reduces correspondingly in most of the cases because the host computer allows a larger number of preferred gears. But the workload pattern may also affect migration, so occasionally, the number of migration may be higher when the number of preferred gears is higher (In Fig. 6(b), the number of preferred gears is 2).

Figs. 6(c), 6(f), and 6(i) show the impact of the number of preferred gears on the number of migrations under workloads of 0.5x, 1x, and 2x the original workload, respectively. The number of shutdowns will reduce as the number of preferred gears increases.

One interesting observation is that with the increase in the number of preferred gears, Fujitsu Primergy RX1330 M1 kept significant energy conservation rates until the number of preferred gears was 4, and Inspur NF5280M4 and IBM NeXtScale nx360 M4 kept a significant energy conservation rates until the number of preferred gears was 3. However, Dell PowerEdge R820 only kept good energy conservation rates until the number of preferred gears was 2. This observation indicates that the effective number of preferred gears depends on the number of cores that each host has. The more cores a host has, the more VMs a host can execute.

Fig. 6(e) shows the impact of the number of preferred gears on migration under the original workload 1x. According to Fig. 6(e), the more cores that a host has, the fewer migrations will occur in the cloud due to the greater computing capacity. Fig. 6(f) reveals the impact of the number of preferred gears on the number of shutdowns under the original workload 1x. The impact fades

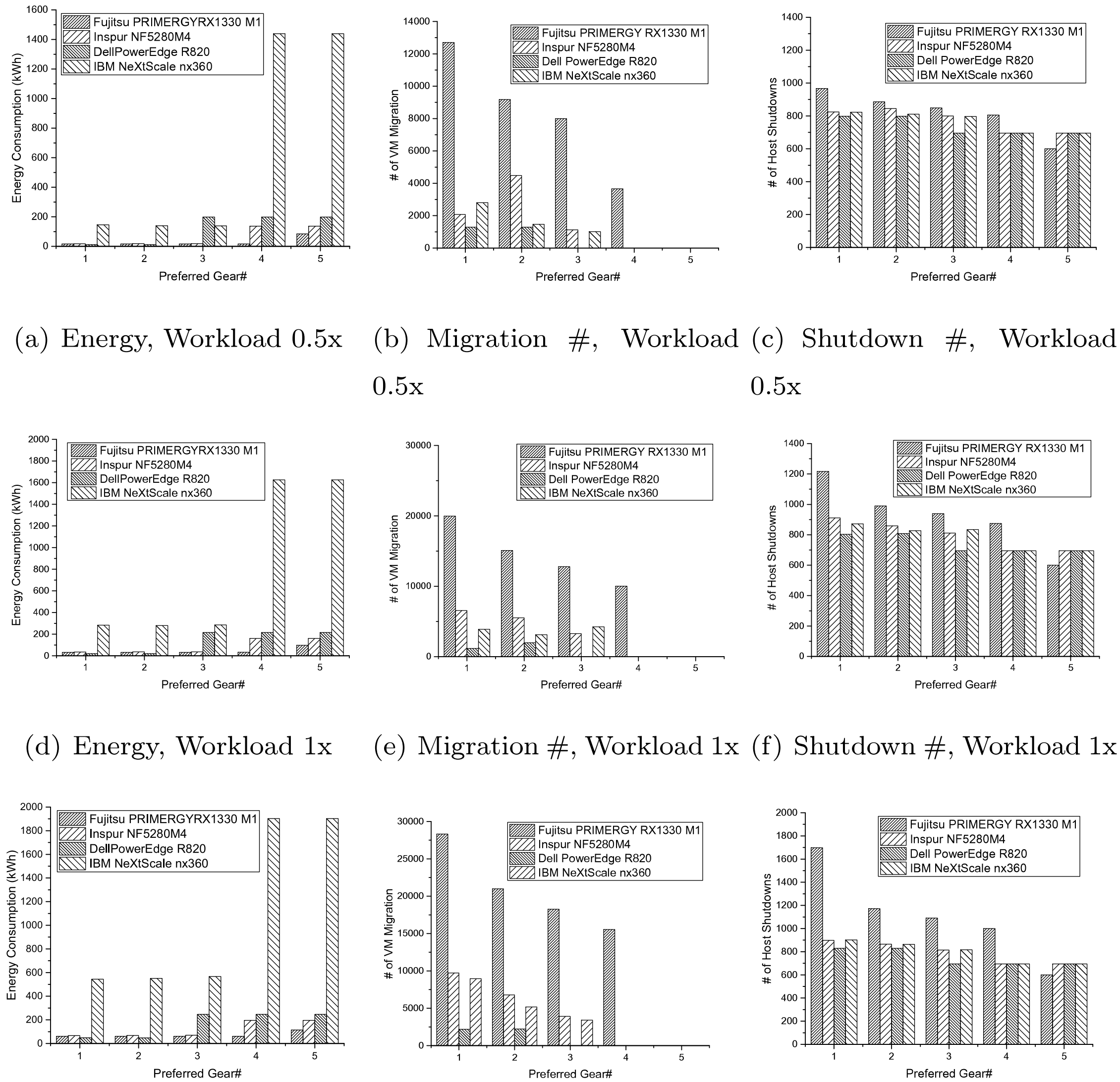
Table 4
Computing performance of host models in ssj_ops [23–26].

