# Virtual Resource Consolidation for Green Computing Based on Virtual Cluster Live Migration

Xiaohui Wei, Yue Jin, Hongliang Li, Xingwang Wang, and Shoufeng Hu
College of Computer Science and Technology, Jilin University, Changchun 130012, China
Email: {weixh, lihongliang}@jlu.edu.cn; yuejinjy@yahoo.com; wangxw2109@mails.jlu.edu.cn; shoufenghu@foxmail.com

***Abstract*** —Data centers hosting applications consume huge amounts of energy. Flexible management of virtual clusters is an effective way to reduce the overall power consumption of data centers. In order to save energy we focus on decreasing the total power-on time of physical nodes in data centers and increasing the runtime efficiency of physical nodes. For a specific physical node how long it is power-on is determined by the job with longest completion time. In this work, we formulate the problem into maximizing the amount of data which is processed by unit energy. AVC-based migration approach is proposed, which exploits the completion time of parallel jobs encapsulated by virtual clusters to guide virtual machines migrations. Algorithm *VCGM* is designed to construct the final mapping state which is energy-efficiency. In *VCGM*, all virtual clusters are grouped by their remaining time and every virtual cluster group is "compacted" into few physical nodes one by one. Then we adopt the low-cost perfect matching in the bipartite matching problem to obtain a migration plan. Algorithms are simulated and evaluated to verify the effectiveness of our energy efficiency scheme.

***Index Terms***—Virtual cluster migration, resource consolidation, green computing

## I. Introduction

Cloud computing has gained more and more attention from industry and academic in recent years. It leverages virtualization technology allowing customers to on-demand provision resources based on a pay-as-you-go utility model. Virtualization technologies have several properties: encapsulation, isolation and hardware independence, and so on. These properties make the way of utilizing data centers'computing resources in cloud changed. Virtualization is the core technology of cloud computing and views computing resources as a pool of unified resources, which reduces the complexity and manageability of data centers.

Infrastructure providers maintain thousands of physical computing nodes to meet the continuously increasing requirement. Cloud computing brings convenience to users and vendors, while we should not ignore the huge energy consumption it brings too [1], [2]. The work [3] estimates that a data center with 50,000 computing nodes may consume more than one hundred million kwh/year energy. This enormous energy consumption is equivalent to the electricity consumption for a 100,000 population urban in one year and it will generate heat and lead to expensive cooling costs.

The reasons for this extremely high energy consumption are: mass of computing resources, hardware power inefficiency, and resources inefficient usage. Large number of underutilized servers has become a major problem to cloud providers that needed to be figured out. As the base power consumption is the dominant part of total power consumption, no matter whether the node is busy or idle the energy drawn by it is huge. How long a physical node is power-on depends on the job with longest completion time. VM consolidation is an effective method to increase physical resources utilization and reduce the power consumption. Live migration is a valid method that migrate some VMs to other physical machines if necessary for VM consolidation.

Virtual Cluster (VC) [4], [5] is a group of VMs which are used to complete a parallel job. Virtual cluster becomes popular to run high performance computing workloads. And its sub-tasks will run on several virtual machines. The interaction operations between tasks are required for synchronization. The parallel job is not completed until all its tasks have been finished.

After a parallel job is assigned to a virtual cluster, the required resources of physical nodes must be allocated to all VMs belonging to this virtual cluster. The random placement of VMs belonging to same virtual cluster may cause some scattered resources fragment or some underutilized physical nodes generated when the assigned job finish. This is adverse to energy saving. At present, the researches on virtual cluster mainly focus on its deployment, but have little attention on its live migration. Virtual cluster live migration concerns the overall computing environment's migration. The impaction of

Manuscript received August 5, 2015; revised February 2, 2016.
This work was supported by the National Natural Science Foundation of China (NSFC) (Grant No.61170004), IBM Shared University Research (SUR) project (3D514BF41421), Specialized Research Fund for the Doctoral Program of Higher Education (20130061110052), Key Science and Technology Research Project of Science and Technology Department of Jilin Province (20140204013GX), Special Fund for Scientific Research in the Public Interest (SinoProbe09-01), and the research supported by China Geological Survey project (12120113006300).
Corresponding author email: lihongliang@jlu.edu.cn.
doi:10.12720/jcm.11.2.192-202

migration to parallel job performance should be considered.

Based on the mechanism Limevi [6] which supports virtual cluster live migration in parallel computation of large-scale distributed applications, we study the energy saving issue of a data center in cloud environment. In our data center, the future workloads are not been predicted and workloads change with time. We want to consolidate current workloads and make data centers in a low energy consumption state for a period of time. On one hand, we adopt VM consolidation to improve the amount of data which is processed by unit energy. On the other hand, during the consolidation we try to "compact" the scattered virtual cluster by the way of grouping. In a specific group, the remaining times of virtual machines to release resource are similar. It is assumed that the remaining times of virtual clusters are known when jobs are submitted. This can be predicted by machine learning in the practical environment [7]. Virtual cluster in one groupwill be remapped onto one or a set of physical nodes to increase the runtime efficiency of these physical nodes.

Specifically, our contribution is mainly about:

1) We analyze how the placement of virtual clusters impacts on the energy consumption of data centers. Then we obtain the VMs relationship in a certain virtual cluster in order to release more resources when some jobs finish.

2) We propose a power model which has a linear relationship with the utilization of physical nodes. With the workloads change with time, the energy consumption can be defined as a continuous function.

3) We group virtual clusters by their remaining time and generate several virtual cluster groups (*VCGs*). These *VCGs* are sorted by their remaining time in descendant order. We use the algorithm *VCGM* to place each *VCG* into the data center one by one for constructing the final mapping state.

4) In order to get the final mapping state from the initial mapping state, we regard it as a bipartite matching problem. By using the method of low-cost perfect matching, we get a migration plan whose migration cost is lower.

The rest of this paper is organized as follows. In Section 2 we present relate works. In Section 3 we discuss the motivation of energy efficient problem in virtual cluster and formulate it. In Section 4 we propose our consolidation method with bipartite matching. In Section 5 we evaluate and analyze the obtained experiment results. Finally, we make a conclusion for this paper and discuss future research directions in Section 6.

## II. Relate Works

There are a lot of researches on the energy saving resource management in the context of data center.

