Information Sciences 258 (2014) 452–462

Contents lists available at SciVerse ScienceDirect

Information Sciences

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

Optimizing energy consumption with task consolidation in clouds

Ching-Hsien Hsu [a,*], Kenn D. Slagter [b], Shih-Chang Chen [b], Yeh-Ching Chung [b]

[a] Department of Computer Science, Chung Hua University, Hsinchu, Taiwan 300, ROC
[b] Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan 300, ROC

ARTICLE INFO

Article history:
Available online 1 December 2012

Keywords:
Energy consumption
Cloud computing
Task consolidation

ABSTRACT

Task consolidation is a way to maximize utilization of cloud computing resources. Maximizing resource utilization provides various benefits such as the rationalization of maintenance, IT service customization, and QoS and reliable services. However, maximizing resource utilization does not mean efficient energy use. Much of the literature shows that energy consumption and resource utilization in clouds are highly coupled. Consequently, some of the literature aims to decrease resource utilization in order to save energy, while others try to reach a balance between resource utilization and energy consumption. In this paper, we present an energy-aware task consolidation (ETC) technique that minimizes energy consumption. ETC achieves this by restricting CPU use below a specified peak threshold. ETC does this by consolidating tasks amongst virtual clusters. In addition, the energy cost model considers network latency when a task migrates to another virtual cluster. To evaluate the performance of ETC we compare it against *MaxUtil*. *MaxUtil* is a recently developed greedy algorithm that aims to maximize cloud computing resources. The simulation results show that ETC can significantly reduce power consumption in a cloud system, with 17% improvement over *MaxUtil*.

1. Introduction

Cloud computing has recently become popular due to the maturity of related technologies such as network devices, software applications and hardware capacities. Resources in these systems can be widely distributed and the scale of resources involved can range from several servers to an entire data center. To integrate and make good use of resources at various scales, cloud computing needs efficient methods to manage them [4]. Consequently, the focus of much research in recent years has been on how to utilize resources and how to reduce power consumption.

One of the key technologies in cloud computing is virtualization. The ability to create virtual machines (VMs) [14] dynamically on demand is a popular solution for managing resources on physical machines. Therefore, many methods [17,18] have been developed that enhance resource utilization such as memory compression, request discrimination, defining threshold for resource usage and task allocation among *VMs*. Improvements in power consumption, and the relationship between resource usage and energy consumption has also been widely studied [6,10–12,14–18]. Some research aims to improve resource utilization while others aim to reduce energy consumption. The goals of both are to reduce costs for data centers. Due to the large size of many data centers, the financial savings are substantial.

Energy consumption varies according to CPU utilization [11]. Higher CPU utilization usually implies greater energy consumption. However, higher CPU utilization does not equate to energy efficiency. This phenomenon motivates the idea of not

* Corresponding author.
E-mail addresses: chh@chu.edu.tw (C.-H. Hsu), kennslagter@sslab.cs.nthu.edu.tw (K.D. Slagter), scchen.thomas@gmail.com (S.-C. Chen), ychung@cs.nthu.edu.tw (Y.-C. Chung).

0020-0255/$ - see front matter
http://dx.doi.org/10.1016/j.ins.2012.10.041

exhausting CPUs with high levels of utilization (for example, 80–100%) in order to save energy. To this end, we propose an energy-aware task consolidation (*ETC*) method that minimizes energy consumption.

The main idea of *ETC* is to ration CPU utilization and manage task consolidation amongst virtual clusters. In addition, our energy cost model considers network latency when a task migrates to another virtual cluster. The main contributions of our work are as follows. First, we present a method to ration CPU utilization and manage task consolidation amongst virtual clusters. Secondly, we show how *ETC* can reduce power consumption significantly by managing task consolidation in a cloud system. Third we compare our results to a recent greedy method called *MaxUtil* [10] that attempts to reduce energy consumption by assigning as many tasks as it can to a *VM*.

The rest of this paper is organized as follows. Section 2 explains our research model. Section 3 presents the proposed techniques on task consolidation and energy saving. In Section 4, the simulation results and performance analysis are given to weigh the pros and cons of the proposed method. In Section 5, we discuss related work. Finally, the conclusion and future work are presented in Section 6.

2. Research model

The research model for this study is presented in Fig. 1, which shows a cloud system that consists of several virtual clusters (VC). Each virtual cluster provides a limited number of *VM*s. Without losing generality, *VM*s are used as a basic unit to execute a task. The percentage of CPU utilization is used to judge whether a *VM* has enough resources available for a service. Fig. 1b gives an example of network bandwidth between virtual clusters, which are geographically distributed. This shows that communication overhead can differ between virtual clusters. The network bandwidth between clusters is assumed to range between 100 Mb/s and 1 Gb/s at different links, which is representative of conditions in practice. The number of nodes at each cluster depends on the resources available for release. It is assumed that tasks are submitted through a queue in the cloud. The submitted tasks give information that is required for CPU utilization and are allocated to the appropriate *VM*s according to the CPU utilization of both tasks and *VM*s. Because the resources available at each cluster can vary at different times, a cluster has its own strategy to consolidate tasks in order to minimize energy consumption. Namely, a cluster can request resource support from other clusters and then consolidate tasks to an appropriate VM with available resources. Each cluster has a job queue that contains information of all the tasks, such as task ID (t_j), arrival time of task t_j ($T_{a,j}$), CPU processing time of task t_j ($T_{p,j}$), data size of task t_j (DS_j) and CPU utilization. Based on this information, task consolidation is able to satisfy the service level agreement (SLA). The purpose of the SLA is to meet the needs of customers and requires the service provider to have a system that can sustain an agreed upon level of energy efficiency, economy and performance.