Host model	Fujitsu Primergy RX1330 M1	Inspur NF5280M4	Dell PowerEdge R820	IBM NeXtScale nx360 M4
Gear0	0	0	0	0
Gear1	50,359	316,136	350,153	2,107,446
Gear2	102,370	636,228	707,185	4,209,415
Gear3	154,189	950,734	1,056,794	6,320,782
Gear4	203,541	1,268,612	1,411,080	8,434,615
Gear5	254,724	1,546,604	1,765,011	10,549,056
Gear6	306,424	1,857,279	2,118,458	12,656,161
Gear7	358,373	2,193,932	2,472,450	14,752,896
Gear8	408,824	2,525,204	2,827,529	16,897,410
Gear9	460,101	2,852,241	3,183,264	18,997,296
Gear10	512,425	3,145,159	3,508,442	21,076,634

when the number of preferred gears gets larger according to Fig. 6(f).

7. Related work

An important aspect of energy-efficiency clouds is accomplishing more jobs with less power. In energy-efficient clouds, power consumption is measured at computing node level since different components, like processors, memory, and second-level storage [31], have different power consumption models. It is indirect and difficult to measure power consumption for individual components in order to evaluate overall power consumption. According to recent studies, although DVFS demonstrates that the relationship of CPU power consumption and utilization is exponential, the relationship of overall power consumption and CPU utilization is nearly linear [21,22,32]. Based on this conclusion, a lot of research about energy-efficient clouds was conducted on Virtual Machine level.



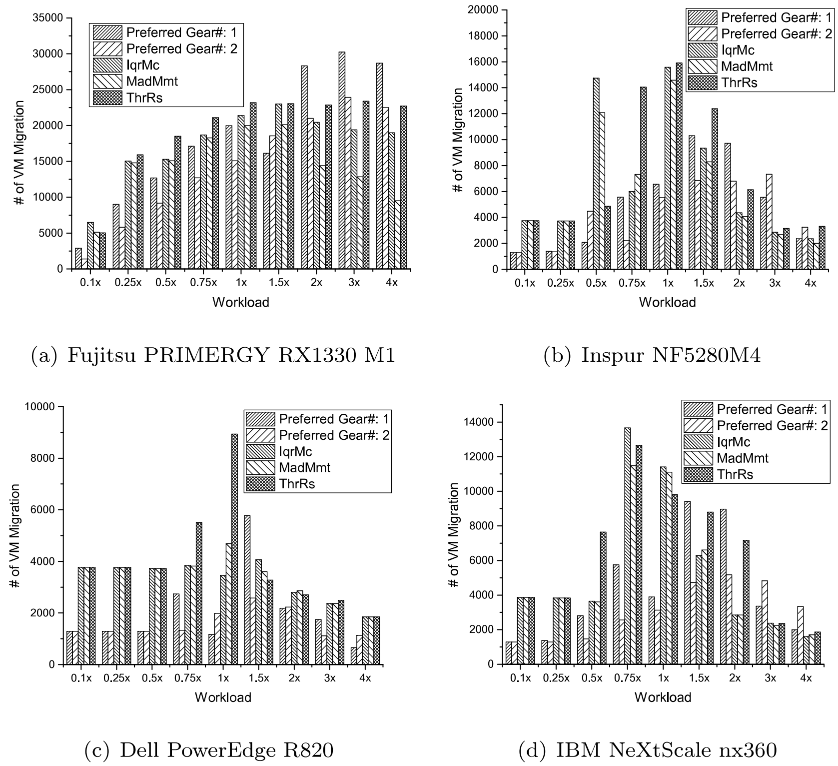
(a) Energy, Workload 0.5x (b) Migration #, Workload 0.5x (c) Shutdown #, Workload 0.5x