PMapper [8] is a power-aware application placement controller in a heterogeneous virtualized server system. The VMs placement is optimized to minimize power consumption and migration cost under performance constraints. The migration cost is determined by the impaction on throughput and the revenue loss based on Service Level Agreement (SLA). However, algorithms proposed in pMapper are locally optimal. In contrast to our work, they only get an energy-efficient placement scheme when consolidation, the energy cost of servers after the consolidation finishing is not considered.

In [9], Liu H *et al*. accurately predict the performance and energy consumption of each VM migration in a server farm. They design two models to estimate VM migration performance. A primary goal of its models is to determine which VM should be migrated within a server farm with minimum migration cost. They show that VM memory size, network speed and memory dirtying rate have most impaction on migration performance in terms of migration downtime, migration latency and network traffic. They only propose an energy model of VM migration and focus on the energy consumption when VM is migrating while the total energy consumption of the data center is not considered.

Beloglazov A *et al*. [10] focus on optimizing data center resource management for energy efficiency, while maintaining high service level performance. They define an architectural framework and present resource provisioning and three allocation algorithms for energy-efficient management of cloud computing environments. A power model which has a correlative with CPU utilization is proposed for formulating the energy consumption of modern data center. Hieu N T *et al*. [11] present a VM Consolidation with Usage Prediction (VMCUP) algorithm that limits the frequency of migrations and server switches, so as to reduce the energy consumption of a cloud data center. They used the multiple linear regression methods to estimate the utilization of resources in short-term future. Both [10], [11] only consider the resource of CPU, but the memory technology is still not energy-efficient and should to be taken into account as like what we have done.

Takahashi S *et al*. [12] consolidates servers based on the purpose of minimum the power consumption. They proposed a computational formula for the power decreasing by the migration of VMs from one server to other server. Then they use a matching algorithm to calculate a migration scheme which has the most weight with largest power consumption reduction. However, the proposed virtual machine placement algorithm is greedy heuristic and the algorithm complexity is higher.

Yang C T *et al*. [13] propose a power management scheme that determines the amount of physical nodes should be run by controlling the load ratio. They define a value that is the gross occupied resource weight ratio. A standby physical machine is wakened up when the value is greater than the maximum tolerant occupied resource weight ratio in order to prevent the degradation of performance. While if the value is less than the minimum occupied resource weight ratio that a running physical machines is selected to be migrated for energy saving. This work certainly reaches the goals of significant energy saving. However, it does not take the migration cost into account and a more actual model should be constructed.

Beloglazov A *et al*. [14] propose a VM placement optimization scheme which is decentralized and paralleled. The system architecture is hierarchical. The decentralization removes the problem of Single Point Failure (SPF) and improves scalability. The allocation policy has three stages and they combined in overall solution. The upper and lower utilization thresholds are used to judge which virtual machine should to be migrated.

For the problem of the whole virtual cluster or multiple virtual clusters migration has insufficient research and techniques.

Hsu C H *et al*. [15] propose a method of Energy-aware Task Consolidation (ETC), which to ration CPU utilization and manage task consolidation amongst virtual cluster. In their work, the network latency when task is migrating to another virtual cluster is considered. They use a non-linear energy consumption model which is different from ours. This model divides the energy consumption of a VM into seven different levels. Based on this power model, ETC uses the algorithm of Best Fit Decreasing (BFD) to consolidate tasks in order to keep the CPU utilization of virtual machines under its peek value. However, they only use virtual cluster scheduler to dispatch tasks, while not consider migrations of Virtual clusters like what we do. Ye K *et al*. [16] studies the performance and live migration costs of virtual cluster and researches on various live migration strategies for virtual cluster. They propose some optimization principles about virtual cluster migration based on the experiment results. But an efficient migration algorithm for the migration of virtual cluster does not been proposed.

In our work, we propose an effective virtual cluster migration scheme. In order to save energy the remaining times of virtual machines residing on a certain physical node should be similar when we construct the final mapping state. We take performance as constraint to meet SLA. When we migrating virtual clusters, we also take migration cost into account.

## III. Preliminaries and Model

In this section, we describe the motivation of virtual cluster based consolidation scheme. We then introduce a power model which is lineage with the resource utility. Finally, we formulate the energy consumption optimization problem.

### *A. Motivation*

The behavior characteristics of applications are different, so different application focus on different type resource. Compared with CPU and network resource, data intensive jobs have much more requirement for storage resource. Virtual cluster used to encapsulate these jobs and they will have different requirements to all resources.

We assume that the resources which are utilized by a virtual cluster will not be released until all tasks in it have been finished. That is to say, all VMs in a same virtual cluster will release its resources at same time. Based on this assumption, we analyze how the placements of virtual clusters impact on the energy consumption of data centers.

The random placement of virtual cluster will make its VMs' distribution scattered. When some short jobs finish, a small amount of resource will be released on some physical nodes. Usually, the parallel job running time is long and their resource requirement is diverse, so these released resources may not been used by other large scale virtual clusters. This will lead to these physical nodes in a low utilization ratio and inefficient energy usage. Especially for large and long jobs, scattered distribution of them will increase the total active time of physical nodes in a data center (such as Fig. 1 shows).

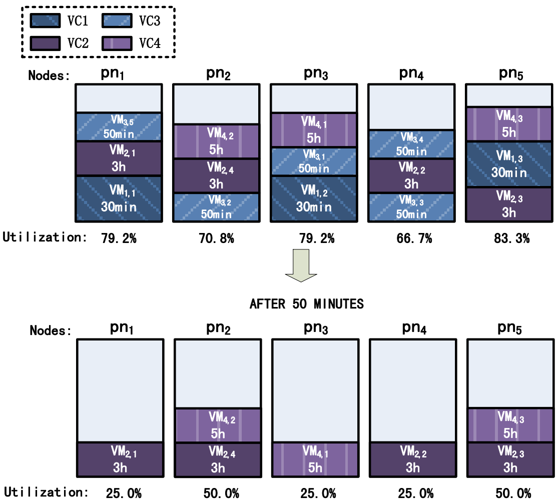



Fig. 1. System state changes after some VC finished.

Fig. 1 illustrates partial physical nodes states of data center. At the beginning, the system is efficiently used while the virtual clusters are mapping scattered. After 50 minutes, $VC_1$ and $VC_3$ finish their jobs and release resources occupied by them. The released resources of short jobs lead to inefficient usage of physical nodes and unnecessary energy consumption.