Various research results in the literature show that CPU utilization significantly affects energy consumption. One of the general concepts in the literature shows how energy consumption can be separated in two states [8,12,17], an idle state and a running state. A well-established energy consumption model [12] shows the relationship between CPU utilization and energy consumption is not a linear one, as is shown in Fig. 2a. The curves in Fig. 2a represent power consumption on different machines. We observe that the slopes on these curves are smallest when CPU utilization is between 0% and 20%. Essentially, this interval can be regarded as the idle state for the VMs. Between 20% and 50% CPU utilization, power consumption increases slightly. Between 50% and 70% CPU utilization, power consumption noticeably rises. Finally, between 70% and 100% CPU utilization the rate of power consumption increases greatly. It is worth mentioning that the slopes of the curves in Fig. 2a represent energy efficiency, which is defined as the amount of CPU resources obtained per energy unit.

Based on this study, a simplified energy model can be constructed. This simplified energy model is outlined in Fig. 2b. In Fig. 2b the energy consumption of a *VM* is divided into six different levels. These levels consist of an idle state and six

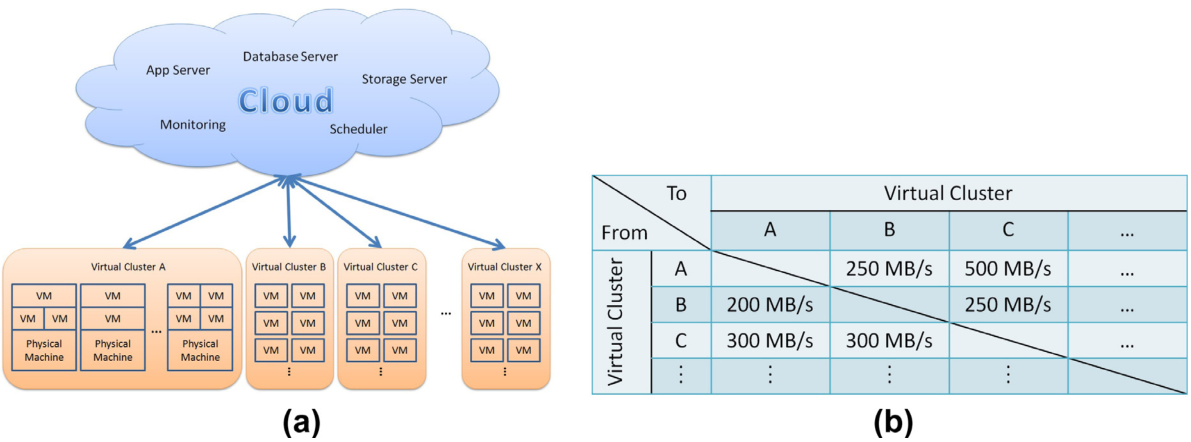

Fig. 1. Research model in this paper: (a) a cloud system composed of virtual clusters and (b) network bandwidth between virtual clusters.

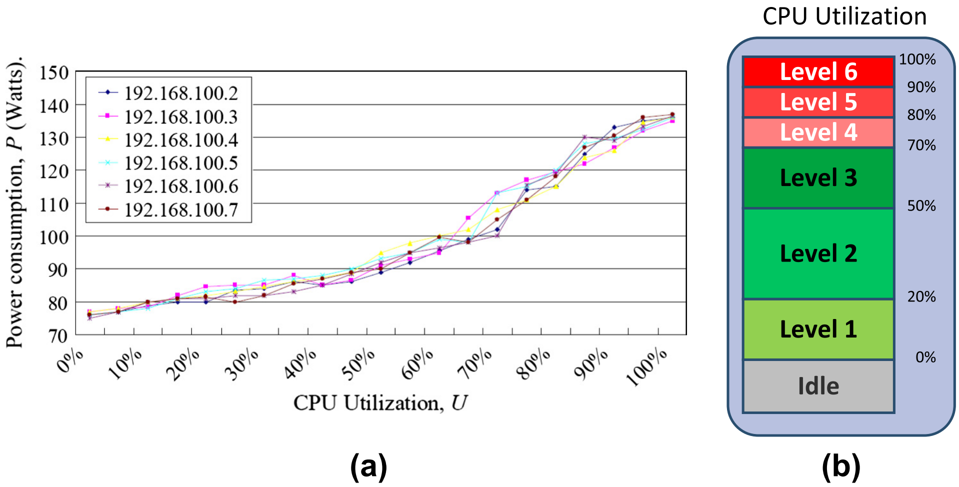

Fig. 2. The relationship between energy consumption and CPU utilization: (a) experimental results in [12] and (b) the suggested model in this paper.

different levels of CPU utilization. All of these levels are based on line segment approximations of the curve in Fig. 2a. In summary, because CPU use has the most impact on energy use, and because the relationship of energy consumed and CPU utilization is not linear, one can consume less energy if one shares the workload amongst multiple CPU's. This gives us the motivation to keep CPU utilization below a certain level. In this paper, this is done by redistributing tasks evenly amongst VM's.

Because resources on each cluster are limited, task migration between virtual clusters can occur if there are not enough resources (VMs) on a local cluster. In these circumstances, it is possible to consolidate tasks with resources that reside on different clusters. Since task migration incurs extra overhead, there is a tradeoff between the energy efficiency task consolidation provides and the cost of migrating the tasks. Based on the study in [2,3], the network infrastructure of a cloud system can be regarded as a passive optical network, FTTN or point-to-point optical system. As a result, the power consumption for network access and communications is about 2β W/s with access rates ranging from 100 Mb/s to 1 Gb/s.

In this paper, we assume that network transmissions occur on virtual clusters. Although they might be on different racks, we assume all virtual clusters and virtual machines reside within the same data center. Therefore, we assume network transmissions between virtual clusters are constant. A corresponding energy cost model and a detailed strategy for task scheduling is addressed in the next section.

