PAPER

Honeyguide: A VM Migration-Aware Network Topology for Saving Energy Consumption in Data Center Networks

Hiroki SHIRAYANAGI[†a)], **Hiroshi YAMADA**[†], *Nonmembers*, *and* **Kenji KONO**[†], *Member*

SUMMARY Current network elements consume 10–20% of the total power in data centers. Today's network elements are not energy-proportional and consume a constant amount of energy* regardless of the amount of traffic. Thus, turning off unused network switches is the most efficient way of reducing the energy consumption of data center networks. This paper presents *Honeyguide*, an energy optimizer for data center networks that not only turns off inactive switches but also increases the number of inactive switches for better energy-efficiency. To this end, Honeyguide combines two techniques: 1) virtual machine (VM) and traffic consolidation, and 2) a slight extension to the existing tree-based topologies. Honeyguide has the following advantages. The VM consolidation, which is gracefully combined with traffic consolidation, can handle severe requirements on fault tolerance. It can be introduced into existing data centers without replacing the already-deployed tree-based topologies. Our simulation results demonstrate that Honeyguide can reduce the energy consumption of network elements better than the conventional VM migration schemes, and the savings are up to 7.8% in a fat tree with $k = 12$.
key words: *energy savings, virtualization, network switches, migration*

1. Introduction

Data centers are constructed from a lot of network elements. A large number of network elements consume a considerable amount of total power; they consume 10–20% of the total power in data centers [1]. The total power consumed by the networking elements in data centers in 2006 in the U.S. alone was 3 billion kWh and has been rising year by year [2].

Unfortunately, today's network elements are not energy-proportional. Ideally, network switches consume energy proportional to the amount of traffic, but today's switches consume a constant amount of energy regardless of the amount of traffic. More energy-efficient network elements have been actively developed to address this problem [3]–[6]. However, the maximum efficiency comes from a combination of the improved elements and their sophisticated management.

Our approach to reducing the energy consumption of networks is to reduce the number of active (turned on) networking switches. If there are network switches through which no traffic is flowing, we can turn them off to save energy. In our approach, which is called *Honeyguide*, we not only turn off inactive switches, but also try to increase the number of inactive switches. We combine two techniques to increase the number of inactive switches: 1) virtual machine (VM) and traffic consolidation, and 2) a slight extension to the existing tree-based topologies. Honeyguide has the following characteristics:

- *Leveraging VM migration:* Honeyguide uses VM migration to increase the number of unused network switches. VM migration is extensively used to reduce the power consumption of physical machines, but it is not used for reducing the power consumption of network elements. The existing approaches such as ElasticTree [7] use traffic consolidation to reduce the number of active network elements but do not make use of VM migration.
- *Fault Tolerance:* The network elements in data centers are usually redundant to tolerate any unexpected failures of the network switches or cable cuts. Our design for Honeyguide retains the fault tolerance of the original tree-based topologies, even though Honeyguide turns off any unused (inactive) switches. It also satisfies the requirements imposed on the VM replacement for fault tolerance. For example, Honeyguide can assign two VMs in separate racks to avoid rack-level failures.
- *Easy Deployment:* Honeyguide is carefully designed to ensure easy deployment on existing data center networks. Although Honeyguide uses a slightly extended version of existing tree-based topologies, it can be introduced by adding some additional links, especially when a data center uses 2N tree topologies such as a fat tree.

This paper presents the simulation results from workloads modeled after real data center workloads to demonstrate the power-saving effect of Honeyguide. Our simulation is extensively conducted by changing the number of VMs and the size of a fat tree topology to cover various data centers configurations. The results suggest that Honeyguide can reduce energy consumption more than conventional VM migration schemes and its savings are up to 7.8% in a fat tree with $k = 12$.

The remainder of this paper is structured as follows. Section 2 describes the key considerations for data centers. Section 3 presents Honeyguide and Sect. 4 presents our simulation results. Section 5 discusses the work related to ours, and Sect. 6 concludes the paper.

Manuscript received September 11, 2012.
Manuscript revised March 21, 2013.
[†]The authors are with Keio University, Yokohama-shi, 223–8522 Japan.
a) E-mail: yanagi@sslab.ics.keio.ac.jp
DOI: 10.1587/transinf.E96.D.2055

*We use power and energy interchangeably in this paper.

2. Data Center Networks

2.1 Power Consumption of Network Switches

Current network switches are not energy-proportional. Network switches consume a constant amount of energy regardless of the amount of network traffic. Ideally, network switches should consume energy proportional to the amount of network traffic; the energy consumption should be almost 0 when the switches are idle, and increase in proportion to the amount of network traffic. In current network switches, constant energy is needed regardless of the traffic volume because fans, switch chips, and transceivers waste a constant amount of energy even when the network switches are idle. Current research and development efforts are devoted to making network switches energy-proportional. Even if energy-proportional switches are created, it would cost a lot to replace existing switches with such high-end ones since a large number of network elements are deployed in data centers.