VM consolidation is a valid way to increase the utilization of physical nodes and reduce the energy consumption, but it will not lower the possibility of generating underutilized physical nodes. We want to consolidate virtual machines into few physical nodes meanwhile to "compact" virtual clusters.

If we can consolidate VMs that belong to the same virtual cluster into fewer physical nodes, this virtual cluster will release more resource on a physical node when its work finish. Moreover, in order to improve the migration performance of virtual clusters the, VMs belonging to the same virtual cluster should be deployed together as much as possible [16]. This will bring a benefit that the communication and Synchronization latency can be reduced mostly.

Based on the analysis above, we design a VC-based consolidation scheme which uses the correlations among VMs in same virtual cluster to save more energy.

### B. Energy Consumption Model

The power consumption of physical node generally is determined by CPU, memory, disk storage, and network. The utilization of CPU has more influence on its power consumption than other system resources. Physical node in different work state will have different power consumption. We divide the work state of a physical node into three modes: active, idle and off. In the off mode, the power consumption is 0. In the active and idle modes, the physical node is power-on and its power consumption is determined by its workload. References [10], [12], [17] show that a specific physical node in idle mode consumes 50% ~ 70% power of its saturated. That means physical node with lighter workload still consumes huge power. The base power consumption is the main part of a physical node's power consumption as long as it is power-on. So we focus on the base power consumption of physical nodes and want to reduce it as much as possible.

As discussed above, VM consolidation is a key technology to reduce the total power consumption for data centers. This is justified by judging whether a physical node with lightly workload has the opportunity to be an idle node or not. If so, VMs mapping on these physical nodes can be migrated to a few optimal set of physical nodes. Physical nodes inactive for a period of time can be switched to the Advanced Configuration and Power Interface (ACPI) S3 stand-by mode in which the power consumption is very low [18]. Our purpose is trying to minimize the total energy consumption of a data center without impacting the performance of application by eliminating unnecessary power consumption of lightly physical nodes.

We assume that the resources requirements of VMs are proportional. The resources of physical nodes also configured in proportion. This can be made sense from Amazon EC2 instance type [19]. For example, the instance of R3 in Amazon EC2, it has five different kinds of types. Table I shows the resource allocations of them. We consider that each type VM has a fixed bundle of resources. We can use one type resource to reflect the VM's resource requirement, here is memory.

TABLE I: RESOURCE ALLOCATIONS OF AMAZON EC2 INSTANCE TYPES.

| Type | vCPU | Memory(GB) | SSD(GB) |
|---|---|---|---|
| r3.large | 2 | 15.25 | 1×32 |
| r3.xlarge | 4 | 30.5 | 1×80 |
| r3.2xlarge | 8 | 61 | 1×160 |
| r3.4xlarge | 16 | 122 | 1×320 |
| r3.8xlarge | 32 | 244 | 2×320 |

### C. Formalization Description

Suppose that there are $N$ nodes in this data center and $M$ virtual clusters. Let $PN_i$ represents a single physical node in this data center and $VC_k$ represent a homogenous virtual cluster. The resources capacities of $PN_i$ and the resources requirements of $VC_k$ are represented by two vectors $c_i(cpu_i, mem_i, disk_i, net_i)$ and $r_k(cpu_k, mem_k, disk_k, net_k)$ respectively. In each virtual cluster $VC_k$, it has $Num_k$ virtual machines which are represented by $VM_{(k,j)}$. We use a binary variable $x_{i,(k,j)}^t$ to represent whether $VM_{(k,j)}$ is running on a node $PN_i$ at thetime $t$. We also use a variable $l_i^t$ to indicate whether $PN_i$ is active at time $t$. $PN_i$ is active means that there is at least one VM mapping on it at the moment.

$u_i(t)$ is used to represent the memory utilization of $PN_i$ at the time $t$ and it can be calculated from the following formulas:

$$u_i(t) = \frac{\sum_{(k,j)} r_k(mem_k) x_{i,(k,j)}^t}{c_i(mem_i)} \tag{1}$$

where $\sum_{(k,j)} r_k(mem_k) x_{i,(k,j)}^t$ shows how many memories of $PN_i$ are allocated tothose VMs which belongs to certain virtual clusters. It is divided by the capacity of $PN_i$, and then we can get the memory utilization of $PN_i$.

The power consumption of $PN_i$ that is correlated with memory utilization is given by:

$$P(u_i(t)) = \begin{cases} P_{base} + P_{active} \cdot u_i(t), & PN_i \notin off \\ 0, & PN_i \in off \end{cases} \tag{2}$$

where $P_{base}$ represents the base power consumption of an idle physical node; $P_{max}$ is the power consumption of a saturated physical node. $P_{active} = P_{max} - P_{base}$.

The workloads of the data center will change with time, so we define the total energy consumption of a physical node over a period of the time $(t_0, t_1)$ as:

$$E_i = \int_{t_0}^{t_1} P(u_i(t))dt \tag{3}$$

Our objective is to improve the amount of data which is processed by unit energy. It is formulated as follow:

$$maximize \left\{ \frac{\sum_k r_k(mem) \cdot NUM_k}{\sum_{PN_i \notin off} E_i} \right\} \tag{4}$$

where $\sum_k r_k(mem) \cdot NUM_k$ is used to show the total workloads of all jobs which have been dispatched to this data center and needed to be processed within the time span of $(t_0, t_1)$. $\sum_{PN_i \notin off} E_i$ represents the amounts of energy which are consumed by this data center during these jobs' processing.

Based on previous assumptions that the memory requirement of VM is stable all the time and the data center is homogenous, we give the theorem below.

***Theorem1:*** minimizing the total energy consumption of a data center can be simplified as minimizing the number of active physical nodes.

$$\text{minmize} \sum_{PN_i} \mathrm{E}_i = \text{minmize} \int_{t_0}^{t_1} \sum_{i=1}^{N} l_i(t) \quad (5)$$

*Proof.* In the homogenous data center, all physical nodes capacities of resources are same. At time *t*, the total energy consumption of this data center is:

$$\sum_{PN_i} \mathrm{E}_i = \sum_{t=t_0}^{t_1} \sum_{i=1}^{N} P(u_i(t))$$
$$= \sum_{t=t_0}^{t_1} \sum_{i=1}^{N} (P_{base} + P_{active} \cdot u_i(t))$$
$$= \sum_{t=t_0}^{t_1} \sum_{i=1}^{N} l_i^t \cdot P_{base} + P_{active} \cdot \sum_{t=t_0}^{t_1} \sum_{i=1}^{N} u_i(t)$$

Because the workload of this data center is determined at time *t*, $\sum_{t=t_n}^{t_1} \sum_{i=1}^{N} u_i(t)$ is a constant value. $\sum_{PN_i} \mathrm{E}_i$is determined by how many physical nodes are in active mode, which is $\sum_{t=t_0}^{t_1} \sum_{i=1}^{N} l_i^t$.