3. Energy efficient task consolidation

In this section, we present an *energy-aware task consolidation* (*ETC*) method to optimize energy usage in cloud systems. In the energy model presented in Fig. 2b, a *VM* is assumed to consume α W/s in its idle state. An additional β W/s is required for executing tasks when CPU utilization is between 0% and 20%. If CPU utilization is between 20% and 50%, the additional energy consumed increases to 3β W/s. Energy is consumed at a greater rate as CPU utilization increases. For instance, when a virtual machine has 50% CPU utilization, it consumes $\alpha + 5\beta$ W/s. Using Fig. 2a as a guide the energy consumption of a virtual machine V_i is defined as follows:

$$E(V_i) = \begin{cases} \alpha \text{ W/s}, & \text{if idle} \\ \beta + \alpha \text{ W/s}, & \text{if } 0\% < \text{CPU utilization} \leqslant 20\% \\ 3\beta + \alpha \text{ W/s}, & \text{if } 20\% < \text{CPU utilization} \leqslant 50\% \\ 5\beta + \alpha \text{ W/s}, & \text{if } 50\% < \text{CPU utilization} \leqslant 70\% \\ 8\beta + \alpha \text{ W/s}, & \text{if } 70\% < \text{CPU utilization} \leqslant 80\% \\ 11\beta + \alpha \text{ W/s}, & \text{if } 80\% < \text{CPU utilization} \leqslant 90\% \\ 12\beta + \alpha \text{ W/s}, & \text{if } 90\% < \text{CPU utilization} \leqslant 100\% \end{cases} \quad (1)$$

If $\alpha = \beta$, then each β energy consumed contributes to 25% CPU utilization. This occurs when CPU utilization is below 50%. When a virtual machine has 70% CPU utilization, it consumes 3β W/s, which means that each β energy consumed contributes to 23.3% CPU utilization. For the cases of 80%, 90% and 100% utilization, the energy efficiency is at 20% per β W/s, 18% per β W/s and 16.6% per β W/s, respectively. Based on the definition above, the total energy consumption of V_i during the time period $t_0 \sim t_m$ can be given the following formula,

$$E_{0,m}(V_i) = \sum_{t=0}^{m} E_t(V_i) \tag{2}$$

Given a virtual cluster, VC_k which consists of n VMs, the energy consumption of VC during the time period $t_0 \sim t_m$ is as follows,

$$E_{0,m}(VC_k) = \sum_{i=0}^{n} E_{0,m}(V_i) \tag{3}$$

The main idea of *ETC* is to consolidate tasks and to keep the CPU utilization of virtual machines under the specified CPU Utilization Threshold (*CUT*). Given a cloud system composed of multiple virtual clusters (*VC*), for example, three *VC*s *A*, *B* and *C* (denoted by VC_A, VC_B and VC_C), the task consolidation strategy within a virtual cluster (for example, VC_A) can be described as follows:

1. The scheduler of VC_A dispatches task t_j to a *VM*. If more than one *VM* is available, an appropriate *VM* is selected based on the best-fit strategy.
2. If there is no *VM* available, and V_i is below the specified CPU utilization threshold, VC_A asks for resource support from other *VC*s, e.g., VC_B or VC_C.
3. If both VC_B and VC_C can provide *VM*s that run below the specified CPU utilization threshold, then Eq. (4) is used to select the *VM* from the *VC* that consumes the least amount of energy when transmitting and executing the task.
4. If none of the *VCs can provide a VM* below the specified CPU utilization threshold, then t_j is assigned to the V_i that consumes the least amount of energy locally (i.e., VC_A).

The task consolidation strategy uses the best-fit strategy to optimize resource utilization. The best-fit strategy achieves this by migrating tasks to whichever VM will most closely approach the target CPU utilization threshold. The CPU utilization threshold depends on hardware architecture and may differ on different cloud systems. Based on the hardware commonly found in data centers and our research model, a 70% CPU utilization threshold is considered an appropriate cutoff point. For the sake of simplicity, we shall use 70% CPU utilization as the default threshold for the rest of the paper.

Let us use an example to clarify the ideas above. Fig. 3 outlines the basic information of a set of tasks. To simplify the presentation, we assume there are three virtual machines in VC_A. The first scenario assumes that a task is assigned to virtual machines locally in VC_A. As shown in Fig. 4, after having five tasks (t_0 to t_4) assigned amongst the virtual machines, the sixth task t_5 can be assigned to either V_0 or V_1. Applying the best-fit strategy, task t_5 is assigned to V_1 because the total CPU utilization of V_1 and t_5 is closer to 70% CPU utilization.

Problems occur when attempting to dispatch and execute task t_6. This is because t_6 needs to utilize 50% of the CPU. Assigning t_6 to any *VM* in VC_A will surpass the 70% CPU utilization threshold. In order to keep CPU utilization under 70%, cluster *A* asks for resource support from other clusters, for example, VC_B and VC_C. As shown in Fig. 5, both VC_B and VC_C have resources available below the desired 70% threshold, thus VC_A needs to identify which of these clusters can save the most amount of energy. To estimate the amount of energy consumed when consolidating a task with resources located on a different cluster, both CPU computation and network transmission is considered. Given a task t_j that is to migrate from VC_P and consolidate with virtual machine V_i that resides in VC_Q, the expected energy consumption is formulated as follows,

Task t_j	Arival Time ($T_{a,j}$)	Processing Time ($T_{p,j}$)	CPU Utilization	Data Size
t_0	0s	50s	30%	150Mb
t_1	10s	20s	30%	75Mb
t_2	12s	35s	40%	20Mb
t_3	15s	15s	30%	150Mb
t_4	20s	30s	60%	250Mb
t_5	30s	25s	30%	110Mb
t_6	35s	10s	50%	210Mb

Fig. 3. A list of tasks.

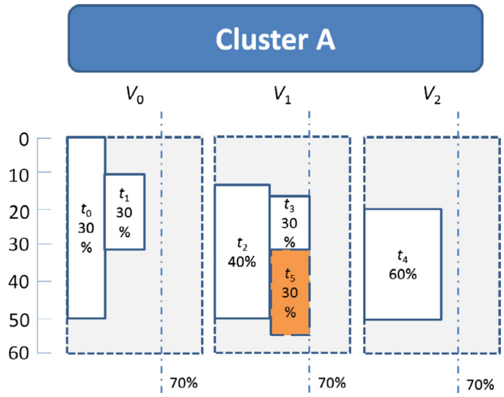

Fig. 4. Assigning t_5 in VC_A.