(d) Energy, Workload 1x (e) Migration #, Workload 1x (f) Shutdown #, Workload 1x

(g) Energy, Workload 2x (h) Migration #, Workload 2x (i) Shutdown #, Workload 2x

Fig. 6. Impact of number of preferred gears on energy consumption, migration and shutdown times under the 0.5x, 1x, and 2x of the original workload.

Virtual Machine allocation, migration, and consolidation have been explored for both performance [40] and energy efficiency based on different strategies. Power-aware VM consolidation saves significant amount of energy in clouds but may cause noteworthy performance degradation. Beloglazov et al. analyzed the energy-performance tradeoff for energy and performance efficient dynamic VM consolidation [19]. Consolidation can be triggered by conditions based on different policies. Agrawal et al. proposed pSciMapper, a power-aware consolidation framework based on the characteristics of scientific workloads [11]. Xu et al. designed algorithms to consolidate workload with minimizing both energy consumption and network workloads [12]. Assuming all VMs have been placed on physical hosts, VMs will be reassigned again with both energy consumption and network overhead concerns. Kim et al. proposed a strategy of VM placement based on correlation information of core utilization [33]. Kansal et al. proposed an energy-aware virtual machine migration technique based on the Firefly algorithm [34]. Farahnakian et al. proposed an Ant Colony System-based VM Consolidation approach that finds a near-optimal solution based on a specified objective function [35]. The strategy proposed by Mosa et al. dynamically assigns VMs to Physical Machines to co-optimizes energy consumption and service level agreement (SLA) violations while the primary goal of PPRGear is to optimize the energy consumption [36]. Portaluri et al. studied VM placement strategies to take advantage of Software Defined Network to reduce power consumption in cloud computing data centers [37]. Zhao et al. proposed a power-aware and performance-guaranteed VM placement strategy that applies Ant Colony Optimization [41]. In order to reduce both energy consumption and network overhead, Xu et al. applied VM packing algorithms and interesting trade-offs have been found between energy consumption and network overhead [12]. Verma et al. presented pMapper which places applications in virtualized systems with power and migration cost awareness [38]. Ghribi et al. explored VM placement problem and used exact algorithms for both VM placement and workload consolidation to reduce energy consumption [42]. When workload is extremely low (utilization<10%), it is hard to conserve energy