Before the formulation of our problem, we define a notation$r_k < \mathrm{c}_i$ which means that$cpu_k < cpu_i$, $mem_k < mem_i$ , $disk_k < disk_i$ and $net_k < net_i$ . Based onTheorem1, the objective (4) can be simplified as:

$$minimize \int_{t_0}^{t_1} \sum_{i=1}^{N} l_i(t)dt \quad (6)$$

Subject to:

$$\sum_{(k,j)} r_k \, x_{i,(k,j)}^t \le \mathrm{c}_i l_i^t \quad (7)$$

$$\sum_{i} x_{i,k,j}^t = 1 \quad (8)$$

$$x_{i,(k,j)}^t \in \{0,1\} \quad (9)$$

$$l_i^t \in \{0,1\} \quad (10)$$

Constraints (7) represents the resource constraint, that is, the total requirements of all VMs on a physical node cannot exceed its resource capacity. This constraint can avoid the performance degradation. Equation (8) represents that at any time there is only one $PN_i$ this $VM_{(k,j)}$ can reside on it.

In order to achieve this goal, we discretize object (6) and make the number of physical nodes that virtual cluters is mapping on as few as possible at any moment during $(t_0, t_1)$. This problem can be regarded as bin packing problem which have been proven to be NP-complete [2]. To quickly obtain an approximate solution, the heuristic algorithm First Fitting Decreasing (*FFD*) [20] is a state-of-art algorithm. By simply using *FFD*, we can get a solution which uses fewest physical nodes. *FFD* indeed improves the amount of data which is processed by unit energy and save energy of data centers at the consolidation moment. However, after the consolidation, the runtime efficiency of physical nodes cannot be guaranteed. How to release physical nodes as much as possible when short virtual clusters' jobs are finished later is not considered by *FFD*, which we will solve in next section. We modify the process of bin packing, and try our best to make those VMs with similar remaining time reside on some physical nodes. This may lead to use more physical nodes than *FFD*, but we try to keep it in an acceptable range.

## IV. Virtual Cluster Consolidation

Based on what we discussed in Section 3, we can reduce the total energy consumption by monitoring and controlling the number of active physical nodes. Determining the amount of active physical nodes is the first step of our VC-based consolidation scheme, and the virtual machines distribution of a virtual cluster will be considered next in order to make the release of resources more concentrated.

Algorithm *VCGM* described in this section is used to construct the final mapping states of all physical nodes in the data center. After the construction of final mapping state, we adopt the bipartite matching method to generate a migration plan with minimum migration cost. Based on this plan, virtual machines can be migrated in a feasible way.

### A. Final Mapping State Construction

For a virtual cluster, the more "compact" its VMs are, the bigger size of resources can be released from one physical node when the job finish. For a physical node, the more similar the remaining times of its VMs are, the higher the possibility of it to change into the stand-by mode or even shut down.

For VMs from the same virtual cluster, the remaining times that VMs occupy resources to handle tasks are same. We "merge" jobs whose remaining times are close. For detail, we sort virtual clusters based on their remaining times and then group them according to their remaining time. We put virtual clusters whose remaining times in $(-\Delta_t + remain_k,\ remain_k)$ into the *k*th virtual cluster group (*VCG*). At here, we use k-Means clustering algorithm [21] which is an efficient and simple algorithm to generate all groups. After grouping, we can get a set$\{VCG_i\}$ that each $VCG$ in it includes one or more virtual clusters.

How to construct an optimal final mapping state is critical to the final result. We want to make every active physical node in the final mapping state contains VMs belonging to different groups as few as possible. So we introduce the mapping state entropy of a physical machine in data centers. For a certain physical node, the fewer number of distinct *VCGs* which it's VMs belong to is, the more order the mapping state of this physical node is and the smaller the mapping state entropy is.

In order to get a final mapping State of this data center which mapping state entropy is lowest, we use the heuristic algorithm *VCGM* to remap VMs of every *VCG* in $\{VCG_i\}$ .*VCGM* is a modified first fit decreasing algorithm. We regard physical nodes as bins whose sizes are equal to physical nodes' resource capacity and VMs as items whose sizes are equal to VMs' resource requirement. As what we described before, all physical nodes are uniform. Each *VCG* in $\{VCG_i\}$ will be sorted first, and then will be packed one by one. The sorting order will be evaluated in Section 5. The algorithm is performed on every *VCG* and all its *VMs* will be sorted in descending order based on their resource requirement. The function acceptable() is called to guarantee the performance constraint. The procedure is as follows:

**Algorithm 1** *VCGM* (*pnSet*, $\{VCG_i\}$)

**Input:** a physical nodes set *pnSet*; a virtual cluster group set $\{VCG_i\}$;
**Output:***pnSet*;

```
sort VCGs in {VCG_i} by remaining time;
for VCG ∈ {VCG_i} do
  descending sort virtual machines of VCG by resource
      requirement;
for vm ∈ VCG do
for PN_1 ∈ pnSet do
  if acceptable(PN_1, vm) then
  mapping vm to PN_1;
return pnSet;
```

### B. A Low-Cost Perfect Matching

How to get the final mapping state from the initial mapping state with lower cost can be regarded as a bipartite matching problem. The initial mapping state and final mapping state can be divided into two physical nodes set *IS* and *FS*. Both *IS* and *FS* have *n* elements. $IS = \{p_1, p_2, \dots, p_n\}$ and $FS = \{q_1, q_2, \dots, q_n\}$. For $p_i$ in *IS* and $q_j$ in *FS*, they have their own list $IList_i$ and $FList_j$ which are used to save all virtual machines mapping on $p_i$and $q_j$.Every elements in *IS* are all connections with *FS*(as Fig.2 shows). No such edge exists that its endpoints in the same set of $IS$or$S$. A matching $T$ defined here is a collection of edges that every element of $\{IS \cup FS\}$ is incident to at most one edge of *T*. *T* is a subset of the whole edges set. We say an element is exposed means that there is no edge of *T* incident to it. A matching is perfect if no element is exposed. For a perfect matching, its cardinality is equal to $|IS| = |FS|$.