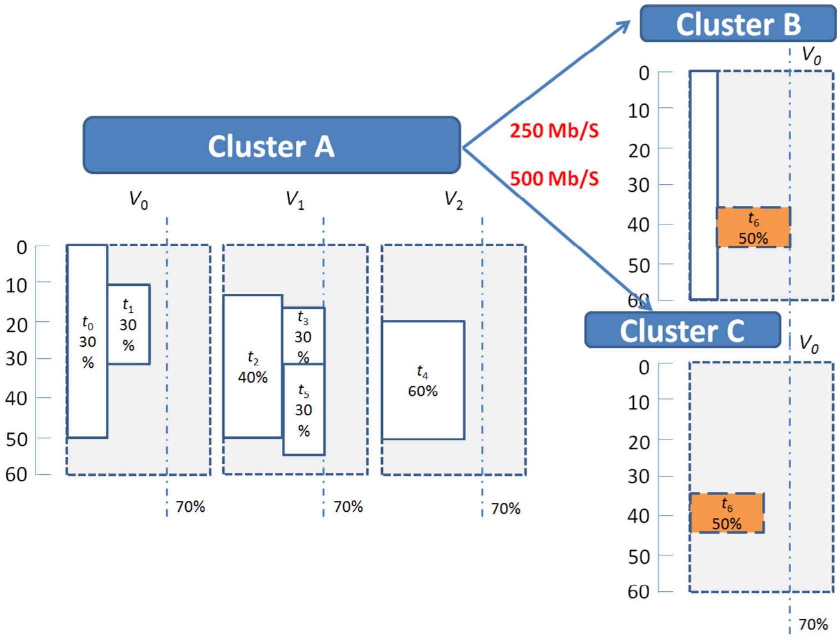

Fig. 5. VC_A asks for resource support when assigning t_6.

$$Cost_{i,j} = \sum_{t=T_{a,j}}^{T_{a,j}+T_{p,j}} E_t(V_i) + \frac{DS_j}{BW_{PQ}} \times 2\beta \text{ W/s} \tag{4}$$

where BW_{PQ} and DS_j represent the bandwidth from VC_P to VC_Q and the size of task t_j's data set, respectively.

According to (4), migrating task t_6 to VC_B and VC_C will consume $(3\beta + \alpha) * 10 + 210/250 * 2\beta$ and $(2\beta + \alpha) * 10 + 210/500 * 2\beta$, respectively. It is easy to see that VC_C will consume less energy, therefore task t_6 is consolidated onto a virtual machine in VC_C. Fig. 5 shows the above scenario.

The last example shows what happens if VC_B and VC_C do not have enough resources if VC_A asks support for t_6. In Fig. 6, CPU utilization of the *VMs* in VC_B and VC_C are higher than or equal to 70%. Thus, VC_A will not seek outside resources due to the extra overheads it would incur. Consequently, it assigns t_6 to local resources even though the *VM* could not conform to 70% CPU utilization. In this example, V_0 is chosen.

A high-level description of the *ETC* algorithm is described as follows.

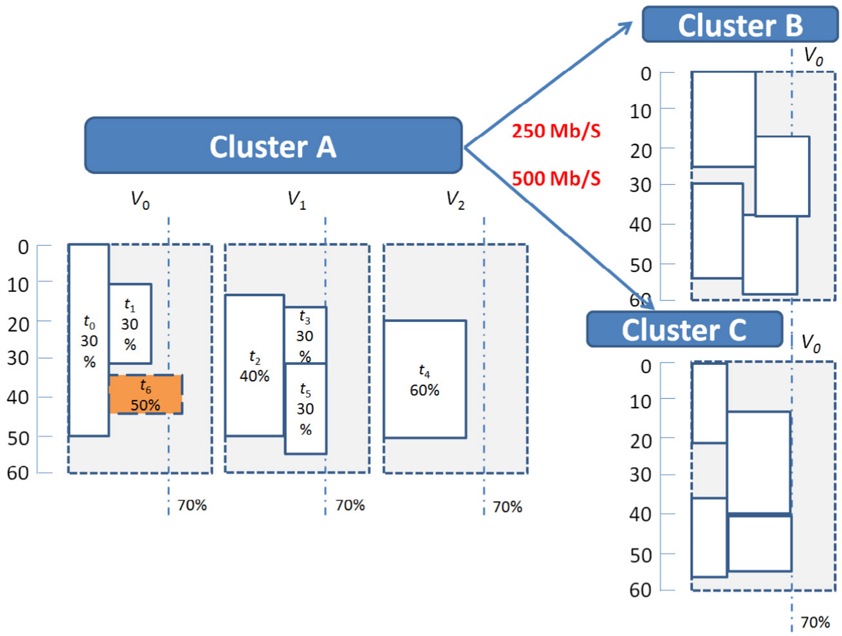

Fig. 6. VC_A assigns t_6 to V_0 without conforming to 70% CPU utility.

Algorithm 1. *ETC*

```
{
  Confirm_CUT (VC_A, t_j)// CPU Utilization Threshold
  //Check the CPU Utilization Threshold for all tasks in the job queue of a virtual cluster VC_A
  If none of the resources in VC_A can execute task t_j without surpassing the CPU Utilization Threshold
  {
    Confirm_CUT (VC_B, t_j)
    Confirm_CUT (VC_C, t_j)
      If more than one resource can execute task t_j without surpassing the CPU Utilization Threshold
        dispatch task t_j to the VC that will consume the least amount of energy.
      Otherwise
        consolidate task t_j to the VM in VC_A that consumes the least amount of energy.
  }
}
```

4. Evaluation of performance

4.1. Experiment configuration

To evaluate the performance of the proposed technique, we implemented the *ETC* method and the *MaxUtil* [10] method. The *MaxUtil* method consolidates tasks and assigns as many tasks as it can to a *VM*. Overall, *MaxUtil* is the same as *ETC* except it has a 100% CPU utilization threshold. The cloud computing model in this section is the same as we described in Fig. 1 but for the purpose of analysis, the numbers of nodes in the virtual clusters are different. In our test, we change the number of nodes in VC_A and use 5, 10 or 15 nodes. This represents a low number of resources (*LR*), a medium number of resources (*MR*) and high number of resources (*HR*), respectively. Both VC_B and VC_C have a medium number of resources (*MR*) in these tests. The virtual clusters are then assigned a different number of tasks. These represent different workloads, with 1000, 2000 and 3000 tasks representing low loading (*LL*), medium loading (*ML*) and high loading (*HL*), respectively. For the sake of clarity, the experimental parameters used in this study are presented in Table 1.