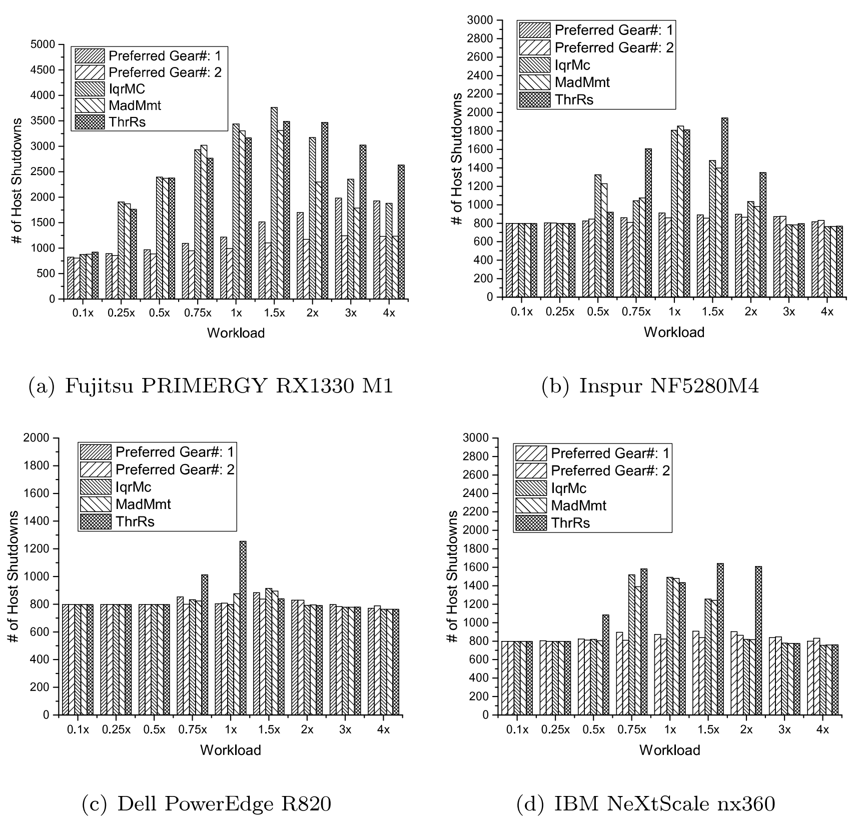
(a) Fujitsu PRIMERGY RX1330 M1

(b) Inspur NF5280M4

(c) Dell PowerEdge R820

(d) IBM NeXtScale nx360

Fig. 7. Migration #.



(a) Fujitsu PRIMERGY RX1330 M1

(b) Inspur NF5280M4

(c) Dell PowerEdge R820

(d) IBM NeXtScale nx360

Fig. 8. Shutdown #.

Table 5
Related work comparison table.

Work name	PPR aware	Online algorithm	VM placement	Live migration	Notes
PPRGear	Y	Y	Y	Y	The presented work
ThrRs [19]	N	Y	Y	Y	Randomly select VM for migration
MadMmt [19]	N	Y	Y	Y	Needs to collect historical data
IqrMc [19]	N	Y	Y	Y	Needs to collect historical data
DVFS	N	Y	N	N	Local algorithm
pSciMapper [11]	N	N	N	N	Designed for scientific work flow (DAG-based computation),56% energy efficiency but 15% slow down. Offline analysis is required.
Correlation-aware VM allocation [33]	N	Y	Y	N	Only about the initial VM placement, need to collect correlation data first
Firefly [34]	N	N	N	Y	Need to know workload
ACS-VMC [35]	N	Y	Y	Y	VM consolidation based on Ant Colony Optimization. Complex mathematic model.
Utility functions [36]	N	Y	Y	Y	Reduce some energy cost but mainly SLA violations
PPVMP [37]	N	Y	Y	N	Static VM placement
Pmapper [38]	N	Y	N	N	Application level algorithm instead of VM level
Dynamic resource allocation with prediction [39]	N	Y	Y	Y	Focus on workload balance but energy conservation

since there is not much room to further reduce utilization to conserve energy (DVFS strategy). Hence, Wang et al. proposed request batching [14] to group received requests in batches. The requests are served in batches and hosts are shut down between batches. Xiao et al. used VM to allocate system resource based on skewness to conserve energy conservation [39].

As an extension of [43], this paper presents exact and approximate calculation of the best gear and extensive experimental simulations under different settings. Table 5 summarizes some of the above-mentioned related work. Note that we chose IqrMc, MadMmt, and ThrRs as comparison baselines because they are the most comparable works to PPRGear. PPRGear takes into account the performance-to-power ratio as a key factor in order to make VM placement and migration decisions without negative performance impact.

Compared with all the aforementioned works, our strategy does not require any extra information for scheduling. Specifically, it does not need to know the correlated information of core utilization or the workload patterns of computing nodes over the underlying networks. In other words, our strategy only relies on host utilization to conserve overall energy consumption. As a result, our strategy can be implemented on the VM scheduler level directly, leading to lower energy cost.