An edge $e = (p_i, q_j)$ existsbetween $p_i$ and $q_j$ means that the physical node $PN_i$ 's initial state $p_i$ will be transformed into the final state $q_j$. The transformation from $p_i$ to $q_j$ will bring migration cost $c_{ij}$ which can be calculated from the difference between $ILis\ _i$ and $FList_j$. That is to say, a migration must have happened if $vm \in FList_j$ and $vm \notin IList_i$ and it will bring a migration cost. The cost of a virtual machine's migration depends on its resource requirement.

$$c_{ij} = \sum_{vm} r_k(mem) \quad vm \in FList_j, vm \notin IList_i \quad (11)$$

We have given costs $c_{ij}$ for every edgesand the goal is to find a perfect matching $T$ minimizing $\sum_{(p_i,q_j)\in T} c_{ij}$. We can use the minimum cost perfect matching model to solve it. This problem can be formulated as an Integer Linear Programming (ILP). For a given perfect matching *T*, let its incidence vector be *y* where $y_{ij} = 1$ if $(p_i, q_j) \in T$, otherwise $y_{ij} = 0$. So we can formulate the minimum weight perfect matching problem as follows:

$$\text{minimize} \sum_{i,j} c_{ij}\, y_{ij} \quad (12)$$

Subject to:

$$\textstyle\sum_j y_{ij} = 1 \quad i \in A \quad (13)$$

$$\textstyle\sum_i y_{ij} = 1 \quad j \in B \quad (14)$$

$$y_{ij} \in \{0,1\} \quad (15)$$

This optimization problem can be solved by a classical algorithm *Kuhn-Munkras* (KM) [22]. The solution to this ILP corresponds to a perfect matching and this matching is corresponding to a migration plan.

### C. Migration

After the migration plan is generated, we should make each migration feasible in order to get the real final mapping state. In first step, we will generate a matching triples list *mtList* based on the migration plan that includes all matching triples which not been migrated yet. A matching triple includes three components: the migrating virtual machine *vm*, the source physical node *sp*, the target physical node *tp*. In a matching triple, if *sp* is same with *tp*, which means that the destination of *vm* equals to the source, and *vm* will not be migrated. This kind of matching triples can be deleted from *mtList*.

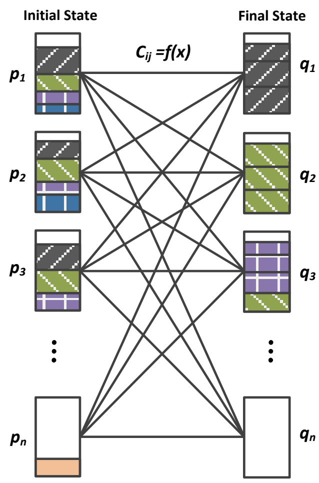



Fig. 2.The connection between *IS* and *FS*

In second step, we pick out initial feasible matching triples which can be migrated successfully. Those feasible matching triples can be deleted from *mtList* and conduct migrations according to these triples. After the preceding migrations, some infeasible matching triples will become feasible. And this process will be run again until there is no more feasible matching triples can be found.

In third step, there are two situations. The first is that there is not have any matching triples in the list. That is to say, the migration plan has been finished successfully. The second is that there exists some deadlock (or cyclic) among matching triples. We exploit a physical nodes which role as a buffer to temporarily hold one or some VMs in order to break the deadlock or cyclic. Make some infeasible matching triples to be feasible and return back to second step. Based on three steps above, we can get the final mapping state successfully.

## V. Evaluation

In this section, the evaluation of our approach has been done. We introduce the experiment setup first. The simulation results have been obtained from a set of experiments. We also compare those results with the FFD and low cost FFD (*LC-FFD*) schemes.

For a generic cloud computing environment, it is essential to evaluate it on a large-scale virtualized data center. While it is difficult to conduct repeatable experiments on a real environment, simulations have been chosen to evaluate the performance of the proposed heuristic algorithm.

### A. Experiment Setup

We have simulated a data center comprising 1000 homogeneous physical nodes. Each node is modeled to have a dual-core CPU with the performance equivalent to 3000MIPS, 8GB of RAM. The power model has been proposed in Section 3. According to it, the base power consumption of a physical node is 175W, and the saturated nodes will consume about 250W.

In order to simulate a real workload, we classify all *VC*s. According to virtual cluster's resource requirement, we divided all *VC*s into three types: light, medium, and heavy. For the light *VC*s, their resource requirement is set under 30% of nodes resource capacity; for the medium *VC*s, their requirement is set between 30% and 50% of nodes resource capacity; for heavy *VC*s, their requirement is above 50% of nodes resource capacity. Based on the remaining time, each kind of *VC*s also can be classified into short-term, medium-term and long-term.

As what we described above, we totally have 9 types of *VC*s. And for a single virtual cluster, the number of virtual machines belongs to It is between five and ten. At the beginning of this experiment, we have an initial mapping state. The proportions of all types of *VC*s are normally distributed. All experiments have been run 50 times and take the average.

### B. Simulation Results

As we mentioned in Section 4, *VCGs* belonging to $\{VCG_i\}$ should be sorted by a certain order in algorithm *VCGM*. In the first experiment, we test two kinds of sorting order: *LRF* (Long remaining time first) and *SRF* (Short remaining time first). As Fig. 3 shows, the number of active physical nodes in *VCGM-SRF* is less than that in *VCGM-LRF* when the consolidation finishing at time step 1. However, with some jobs completed, *VCGM-LRF* released more physical nodes than *VCGM-SRF* did. The reason is that long-term *VCG* is consolidated firstly which will migrate virtual machines belonging to it away from some physical nodes. In the situation of without new jobs submitting, this will increase the stand-by probability of the sephysical nodes and reduce their total power-on time which is good for energy saving. So when we used *VCGM* to construct the final mapping state, *VCGM-LRF* is better than *VCGM-LRF*. In the following experiment we acquiescently adopt *LRF* as our sorting order.