Figure 1 shows the energy consumption of a CISCO Catalyst 3750G network switch, equipped with 48 ports [8]. The energy consumption is measured in two ways: 1) Idle: no traffic flows through the switch and 2) Busy: all links are fully used. The number of connected links are changed from 2 to 6 in both cases. As can be seen from the graph, the switch consumes 90 W even when no links are connected (indicated by 0 connected links). Furthermore, there is no difference in the power consumption levels in Idle and Busy; the network switch is not energy-proportional at all.

Our approach to reducing the energy consumption of networks is to reduce the number of active (turned on) networking switches. If there are some network switches through which no traffic is flowing, we can turn them off to reduce the energy consumption. In our approach, we not only turn off the inactive switches, but also try to *increase* the number of inactive switches. We combine two techniques in our approach to increase the number of inactive network switches: 1) virtual machine (VM) and traffic consolidation, and 2) a bypass link (a slight extension to the existing tree-based network topologies). Section 2.2 briefly explains the traffic and virtual machine consolidation, and the technical issues related to them. Section 2.4 outlines our approach, which combines two techniques to reduce power consumption of network elements.

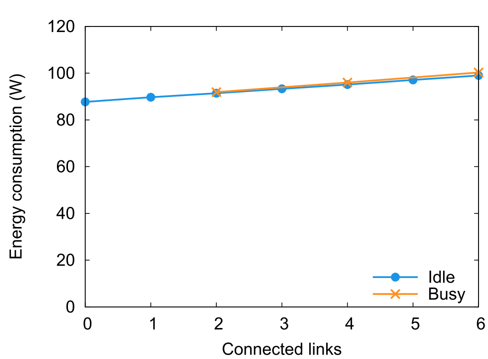


Fig. 1 Energy consumption of network switch.

2.2 Virtual Machine and Traffic Consolidation

Data centers are typically provisioned for peak workloads. Network links and switches are redundant to tolerate any unexpected failure of the switches or cable cuts, and physical machines are also redundant to tolerate any unexpected spikes in workloads and machine failures. As a result, data centers run well below capacity most of the time and the workloads can be satisfied by a subset of the network links, switches, and physical machines. In this paper, we assume that virtualization is used in our target data centers. This assumption is not a critical restriction on our approach since 85% of data centers use server virtualization [9] and more and more data centers are doing so.

Figure 2 shows an example of a data center network. It consists of three layers: 1) core, 2) aggregation, and 3) edge, and is called a $k = 4$ fat tree. K-ary fat tree [10] is a variant of the traditional fat tree. It has k pods. Each pod contains two layers (aggregation and edge) each of which has $k/2$ switches with at least k ports. The half of the ports of the lower-level (edge) switches are connected to $k/2$ physical machines, and the remaining half of the ports are connected to the upper (aggregation) layer switches. Core layer contains $(k/2)^2$ switches with at least k ports. Each switches has one port connected to each of k pods. The i^{th} port of any core switch is connected to pod i such that consecutive ports in the aggregation layer of each pod switch are connected to core switches on $(k/2)$ strides.

This fat tree has redundant links in every layer of the tree. Every pair of machines in adjacent layers has two links in a fat tree. Therefore, even if one link or switch failed, the entire network would continue to correctly work.

Aside from this redundancy in network elements, physical machines are also redundant. Figure 3 (a) shows an example of the VM placement in data center networks. In this figure, there are four physical machines (PM1 to PM4). PM1, PM2, and PM3 are hosting one VM each, while PM4 is not hosting any VMs since it is provisioned for the peak workload. In this case, all the edge switches (C and D) must stay active because there are active physical machines under them.

Virtual machine (VM) migration can be leveraged to

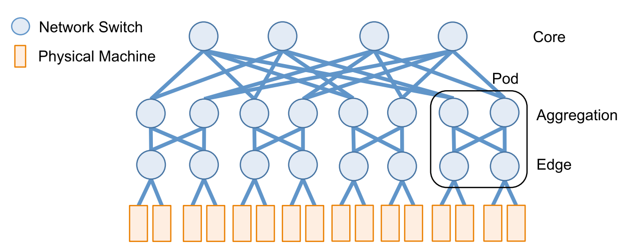
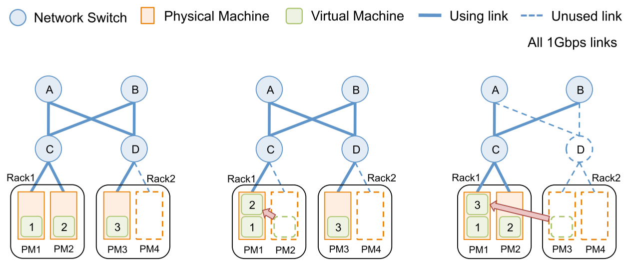


Fig. 2 Fat tree topology.