8. Conclusion

Energy consumption has become a big concern in the last decade since cloud data centers consumed significant power and generated giant power bills. In a cloud computing environment, computing resources are allocated to virtual machines that are generated for customers. The placement and migration of virtual machines have significant impact on both performance and energy cost. In this paper, we presented PPRGear, an energy-efficient virtual machine allocation and migration framework for energy-efficient clouds. To the best of our knowledge, our work is the first to leverage the performance-to-power ratio of computing nodes in virtual machine allocation and migration for energy-efficient cloud solution. By achieving the optimal balance between host utilization and energy consumption, PPRGear is able to guarantee that host computers run at the most power-efficient levels, i.e., the levels with highest Performance-to-Power ratios, so that the energy consumption can be tremendously reduced without much sacrifice of the computing performance. Our extensive experiments with real world traces show that compared with the state-of-the-art approaches, PPRGear is able to reduce up to 69.31% of the energy consumption compared with IqrMc, MadMmt, and ThrRs, as well as leading to fewer VM migrations and shutdowns.

Conflict of interest

None.

Declaration of competing interest

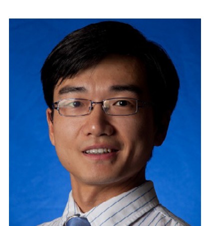
The authors declare that there is no conflict of interest regarding the publication of this manuscript.

References

[1] J. Glanz, The cloud factories Power, pollution and the internet, N.Y. Times (2012) URL http://www.nytimes.com/2012/09/23/technology/data-centers-waste-vast-amounts-of-energy-belying-industry-image.html?pagewanted=all&_r=0.
[2] P. Thibodeau, Data centers are the new polluters, Computerworld (2014) URL http://www.computerworld.com/article/2598562/data-center/data-centers-are-the-new-polluters.html.
[3] K. Ebrahimi, G.F. Jones, A.S. Fleischer, A review of data center cooling technology, operating conditions and the corresponding low-grade waste heat recovery opportunities, Renew. Sustain. Energy Rev. 31 (0) (2014) 622–638, http://dx.doi.org/10.1016/j.rser.2013.12.007, URL http://www.sciencedirect.com/science/article/pii/S1364032113008216.
[4] P.D. Forge, America's data centers consuming and wasting growing amounts of energy, Issue Paper, Natural Resources Defense Council. URL http://www.nrdc.org/energy/data-center-efficiency-assessment.asp.
[5] E. Gelenbe, R. Lent, Optimising server energy consumption and response time, Theor. Appl. Inform. 24 (4) (2012) 257–270.
[6] Z. Guo, Z. Duan, Y. Xu, H.J. Chao, Cutting the electricity cost of distributed datacenters through smart workload dispatching, IEEE Commun. Lett. 17 (12) (2013) 2384–2387, http://dx.doi.org/10.1109/LCOMM.2013.102213.131831.
[7] Z. Guo, Z. Duan, Y. Xu, H.J. Chao, Jet: Electricity cost-aware dynamic workload management in geographically distributed datacenters, Comput. Commun. 50 (2014) 162–174, http://dx.doi.org/10.1016/j.comcom.2014.02.011, URL http://www.sciencedirect.com/science/article/pii/S0140366414000644.