We compare *VCGM* with three schemes. For the benchmark experiment we use the non-consolidation policy (*No-Conso*). *No-Conso* does not execute any consolidation during the whole experimental time. Although some jobs have been finished, some physical nodes are still active with a low utilization. *No-Conso* is used here to show that how much energy consumption can be saved by an effective consolidation scheme.

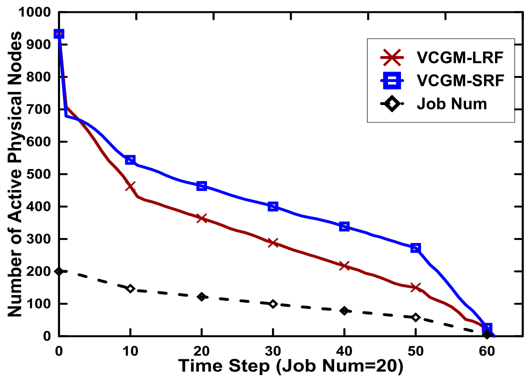



Fig. 3. The efficiency of *LRF* and *SRF*in *VCGM*.

For contrast experiments we use another two modified *FFD* consolidation algorithms. The first is the basic first fit decreasing (*FFD*) algorithm which is a state of art algorithm to solve the bin packing problem. It sorts all virtual clusters by their requirement in descending order and then remapping all virtual clusters one by one.

The second is the low cost first fit decreasing algorithm (*LC-FFD*). In *LC-FFD*, all active physical nodes are sorted by their utilization in descending order and divide into two sets. It calculates the least number of physical nodes ($N^*$) that all jobs needed and put the first $N^*$ nodes into target set and put the rest physical nodes into source set. Those virtual machines mapping on source physical nodes will be migrated to target physical nodes. If the spare capacity on all physical nodes in target set cannot meet the requirement of the migrating VM, the physical node which this VM reside on will be removed from the source set and added into the target set. *LC-FFD* is similar with an algorithm *iFFD* which is proposed in Reference [8]. *LC-FFD* tries to construct an optimal final mapping state while minimize migration costs by migrating as few VMs as possible.

We have studied above algorithms with respect to the active physical nodes numbers varies with time, the total energy consumption, and the migration performance. We conducted a comparative study of the algorithms with change in the system workload. The different workloads are 50, 100, 150 and 200.

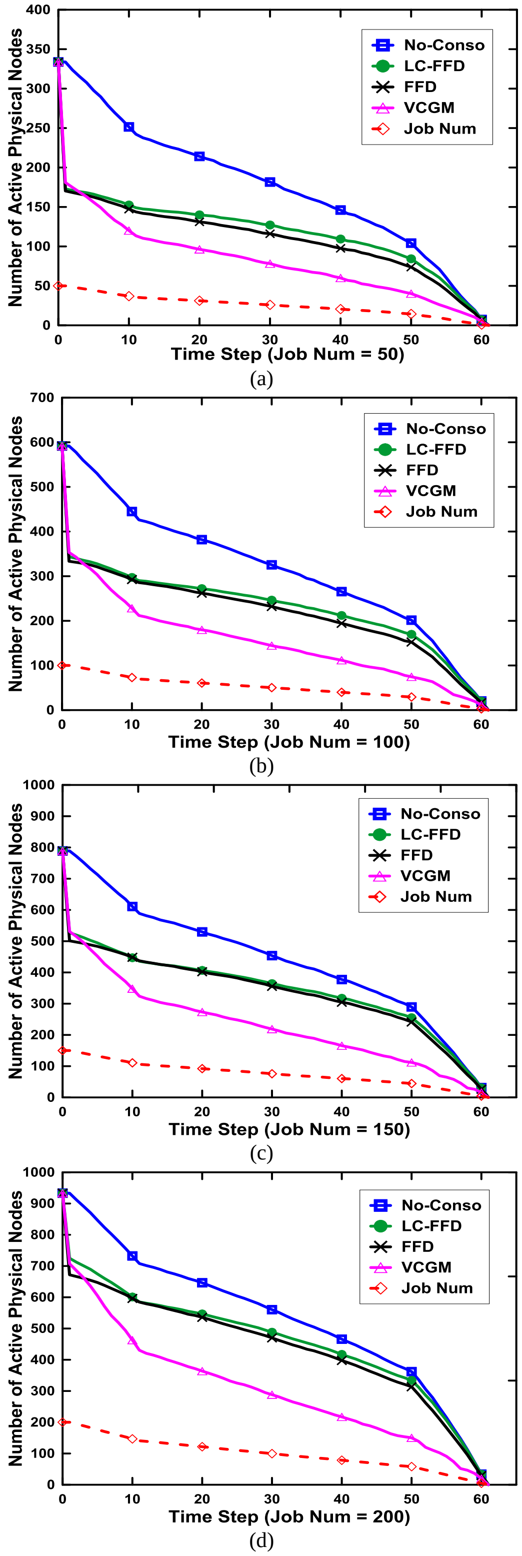



Fig. 4.The number of active physical nodes varies with time.

We first investigate the number of active physical nodes varies with time (Fig. 4 (a) (b) (c) (d)). In Fig. 4, the horizon axis represents the discrete time steps and each time step is a custom smallest time unit. With time goes on, some jobs were finished and the resources occupied by them were also released. At the beginning, the numbers of active physical nodes are dropped heavily and this owe to the effectiveness of consolidation algorithms. While with the release of resources, the freed physical nodes number is different.

With the workload of data center changed from light to heavy, *VCGM* is always better than both *FFD* and *LC-FFD*. For *LC-FFD*, only those VMs mapping on source physical nodes can be migrated. And the virtual machines on physical nodes in target set will never change during the whole migration. For *FFD*, its purpose is how to get an optimal solution of minimizing the active physical nodes after consolidation. *FFD* only concerns the total number of physical nodes to serve all virtual clusters without concerning the remaining running time of allvirtual machines. While for *VCGM*, it will divide all virtual clusters into several groups based on the remaining time and the virtual machines belongs to same group will be consolidated together, that is to say, after consolidation each *VCG* will be more “compact” than before. After using *VCGM* to make a consolidation at time step 1, the remaining time of all VMs mapping on a physical node is similar. Those physical nodes with short remaining time will be freed later and the amount of freed physical nodes at subsequent time steps is more than no-grouping algorithms. We also observed that the heavier the workload of this data center is, the more effective the *VCGM* is.