(a) Original network topology (b) Network topology after network consolidation (c) Network topology after VM and network consolidation

Fig. 3 Energy savings by consolidation.

reduce the power consumption of data center networks. VM migration is often used to increase the number of *inactive* physical machines so that we can reduce the power consumption of physical machines. In this paper, we use VM migration to save power in the network elements as well as physical machines.

When determining the VM placement, Honeyguide takes the traffic flows and network topology into account. Honeyguide consolidates the VMs so that the network elements can be turned off. Suppose that PM1 has enough resources to host either VM2 or VM3. If no special attention is paid to reducing the power consumption of the network elements, VM2 is on a par with VM3; PM2 or PM3 is turned off after the migration. However, if reducing the power consumption of the network elements is taken into account, VM2 is not on a par with VM3. If VM2 is migrated to PM1, network switch D cannot be turned off (shown in Fig. 3 (b)). If VM3 is migrated to PM1, we can turn off network switch D since there is no active machines under switch D (shown in Fig. 3 (c)).

Reducing the power consumption of the network elements through VM migration is possible because VM migration automatically consolidates the network traffic. In the above example, the VM migration from PM3 to PM1 consolidates the network flow in switch D with that in switch C. Honeyguide carefully examines the network flows so that the consolidated flow cannot exceed the capacity of the links to consolidate the network traffic. In the above example, the consolidated traffic does not exceed the capacity. When the VM migration is completed, Honeyguide directs the network switches to appropriately change the routes.

2.3 Redundancy Requirements

We must take the *redundancy requirements* into consideration to consolidate the network traffic and VMs. Data centers usually impose various constraints on the placement of VMs for fault tolerance. For example, a data center prepares two or more replica VMs. When one VM fails, the other VM takes over it and continues to service. To be tolerant to various levels of failures, the location in which a replica VM is placed must be carefully chosen. To be tolerant to a physical machine failure, the replica VM must be placed on a machine different from the original VM. To be tolerant to the edge switch failures, the replica VM must be connected to another edge switch. To be tolerant to power unit failures, the replica VM must be placed in a pod different from the original VM that does not share the power supply unit. For example, if there is requirement for VM3 to be placed in a pod different from VM1, such as in Fig. 3 (c), we cannot migrate VM3 to physical machine 1.

Aside from the placement of VMs, there are redundancy requirements on network switches. Although some network switches are turned off to reduce the power consumption, they must be turned on again to adjust to the ever-changing amount of traffic and unexpected network switch failures. Since it takes longer to turn on a network switch than to change the traffic routing, there is requirement for unused switches to remain turned on to quickly respond to workload changes and network failures. For example, we cannot turn off network switch B in Fig. 3 (b) if there is requirement for two routes to always be available to reach VM1 and VM2.

2.4 Design Principles of Honeyguide

In this paper, we present Honeyguide, which is an energy-efficient network topology that combines two techniques: 1) VM and traffic consolidation, and 2) bypass links. As mentioned earlier, we assume virtualized data centers, but this would not be a critical restriction to Honeyguide because most data centers use virtualization. Honeyguide combines traffic and VM consolidation for better energy efficiency in data center networks. However, as discussed in Sect. 2.3, the redundancy requirement hinders the flexible placement of VMs and turning off unused network switches. To mitigate this problem, Honeyguide introduces *bypass* links that contribute to increasing the number of network switches that can be turned off under severe redundancy requirements.

Honeyguide is novel in two ways. First, to the best of our knowledge, there is no work that leverages the VM migration to save the energy of *network* elements. VM migration is used in various contexts such as load balancing, fault tolerance, and power saving for *physical machines* but no work has been done to save network energy. For example, Hermeniner et al. [11] propose globally sub-optimal VM placement to increase the number of unused physical machines. Verma et al. [12] propose a VM placement that is aware of the power and migration costs. Energy-efficient networks such as VL2 [13], the one proposed in [10], and the flattened butterfly [14] do not leverage VM migration to save energy.

Second, Honeyguide is carefully designed to re-use the existing network topology already deployed in data centers. It costs a lot to dramatically change an existing topology because a large number of network switches are intricately connected, and the network routing and management systems are usually tuned for the topology. To ease the deployment of Honeyguide into existing data centers, Honeyguide only slightly extends the existing tree-based topologies with bypass links.