The arrival time of a task occurs between 0 and 9 s. Thus, different workloads represent different task densities. The average processing time, CPU utilization and data size of tasks used are 50 s, 50% and 100 Mb, respectively. In total, 27 cases were used to test the performance of *ETC*. These cases are presented in Table 2. As seen in Table 1, each case is represented by

Table 1
Experimental parameters.

Parameter	Definition	
LL	Low loading	1000 tasks
ML	Medium loading	2000 tasks
HL	High loading	3000 tasks
LR	Low resources	5 nodes
MR	Medium resources	10 nodes
HR	High resources	15 nodes
$\delta 1$	Workload of local cluster VC_A *L*: low loading *M*: medium loading *H*: high loading	
#	Number of nodes used by local cluster VC_A Eg: 5, 10 or 15 nodes.	
$\delta 2$	Workload of VC_B and VC_C *L*: low loading *M*: medium loading *H*: high loading	
$(\delta 1, \#, \delta 2)$	A tuple that defines an experiment's parameters	

Table 2
Cases with different loading and resource.

VC_A	VC_B and VC_C		
	LL *MR*	ML	HL
LL			
LR(5)	(L,5,L)	(L,5,M)	(L,5,H)
MR(10)	(L,10,L)	(L,10,M)	(L,10,H)
HR(15)	(L,15,L)	(L,15,M)	(L,15,H)
ML			
LR(5)	(M,5,L)	(M,5,M)	(M,5,H)
MR(10)	(M,10,L)	(M,10,M)	(M,10,H)
HR(15)	(M,15,L)	(M,15,M)	(M,15,H)
HL			
LR(5)	(H,5,L)	(H,5,M)	(H,5,H)
MR(10)	(H,10,L)	(H,10,M)	(H,10,H)
HR(15)	(H,15,L)	(H,15,M)	(H,15,H)

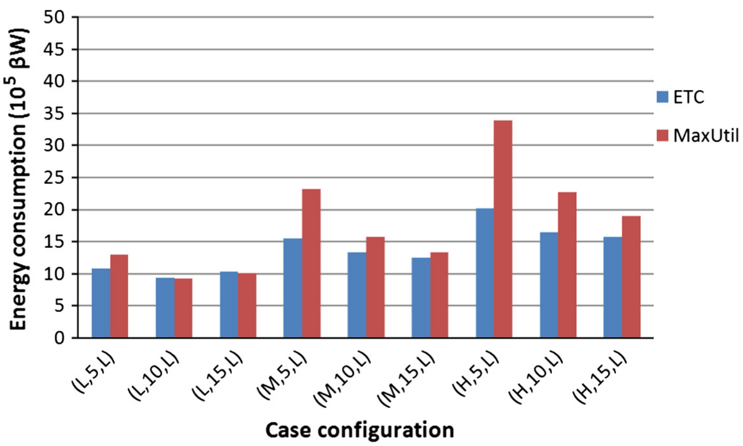

Fig. 7. The results while VC_B and VC_C have low loading.

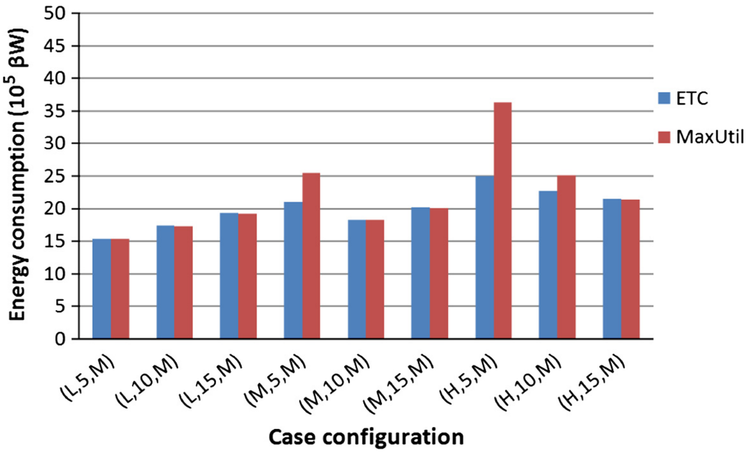

Fig. 8. The results while VC_B and VC_C have medium loading.

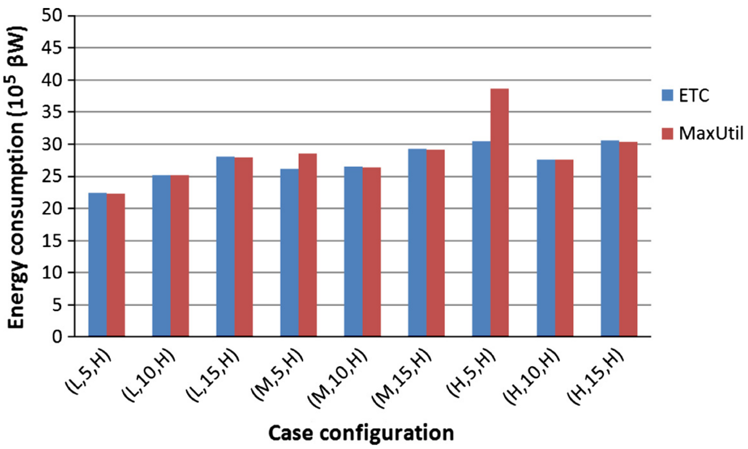

Fig. 9. The results while VC_B and VC_C have high loading.