We next investigate the total energy consumption of this data center which spanning over the whole experiment time steps. Fig.5 represents these results with different workloads. *VCGM* leads to 42% - 55% less energy consumption on average than that without consolidation scheme brings. Comparing with the energy consumption *LC-FFD* brings, *VCGM* can save 32% - 43% of it and the more jobs in the data center the more effective *VCGM* is.

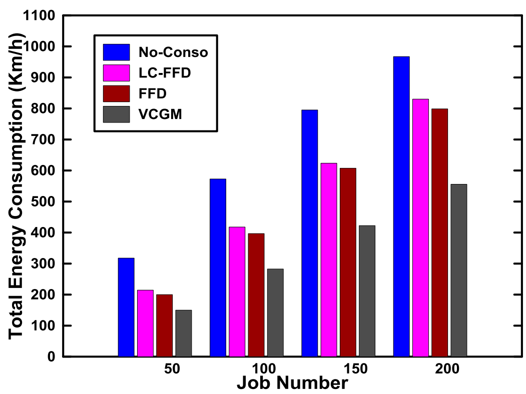



Fig. 5. The total energy consumption.

We observe that the total energy consumption drawn by the scheme with *VCGM* is significantly lower than schemes with *LC-FFD* and *FFD*. It allows us to conclude that *VCGM* can effective minimum the total energy consumption. This benefit from avoiding the long term running of physical nodes being in an under-utilization state in order tore leases physical nodes as more as possible. This establishes the importance of taking the remaining running time into account, while coming up with consolidation virtual clusters. We make the virtual cluster group more "compact" and the total active time of physical nodes in this data center is decreasing.

At last, the experiment of testing migration performance has also been done. The migration performance can be tested from two aspects: the number of active physical nodes before and after consolidation, the migration cost ratio. At time step 0, we have an initial mapping state and this state will be consolidated at next time step. From the Fig. 6 we can get these active physical numbers after consolidation, it shows that *FFD* is the best one. *VCGM* is as good as the *FFD* and outperform *LC-FFD*. For *LC-FFD*, the previous placement on target physical nodes cannot be changed. It seems to appear the situation that some source node is added into the target set for meeting those unsatisfied VMs requirements.*LC-FFD* may use more physical nodes than its expectation. It is obvious that *VCGM* can construct a final mapping state with an acceptable number of active physical nodes.

The migration cost ratio is defined as the total migration cost divided by the total required resource of all virtual clusters. Fig. 7 shows the results of the three consolidation algorithm. Just as *LC-FFD*'s name implies the migration cost of it is significantly lowest and when workload of this data center is almost full its migration cost is much lower. Since the number of physical nodes in target set is almost close to data centers upper limit. So the possibility of virtual machines' successful migration becomes tiny and a little of migration costs will be brought by *LC-FFD*.

For the *FFD* consolidation scheme the migration cost ratio of it is higher than *LC-FFD*. Because its sort all virtual machines in descending order by their resource requirement, this will lead to most of virtual machines are migrated to get the optimal number of active physical nodes. For *VCGM* consolidation scheme the migration cost ratio of it is also higher than *LC-FFD* and the migration cost ratio has no relevant with the amount of workloads in this data system.

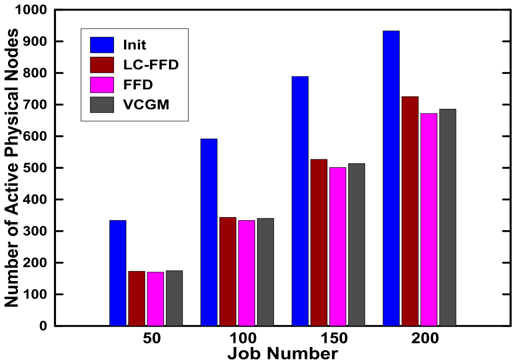



Fig. 6. The number of physical nodes occupied

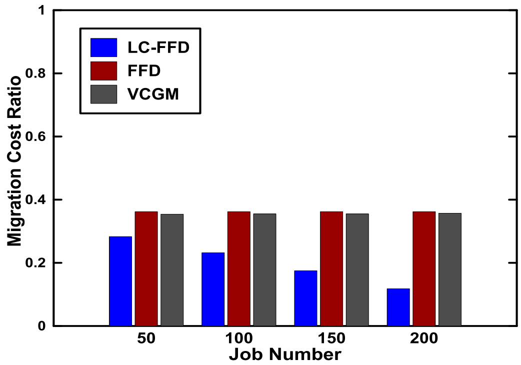



Fig. 7. The migration ratio

That is because, in order to "compact" the *VCGs*in $\{VCG_i\}$, there must have some additional migration happened. The more scattered the *VCG* is, the more migration cost it will bring. But the migration ratio is stable when the workloads of this data center vary from light to weight. Although the ratio of *VCGM* is higher than the ratio of *LC-FFD* and is similar to the ratio of *FFD*, the energy effective is much better than *FFD* and *LC-FFD*.

## VI. Conclusion and Future Work

Virtual resource consolidation in clouds has become an important approach to improve energy efficiency. Based on the fact that resource utilization directly relates to energy consumption, a power model with respect to the physical nodes resource utilization has been defined in this paper. The correlation of virtual clusters deadlines is also be used to guide the virtual machines migration. This approach will classify virtual clusters by the property of deadline and put different type virtual cluster into different physical nodes. This can decrease the total active time of physical nodes in this data center. With the finishing of some jobs, physical nodes will be released and turned into low power model. A validate migration plan which its migration cost is lower can be generated by the low cost perfect matching model. The evaluation results demonstrate that we can save total energy consumption by 42%-53%. The validation of the heuristics *VCGM* shows that our approach increases energy-saving possibilities.

We will make further research on the final mapping state construction in order to get a lower cost migration plan. We plan to extend our work into a heterogeneous data center which is a more general environment.

## Acknowledgment

This work is supported by the National Natural Science Foundation of China (NSFC) (Grant No.61170004), IBM Shared University Research (SUR) project (3D514BF41421), Specialized Research Fund for the Doctoral Program of Higher Education (20130061110052), Key Science and Technology Research Project of Science and Technology Department of Jilin Province (20140204013GX), Special Fund for

Scientific Research in the Public Interest (SinoProbe09-01), and the research supported by China Geological Survey project (12120113006300). Hongliang Li is the corresponding author.