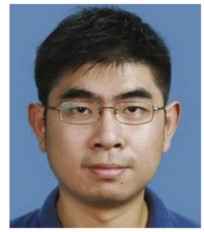
[8] Z. Guo, S. Hui, Y. Xu, H.J. Chao, Dynamic flow scheduling for power-efficient data center networks, in: 2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS), 2016, pp. 1–10, http://dx.doi.org/10.1109/IWQoS.2016.7590399.
[9] A. Berl, E. Gelenbe, M.D. Girolamo, G. Giuliani, H. de Meer, D.M. Quan, K. Pentikousis, Energy-efficient cloud computing, Comput. J. 53 (7) (2010) 1045–1051, http://dx.doi.org/10.1093/comjnl/bxp080.
[10] X. Zhu, L. Yang, H. Chen, J. Wang, S. Yin, X. Liu, Real-time tasks oriented energy-aware scheduling in virtualized clouds, IEEE Trans. Cloud Comput. 2 (2) (2014) 168–180, http://dx.doi.org/10.1109/TCC.2014.2310452.
[11] Q. Zhu, J. Zhu, G. Agrawal, Power-aware consolidation of scientific workflows in virtualized environments, in: High Performance Computing, Networking, Storage and Analysis (SC), 2010 International Conference for, 2010, pp. 1–12, http://dx.doi.org/10.1109/SC.2010.43.
[12] N. Tziritas, C.-Z. Xu, T. Loukopoulos, S. Khan, Z. Yu, Application-aware workload consolidation to minimize both energy consumption and network load in cloud environments, in: Parallel Processing (ICPP), 42nd International Conference on, 2013, http://dx.doi.org/10.1109/ICPP.2013.54.
[13] S. Yeo, M.M. Hossain, J.-C. Huang, H.-H.S. Lee, Atac: Ambient temperature-aware Capping for power efficient datacenters, in: Proceedings of the ACM Symposium on Cloud Computing, in: SOCC'14, ACM, New York, NY, USA, 2014, pp. 17:1–17:14, http://dx.doi.org/10.1145/2670979.2670996, URL http://doi.acm.org/10.1145/2670979.2670996.
[14] Y. Wang, X. Wang, Virtual batching: Request batching for server energy conservation in virtualized data centers, IEEE Trans. Parallel Distrib. Syst. 24 (8) (2013) 1695–1705, http://dx.doi.org/10.1109/TPDS.2012.237.
[15] E. Gelenbe, R. Lent, M. Douratsos, Choosing a local or remote cloud, in: Network Cloud Computing and Applications (NCCA), 2012 Second Symposium on, 2012, pp. 25–30, http://dx.doi.org/10.1109/NCCA.2012.16.
[16] E. Gelenbe, R. Lent, Energy-qos trade-offs in mobile service selection, Future Internet 5 (2) (2013) 128–139, http://dx.doi.org/10.3390/fi5020128.
[17] E. Gelenbe, R. Lent, Trade-offs between energy and quality of service, in: Sustainable Internet and ICT for Sustainability, SustainIT 2012, 4-5 October, 2012, Pisa, Italy, Sponsored By the IFIP TC6 WG 6.3 "Performance of Communication Systems", 2012, pp. 1–5, URL http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6388014.
[18] R.N. Calheiros, R. Ranjan, A. Beloglazov, A.F. De Rose, R. Buyya, Cloudsim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms, Softw. Pract. Exper. (2011) http://dx.doi.org/10.1002/spe.995.
[19] A. Beloglazov, R. Buyya, Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers, Concurr. Comput.: Pract. Exper. 24 (13) (2012) 1397–1420, http://dx.doi.org/10.1002/cpe.1867.
[20] SPEC Power, https://www.spec.org/power/.
[21] X. Fan, W.-D. Weber, L.A. Barroso, Power provisioning for a warehouse-sized computer, in: Proceedings of the 34th Annual International Symposium on Computer Architecture, in: ISCA'07, ACM, New York, NY, USA, 2007, pp. 13–23, http://dx.doi.org/10.1145/1250662.1250665, URL http://doi.acm.org/10.1145/1250662.1250665.
[22] C. Mobius, W. Dargie, A. Schill, Power consumption estimation models for processors, virtual machines, and servers, IEEE Trans. Parallel Distrib. Syst. 25 (6) (2014) 1600–1614, http://dx.doi.org/10.1109/TPDS.2013.183.
[23] SPEC, Fujitsu FUJITSU Server PRIMERGY RX1330 M1, https://www.spec.org/power_ssj2008/results/res2014q3\/power_ssj2008-20140804-00662.html.
[24] SPEC, Inspur Corporation NF5280M4, https://www.spec.org/power_ssj2008/results/res2014q4/power_ssj2008-20140905-00673.html.
[25] SPEC, Dell Inc. PowerEdge R820 (Intel Xeon E5-4650 v2 2.40 GHz), https://www.spec.org/power_ssj2008/results/res2014q2/power_ssj2008-20140401-00654.html.
[26] SPEC, IBM Corporation IBM NeXtScale nx360 M4 (Intel Xeon E5-2660 v2), https://www.spec.org/power_ssj2008/results/res2014q2/power_ssj2008-20140421-00657.html.
[27] W. Voorsluys, J. Broberg, S. Venugopal, R. Buyya, Cost of virtual machine live migration in clouds: A performance evaluation, in: Proceedings of the 1st International Conference on Cloud Computing, in: CloudCom'09, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 254–265, http://dx.doi.org/10.1007/978-3-642-10665-1_23.