$(\delta 1, \#, \delta 2)$, where $\delta 1$ is the workload of the local cluster (i.e., VC_A), # is the number of nodes available on local cluster (VC_A) and $\delta 2$ is the workload of VC_B and VC_C. For example, $(L, 5, L)$ represents low loading for VC_A, five nodes in VC_A and low loading for VC_B and VC_C. The following three figures represent the results of different workloads on VC_B and VC_C with various workloads and resource levels on VC_A. Because the energy consumed by tasks when transmitting for resource support is slight, it is omitted without affecting the simulation results. The simulation results show the energy consumption that occurred when executing tasks, and provides us data with which to weigh the pros and cons of *ETC*.

As mentioned in the last section, a *VM* V_i may consume α W/s energy in the idle state, and may consume between β and 5β W/s depending on which of the five levels of CPU utilization is active when executing tasks. Referring to previous research [12], α is set to 7β in our evaluation.

4.2. Experiment results

Fig. 7 shows the results obtained when the workload of neighbor resources (VC_B and VC_C) have a low loading. This scenario encourages VC_A to migrate tasks to neighboring clusters. For example, in the case of $(L, 5, L)$, VC_A has very limited resources (i.e., only 5), which results in VC_A asking for resource support from VC_B and VC_C in order to reduce energy consumption. For the cases of $(L, 10, L)$ and $(L, 15, L)$, VC_A has enough resources. Therefore, the performance of *ETC* and *MaxUtil* is similar because VC_A does not ask for any resources. In most cases, *ETC* performs better because the workload on VC_A is relatively high compared to its neighboring clusters (VC_A has a medium and high workload, while VC_B and VC_C have a low workload), which can incur a large amount of task consolidation by neighboring clusters. This shows the benefits of the *ETC* strategy.

Table 3
Cases of VC_A with extremely high loading.

VC_A	VC_B and VC_C		
	LL *MR*	*ML*	*HL*
EHL			
LR(5)	$(E,5,L)$	$(E,5,M)$	$(E,5,H)$
MR(10)	$(E,10,L)$	$(E,10,M)$	$(E,10,H)$
HR(15)	$(E,15,L)$	$(E,15,M)$	$(E,5,H)$

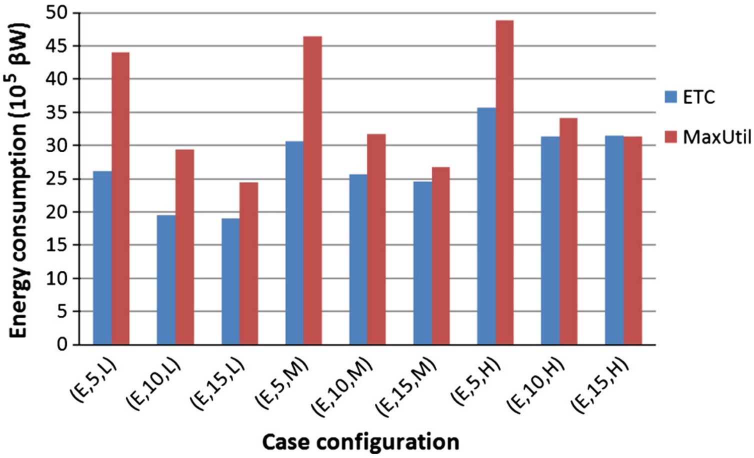

Fig. 10. The results while VC_A has extremely high loading.

Fig. 8 shows the results when neighboring resources (VC_B and VC_C) have a medium workload. In this scenario, VC_A has its tasks migrate to neighboring clusters only if its workload is higher than its neighbor's workload. Consequently, *ETC* and *MaxUtil* exhibit similar levels of performance, except for the three cases, $(M,5,M)$, $(H,5,M)$ and $(H,10,M)$. For $(H,5,M)$ and $(H,10,M)$, the workload of VC_A is higher than VC_B and VC_C. In both these cases, task consolidation can occur often. However, when VC_A has a higher workload than its neighboring clusters $(H,15,M)$, *ETC* does not report significant improvement. This is because VC_A already has enough resources (e.g., 15) to handle tasks in its job queue. In the case $(M,5,M)$, VC_A needs to perform task consolidation due to it having very limited resources.

Fig. 9 shows the results when both neighboring resources (VC_B and VC_C) have a high workload. In this scenario, VC_A is unlikely to migrate tasks to neighboring clusters because VC_A will not ask for resource support unless VC_A resources are lower than its neighbors, for example, $(M,5,H)$ and $(H,5,H)$.

Figs. 7–9 show that *ETC* benefits *VC*s when it is used to manage power consumption. Furthermore, it can produce significant optimizations when the corresponding *VC* either has a relatively high workload or has relatively low resources when compared to external resources. To validate this observation, we used test cases with an extremely high work loading, consisting of 4000 additional tasks. Such cases are denoted as having extremely high loading (*EHL*) as is shown in Table 3. Fig. 10 shows the results of these cases. Extremely high loading forces VC_A to ask resource support form VC_B and VC_C. Obviously, *ETC* outperforms *MaxUtil* in most cases. Overall, *ETC* is able to achieve up to 17% improvement over *MaxUtil*.

5. Related work

Energy consumption is an important issue in many fields of research. Both consumers and industries want their products to use less power in order to reduce energy costs. As systems get larger and more complex they typically consume more energy. This problem extends to networks and consequently extends to cloud computing. Since data centers contain large clusters of computers, any reduction in energy expenditure can result in large economical savings. For this reason and others, there has been much research done on how to reduce energy consumption and on how to reduce energy consumption within a network.