## REFERENCES


[1] L. X. Zhang, *et al.*, "Maximizing reliability with energy conservation for parallel task scheduling in a heterogeneous cluster," *Information Sciences*, vol. 319, pp. 113–131, 2015.

[2] T. Zhou, *et al.*, "An energy-efficient task scheduling algorithm in DVFS-Enabled cloud environment," *Journal of Grid Computing*, pp. 1-20, 2015.

[3] A. Greenberg, *et al.*, "The cost of a cloud: Research problems in data center networks," *AcmSigcomm Computer Communication Review*, vol. 39, no. 1, pp. 68-73, 2009.

[4] I. Foster, *et al.*, "Virtual clusters for grid communities," in *Proc. Sixth IEEE International Symposium on Cluster Computing and the Grid*, 2006, pp. 513-520.

[5] K. J. Ye, *et al.*, "Analyzing and modeling the performance in xen-based virtual cluster environment," in *Proc. 12th IEEE International Conference on High Performance Computing and Communications*, 2010, pp. 273-280.

[6] X. H. Wei, *et al.*, "LimeVI: A platform for virtual cluster live migration over WAN," *Computer Systemsence & Engineering*, vol. 26, no. 5, pp. 353-364, 2011.

[7] M. Dabbagh, *et al.*, "Toward energy-efficient cloud computing: Prediction, consolidation, and overcommitment," *Network IEEE*, vol. 29, no. 2, pp. 56-61, 2015.

[8] A. Verma, P. Ahuja, and A. Neogi, "pMapper: Power and migration cost aware application placement in virtualized systems," *Middleware 2008*, Springer Berlin Heidelberg, 2008, pp. 243-264.

[9] H. K. Liu, *et al.*, "Performance and energy modeling for live migration of virtual machines," *Cluster Computing*, vol. 16, no. 2, pp. 249-264, 2013.

[10] A. Beloglazov, J. Abawajy, and R. Buyya, "Energy-Aware resource allocation heuristics for efficient management of data centers for cloud computing," *Future Generation Computer Systems*, vol. 28, no. 5, pp. 755-768, 2012.

[11] N. T. Hieu, M. D. Francesco, and A. Yla-Jaaski, "Virtual machine consolidation with usage prediction for energy-efficient cloud data centers," in *Proc. IEEE 8th International Conference on Cloud Computing (CLOUD) IEEE Computer Society*, 2015, pp. 750-757.

[12] S. Takahashi, *et al.*, "Virtual machine packing algorithms for lower power consumption," in *Proc. IEEE 5th International Conference on Cloud Computing Technology and Science IEEE*, 2012, pp. 161-168.

[13] C. T. Yang, *et al.*, "A method for managing green power of a virtual machine cluster in cloud," *Future Generation Computer Systems*, vol. 37, no. 7, pp. 26–36, 2014.

[14] A. Beloglazov and R. Buyya, "Energy efficient resource management in virtualized cloud data centers," in *Proc. 10th IEEE/ACM International Conference on Cluster, Cloud and Grid Computing IEEE Computer Society*, 2010, pp. 826-831.

[15] C. H. Hsu, *et al.*, "Optimizing energy consumption with task consolidation in clouds," *Information Sciences*, vol. 258, no. 3, pp. 452-462, 2014.

[16] K. J. Ye, *et al.*, "VC-Migration: Live migration of virtual clusters in the cloud," in *Proc. ACM/IEEE 13th International Conference on Grid Computing*, 2012.

[17] Y. C. Lee and A. Y. Zomaya, "Energy efficient utilization of resources in cloud computing systems," *The Journal of Supercomputing*, vol. 60, no. 2, pp. 268-280, 2012.

[18] L. M. Surhone, *et al*., "Advanced configuration and power interface," *Green Tech*, vol. 16, pp. 85-100, 2010.

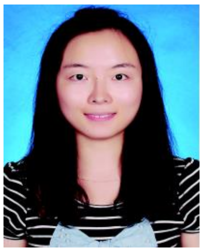
[19] Amazon EC2 Instance Types. [Online]. Available: http://aws.amazon.com/cn/ec2/instance-types/

[20] M. Y. Yue, "A simple proof of the inequality FFD (L)⩽ 11/9 OPT (L)+ 1,∀ L for the FFD bin-packing algorithm," *Acta. Mathematicae Applicatae Sinica*, vol. 7, pp. 4, pp. 321-331, 1991.

[21] T. Kanungo, *et al*., "An efficient k-means clustering algorithm: Analysis and implementation," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 24, no. 7, pp. 881-892, 2002.

[22] J. J. Kosowsky and A. L. Yuille, "The invisible hand algorithm: Solving the assignment problem with statistical physics," *Neural Networks*, vol. 7, no. 3 pp. 477-490, 1994.

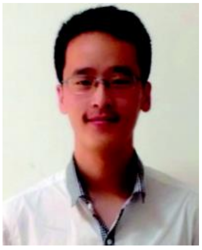

**Xiaohui Wei** is a Professor and the Dean of the College of Computer Science and Technology (CCST) at Jilin University. He is currently the Director of High Performance Computing Center of Jilin University. His current major research interests include resource scheduling for large distributed systems, infrastructure level virtualization, large-scale data processing system and fault-tolerant computing. He has published more than 50 journal and conference papers in the above areas.



**Yue Jin** is a master and received his bachelor's degree from the College of Computer Science and Technology (CCST), Jilin University in 2013. Her research interests include resource management in cloud systems.

**Hongliang Li** received his PhD from the College of Computer Science and Technology (CCST), Jilin University. He is currently a lecturer of CCST. His research interests include resource scheduling and fault tolerance in HPC systems. He is working on a China NFS project on elastic virtual cluster and parallel job scheduling under virtual HPC environment.

**Xingwang Wang** received the bachelor'sdegree in College of Computer Science and Technology from Jilin University in 2013, and is currently working toward the PhD degree in College of Computer Science and Technology, Jilin University. His research interests include cloud computing, mobile computing and security computing.



**Shoufeng Hu** received his master's degree from the College of Computer Science and Technology (CCST), Jilin University in 2015. His research interests include resource management and virtualization in cloud systems. He is currently working at IBM Inc.