[28] K. Park, V.S. Pai, Comon: A mostly-scalable monitoring system for planetlab, SIGOPS Oper. Syst. Rev. 40 (1) (2006) 65–74, http://dx.doi.org/10.1145/1113361.1113374, URL http://doi.acm.org.proxy-wcupa.klnpa.org/10.1145/1113361.1113374.
[29] A. Verma, G. Dasgupta, T.K. Nayak, P. De, R. Kothari, Server workload analysis for power minimization using consolidation, in: Proceedings of the 2009 Conference on USENIX Annual Technical Conference, in: USENIX'09, USENIX Association, Berkeley, CA, USA, 2009, p. 28, URL http://dl.acm.org/citation.cfm?id=1855807.1855835.
[30] T. Gurout, T. Monteil, G.D. Costa, R.N. Calheiros, R. Buyya, M. Alexandru, Energy-aware simulation with {DVFS}, Simul. Model. Pract. Theory 39 (2013) 76–91, http://dx.doi.org/10.1016/j.simpat.2013.04.007, URL http://www.sciencedirect.com/science/article/pii/S1569190X13000786.
[31] S. Yin, X. Ruan, A. Manzanares, X. Qin, K. Li, MINT: A reliability modeling framework for energy-efficient parallel disk systems, IEEE Trans. Dependable Secure Comput. 11 (4) (2014) 345–360, http://dx.doi.org/10.1109/TDSC.2013.47.
[32] D. Kusic, J. Kephart, J. Hanson, N. Kandasamy, G. Jiang, Power and performance management of virtualized computing environments via lookahead control, in: Autonomic Computing, 2008. ICAC '08. International Conference on, 2008, pp. 3–12, http://dx.doi.org/10.1109/ICAC.2008.31.
[33] J. Kim, M. Ruggiero, D. Atienza, M. Lederberger, Correlation-aware virtual machine allocation for energy-efficient datacenters, in: Design, Automation Test in Europe Conference Exhibition (DATE), 2013, 2013, pp. 1345–1350, http://dx.doi.org/10.7873/DATE.2013.277.
[34] N.J. Kansal, I. Chana, Energy-aware virtual machine migration for cloud computing - a firefly optimization approach, J. Grid Comput. 14 (2) (2016) 327–345, http://dx.doi.org/10.1007/s10723-016-9364-0.
[35] F. Farahnakian, A. Ashraf, T. Pahikkala, P. Liljeberg, J. Plosila, I. Porres, H. Tenhunen, Using ant colony system to consolidate VMs for green cloud computing, IEEE Trans. Serv. Comput. 8 (2) (2015) 187–198, doi.ieeecomputersociety.org/10.1109/TSC.2014.2382555.
[36] A. Mosa, N.W. Paton, Optimizing virtual machine placement for energy and SLA in clouds using utility functions, J. Cloud Comput. 5 (1) (2016) 67:1–67:17, http://dx.doi.org/10.1186/s13677-016-0067-7.
[37] G. Portaluri, D. Adami, A. Gabbrielli, S. Giordano, M. Pagano, Power consumption-aware virtual machine placement in cloud data center, IEEE Trans. Green Commun. Netw. 1 (4) (2017) 541–550, http://dx.doi.org/10.1109/TGCN.2017.2725488.
[38] A. Verma, P. Ahuja, A. Neogi, ?pmapper: Power and migration cost aware application placement in virtualized systems, in: Proceedings of the 9th ACM/IFIP/USENIX International Conference on Middleware, in: Middleware'08, Springer-Verlag New York, Inc., New York, NY, USA, 2008, pp. 243–264, URL http://dl.acm.org/citation.cfm?id=1496950.1496966.
[39] Z. Xiao, W. Song, Q. Chen, Dynamic resource allocation using virtual machines for cloud computing environment, IEEE Trans. Parallel Distrib. Syst. 24 (6) (2013) 1107–1117, http://dx.doi.org/10.1109/TPDS.2012.283.
[40] S. Bazarbayev, M. Hiltunen, K. Joshi, W. Sanders, R. Schlichting, Content-based scheduling of virtual machines (vms) in the cloud, in: Distributed Computing Systems (ICDCS), 2013 IEEE 33rd International Conference on, 2013, pp. 93–101, http://dx.doi.org/10.1109/ICDCS.2013.15.
[41] H. Zhao, J. Wang, F. Liu, Q. Wang, W. Zhang, Q. Zheng, Power-aware and performance-guaranteed virtual machine placement in the cloud, IEEE Trans. Parallel Distrib. Syst. 29 (6) (2018) 1385–1400, http://dx.doi.org/10.1109/TPDS.2018.2794369.
[42] C. Ghribi, M. Hadji, D. Zeghlache, Energy efficient VM scheduling for cloud data centers: Exact allocation and migration algorithms, in: Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on, 2013, pp. 671–678, http://dx.doi.org/10.1109/CCGrid.2013.89.
[43] X. Ruan, H. Chen, Performance-to-power ratio aware virtual machine (VM) allocation in energy-efficient clouds, in: Proceedings of the 2015 IEEE International Conference on Cluster Computing, in: CLUSTER'15, IEEE Computer Society, Washington, DC, USA, 2015, pp. 264–273, http://dx.doi.org/10.1109/CLUSTER.2015.46.