Gunaratne et al. [7] proposed a power management method to reduce wasted energy. In their work, they look at how to recover wasted energy on PCs and network links that are fully powered up even when they are idle. In their research, they noted a lack of power management on PCs, switches and internet protocols. The lack of energy awareness by these devices was wasting energy needlessly and was costing the United States billions of dollars per year. In their paper, they suggest disabling unused paths when routing, and to add power management features to PCs. Furthermore, they suggest ways to improve power management so that network administrators were not inclined to turn off power saving features due to the inconvenience they caused. These factors support the need for energy aware techniques such as *ETC*.

Chanclou et al. [5] compared and evaluated point-to-point and point-to-multipoint optical access solutions. They presented a view on the evolution of broadband optical access networks and pointed out some aspects on simulating network development of network environments in the future. A network-based model [3] was presented to estimate the power consumption of core, metro and access networks which are the three main parts of a standard Internet Service Provider's network. Baliga et al. [2] presented how power consumption in networks can be used to analyze optical and wireless access networks, e.g., passive optical network (PON), fiber to the node (FTTN), point-to-point (PtP) optical systems and worldwide interoperability for microwave access (WiMAX). In the results, PON and PtP were the best solutions in terms of energy consumption. Tucker et al. [19] proposed a model to estimate the energy consumption for IP networks. The hybrid fiber-to-the-node was not recommended due to its high energy consumption. Lange and Gladisch [9] compared the energy consumption of FTTH network based on passive optical networks (PON), active optical networks (AON) and point-to-point (PtP) networks. The results showed that FTTH networks based on PON had better performance in terms of energy consumption. These papers present details on energy consumption of network systems and provide a basis on which to model hardware by *ETC*.

Vasić and Kostić [20] tried to reduce energy consumption on the Internet by moving a greater percentage of links to a sleep state. They proposed *Energy-Aware Traffic engineering* (*EATe*) to improve upon traditional works that overlooked energy consumption. *EATe* was able to successfully move links to a sleep state and handle changes in traffic load without affecting traffic rates. The authors note that the popularity of bandwidth-intensive services, such as streaming and video on-demand energy have increased energy consumption on the internet and the rising popularity of cloud computing services increases this further. *EATe* and *ETC* are both energy aware techniques designed to reduce network energy use. However, *EATe* is designed to work with network traffic and hardware holistically, while *ETC* is designed specifically for the cloud computing environment.

Aliza et al. [1] listed the components of hardware and software that affect energy consumption. In the list, the CPU was the most important component as it had a major impact on energy consumption within a computer system. Lien et al. [12] collected power consumption and CPU utilization data, and used them to produce a model that reflected their relationship. They designed a virtual instrumentation software module to measure the power consumption of a streaming media server in real time. Using their design, users can estimate energy consumption accurately and easily without additional hardware. The authors explain the relationship between CPU utilization and energy consumption does not increase linearly. These observations are exploited by the *ETC* method.

A Scalable Multiple Server (SMS) architecture [11] design for resource management in a server center was presented for better performance and power consumption. Linear and exponential power models were proposed for estimating power consumption of different states. The SMS architecture improved energy consumption by 16.9% in the experiment results. In practice, achieving maximum power capacity for a warehouse-size computing system is difficult since power consumption varies mainly with computing activity. Fan et al. [6] studied power usage of thousands of servers (cluster level) and found a noticeable power gap even in well-tuned applications. They argued that both peak performance and activity range should be taken into account when considering power efficiency. Rivoire et al. [15] compared five high-level full-system power models from a laptop to a server and concluded that models based on OS utilization and CPU performance were accurate. Meisner et al. [13] proposed a two-state energy-conservation approach, *PowerNap*, to simplify the complex power-performance states of systems. Furthermore, they demonstrate a *Redundant Array for Inexpensive Load Sharing* (*RAILS*) that in conjunction with *PowerNap* improves average power consumption by 74%. Beri et al. [4] studied energy-saving research for the management of integrated systems. They identified several factors that might occur which could have an impact on energy-saving strategies for cloud computing environments. Power consumption on a physical machine can be measured in modern server hardware. However, power consumption of a *VM* cannot be measured directly by hardware. Kansal et al. [8] proposed *Joulemeter* to solve this problem. There were several coefficients in their formula that changed from time to time. Thus these coefficients had to be adjusted according to the threshold they defined.

Nathuji et al. [14] proposed the *VirtualPower* approach for power management as a way to support virtual machines (*VM*s) and their virtualized resources. The experimental evaluations showed 34% improvement on power consumption. Torres et al. [18] proposed a consolidation strategy for a data center by combining memory compression and request discrimination techniques. They evaluated the proposed strategy with a representative workload scenario and a real workload. Srikantaiah et al. [17] studied the inter-relationships between performance degradation, energy consumption, CPU utilization and Disk utilization. They transformed the consolidation problem into a bin-packing problem and found which server had better performance and lower power consumption for each request. Song et al. [16] proposed a utility analytic model for Internet-oriented servers, which can provide the upper bound of physical server required for QoS and estimate the power and utility. The experiments showed improvements on power and CPU resource utilization without performance degradation. Lee and Zomaya [10] proposed two energy-conscious task consolidation heuristics, *ECTC* and *MaxUtil*, to reduce energy consumption without performance degradation for a cloud environment with homogeneous resources in terms of computing capability and capacity. Both *MaxUtil* and *ETC* consolidate tasks in order to reduce energy consumption. *MaxUtil* tries to minimize power consumption by maximizing utilization and does so by assigning as many tasks as it can to a *VM. ETC* differs from *MaxUtil* by taking into account the increased rate of energy consumption that occurs as CPU utilization increases.