Xiaojun Ruan is an Assistant Professor with the Department of Computer Science at California State University, East Bay. He was an Assistant/Associate Professor of Computer Science Department at West Chester University Pennsylvania from 2011 to 2017. He received his B.E. degree in Computer Science and Technology from Shandong University, Jinan, China in 2005, and Ph.D. degree in Computer Science from Auburn University in 2011. His research interests include cloud computing, data science, storage systems, energy-efficient computing, distributed and parallel computing, Artificial Intelligence, and computer security.



Haiquan Chen is an Assistant Professor with the Department of Computer Science at California State University, Sacramento. He was an Assistant/Associate Professor with the Department of Computer Science at Valdosta State University, Valdosta, GA from 2011 to 2017. He obtained his Ph.D. degree in Computer Science from Auburn University in 2011. He received his M.E. degree and B.E. degree in Computer Science from Xian Jiaotong University (XJTU) in China in 2006 and 2003, respectively. His research interests embrace database and big data management, with emphasis on data mining, machine learning, location-based services, and social network analysis. He is a member of the ACM and the ACM SIGMOD.

Yun Tian is an Assistant Professor with the Department of Computer Science at California State University, Fullerton. She obtained her Ph.D. degree in Computer Science and Software Engineering and M.S.E. degree in Software Engineering from Auburn university in 2013 and 2011, respectively. She received her B.E. degree in Computer Science and Technology from Northwest University, Xi'an, China in 2006. Her current research interests lie in the areas of computer and network security, distributed computing, and parallel and high performance computing. In addition, she is interested in modeling, simulation, deep learning, data mining and other big data related fields.



Shu Yin received his B.S. in Communication Engineering from Wuhan University of Technology (WUT) in 2006 and his M.S. degree in Signal and Information Processing from WUT in 2008. He received the Ph.D. degree in Computer Science from Auburn University in 2012. He was an Associate Professor in the College of Computer Science and Electronics Engineering at Hunan University, Changsha, China. Currently, he is an Assistant Professor of School of Information Science and Technology, ShanghaiTech University, China. During July–December 2011 he has worked as an intern at Los Alamos National Laboratory. His research interests include storage systems, reliability modeling, fault tolerance, energy efficient computing, high performance computing, and wireless communications.