6. Conclusion and future work

Clouds typically consist of multiple resources. These resources can be distributed, heterogeneous and virtualized. A high priority for a cloud computing system is the maximization of profits. The amount of energy consumed by these systems has a big influence on how profitable they are. Much of the literature shows energy consumption and resource utilization in clouds are highly coupled. They show that task consolidation is an effective technique to increase resource utilization and that increased resource utilization can help reduce energy consumption. In this paper, an *energy-aware task consolidation* (*ETC*) technique is presented to minimize energy consumption. Considering the architecture of most cloud systems, a default CPU utilization threshold of 70% is used to demonstrate task consolidation management amongst virtual clusters. Although the idle state of virtual machines and network transmission are assumed to be a constant ratio (7 and 2) of basic energy consumption unit in this study, these values can be adjusted on different cloud systems in order to get better performance from the *ETC* method. The simulation results show that *ETC* can significantly reduce power consumption when managing task consolidation for cloud systems. *ETC* has up to 17% improvement over a recent work [10] that reduces energy consumption by maximizing resource utilization.

ETC is designed to work in a data center for VC and VMs that reside on the same rack or on racks where network bandwidth is relatively constant. In future work it would be desirable for system to work between data centers and take into account fluctuations in network bandwidth. Furthermore, the current implementation uses a manual approach to designate the CPU utilization threshold. Instead of having an administrator set the threshold, it would be better if ETC could adapt automatically to its environment. We leave these tasks for future work.

References

[1] Muhammad H. Alizai, Georg Kunz, Olaf Landsiedel, Klaus Wehrle, Power to a first class metric in network simulations, in: Proceedings of the Workshop on Energy Aware Systems and Methods, February, 2010.
[2] Jayant Baliga, Robert Ayre, Wayne V. Sorin, Kerry Hinton, Rodney S. Tucker, Energy consumption in access networks, in: Proceedings of Optical Fiber Communication/National Fiber Optic Engineers, February, 2008, pp. 1–3.
[3] Jayant Baliga, Kerry Hinton, Rodney S. Tucker, Energy consumption of the internet, in: Proceedings of Conference on Optical Internet and Australian Conference on Optical Fiber Technology, 2007, pp. 1–3.
[4] Andreas Beri, Erol Gelenbe, Marco Di Girolamo, Giovanni Giuliani, Hermann De Meer, Minh Quan Dang, Kostas Pentikousis, Energy-efficient cloud computing, The Computer Journal 53 (7) (2010) 1045–1051.
[5] Philippe Chanclou, Stéphane Gosselin, Juan Fernández Palacios, Victor López Alvarez, Evi Zouganeli, Overview of the optical broadband access evolution: a joint article by operators in the IST network of excellence e-Photon/One, IEEE Communications Magazine 44 (8) (2006) 29–35.
[6] Xiaobo Fan, Wolf-Dietrich Weber, Luiz Andre Barroso, Power provisioning for a warehouse-sized computer, in: Proceedings of the 34th Annual International Symposium on Computer Architecture, 2007, pp. 13–23.
[7] Chamara Gunaratne, Ken Christensen, Bruce Nordman, Managing energy consumption costs in desktop pcs and lan switches with proxying split TCP connections, and scaling of link speed, International Journal of Network Management 15 (5) (2005).
[8] Aman Kansal, Feng Zhao, Jie Liu, Nupur Kothari, Arka A. Bhattacharya, Virtual machine power metering and provisioning, in: Proceedings of the 1st ACM Symposium on Cloud computing, 2010, pp. 39–40.
[9] Christoph Lange, Andreas Gladisch, On the energy consumption of FTTH access networks, in: Proceedings of Optical Fiber Communication, March 2009, pp. 1–3.
[10] Young Choon Lee, Albert Y. Zomaya, Energy efficient utilization of resources in cloud computing systems, The Journal of Supercomputing (2010) 1–13.
[11] Chia-Hung Lien, Ying-Wen Bai, Ming-Bo Lin, Chia-Yi Chang, Ming-Yuan Tsai, Web server power estimation, modeling and management, in: Proceedings of 14th IEEE International Conference on Networks, vol. 2, 2006, pp. 1-6.
[12] Chia-Hung Lien, Ming Fong Liu, Ying-Wen Bai, Chi Hsiung Lin, Ming-Bo Lin, Measurement by the software design for the power consumption of streaming media servers, in: Proceedings of the IEEE Instrumentation and Measurement Technology Conference, 2006, pp. 1597–1602.
[13] David Meisner, Brian T. Gold, Thomas F. Wenisch, PowerNap: eliminating server idle power, in: Proceeding of the 14th International Conference on Architectural Support for Programming Languages and Operating Systems, vol. 44, no. 3, 2009, pp. 205–216.
[14] Ripal Nathuji, Karsten Schwan, VirtualPower: coordinated power management in virtualized enterprise systems, in: Proceedings of Twenty-First ACM SIGOPS Symposium on Operating Systems Principles, 2007, pp. 265–278.
[15] Suzanne Rivoire, Parthasarathy Ranganathan, Christos Kozyrakis, A comparison of high-level full-system power models, in: Proceedings of the 2008 Conference on Power Aware Computing and Systems, 2008.
[16] Ying Song, Yanwei Zhang, Yuzhong Sun, Weisong Shi, Utility analysis for internet-oriented server consolidation in VM-based data centers, in: Proceedings of IEEE International Conference on Cluster Computing, 2009, pp. 1–10.
[17] Shekhar Srikantaiah, Aman Kansal, Feng Zhao, Energy Aware consolidation for cloud computing, in: Proceedings of the 2008 Conference on Power Aware Computing and Systems, 2008.
[18] Jordi Torres, David Carrera, Kevin Hogan, Ricard Gavaldà, Vicenç Beltran, Nicolás Poggi, Reducing wasted resources to help achieve green data centers, in: Proceedings of IEEE International Symposium on Parallel and Distributed Processing, 2008, pp. 1–8.
[19] Rodney S. Tucker, Jayant Baliga, Robert Ayre, Kerry Hinton, Wayne V. Sorin, Energy consumption in IP networks, in: Proceeding of Optical Communication, 2008, pp. 1.
[20] Nedeljko Vasić, Dejan Kostić, Energy-aware traffic engineering, in: Proceedings of the 1st International Conference on Energy-Efficient Computing and Networking, 2010, pp. 169–178.