3. Honeyguide

3.1 Overview

Honeyguide is a slight extension of the traditional tree-based topologies that allows us to consolidate traffic and VMs under various redundancy requirements. It extends a traditional tree-based topology by adding *bypass* links between the upper-tier switches and physical machines. Figure 4 illustrates an example of Honeyguide that has bypass links between the upper-tier network switches and physical machines. The machines to which bypass links are added are called a *honey machine*. The bypass links increase the flexibility of the VM placement and allow us to consolidate the traffic and VMs under severe redundancy requirements. The feasibility of adding bypass links to existing data center networks is discussed later in this section.

To better understand the effect of bypass links, try to consider the situation shown in Fig. 4. In this figure, there is a replica VM that must be placed in another rack different from the original VM. The replica VM in physical machine 4 cannot be migrated to physical machine 1 or 2 under this requirement. Consequently, network switch D cannot be turned off if there is no bypass link. By adding bypass links, Honeyguide allows us to turn off network switch D under the same requirement. Even if network switch D is turned off, the replica VM can communicate with the original VM through network switch B.

The example shown in Fig. 4 is oversimplified for explanatory purposes. When there is no requirement on VM placement, Honeyguide tries to consolidate the VMs on one physical machine to a degree at which no performance penalty is imposed. Then, it tries to gather other VMs in the same rack to increase the number of unused network switches. If there are requirements on VM placement, Honeyguide tries to make use of the bypass links. If one VM cannot be migrated to another rack, Honeyguide tries to gather all the VMs in the same rack on the honey machine and turn off the edge network switch. The details of Honeyguide are described in Sect. 3.2.

Note that the network redundancy is not decreased in Honeyguide even if an edge switch is turned off. In the example shown in Fig. 4, when no bypass links are added, the original and replica VMs cannot communicate with each

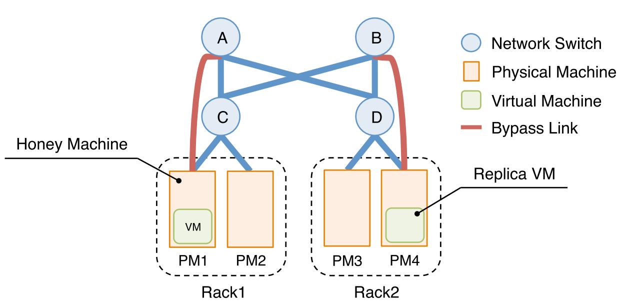


Fig. 4 Honeyguide overview.

other if either network switch C or D is down. After the bypass links are added, they can communicate with each other unless either network switch B or C is down. If the failure rates of network switches B and D are the same, the overall rate of failure is unchanged. Since the upper-tier switches are usually more expensive than the edge ones, the overall rate of failure may be decreased if we use bypass links.

Adding bypass links is not expected to cause serious administrative problems. Recent network switches have a lot of ports and unused ports in the upper-tier switches still remain, as pointed out in [15] and [16]. The more ports a network switch has, the cheaper the cost per port becomes. So, the upper-tier switches in data centers tend to have a lot of ports, some of which are not used. Even if there are no redundant ports in the upper-tier switches, it is not difficult to deploy Honeyguide since we only have to replace the upper-tier switches. Replacing the switches costs less than replacing the entire topology of data centers. For example, all the cabling must be changed to deploy Flattened Butterfly [14].

Honeyguide is not specific to k-ary fat-tree. In this paper, we have chosen k-ary fat-tree only for explanatory purposes because there will be many unused ports in k-ary fat-tree. Honeyguide can be applied to tree-based topologies if bypass links can be prepared. If there is a link that bypasses intermediate switches to connect an upper layer switch directly to a physical machine, we can regard the directly connected machine as a honey machine. If a honey machine can consolidate all VMs hosted by the machines under the bypassed network switch, Honeyguide can turn off the bypassed switch. The rate of power reduction is mainly controlled by the number of bypassed switches and the number of physical machines under each bypassed switch. If these conditions are the same, we can get almost the same simulation results even if the original fat-tree is used.

3.2 Consolidating VMs and Traffic

Honeyguide is a network-wide energy optimizer that continuously monitors the data center traffic conditions and VM workloads. It determines the placement of virtual machines that satisfy the redundancy requirements and capacity limits of physical machines. Then, it chooses the set of network elements that must stay active and powers down as many unneeded links and switches as possible.

Honeyguide can use a variety of methods to decide the VM placement and which subset of links and switches to use. One important thing to be taken into account is that honey machines should be preferred as target machines of migration. If all the VMs under an edge switch are consolidated on a honey machine, we can turn off that edge switch because the honey machine can bypass it.

In this paper, Honeyguide is applied to fat tree topologies and uses the first-fit algorithm to determine the VM placement. The first-fit algorithm is widely used for VM consolidation in data centers. The hardware configuration of each physical machine — its CPU, network, disk, and memory characteristics — is assumed to be known to Honeyguide. Honeyguide uses a monitoring engine in a hypervisor that gathers the processor, network, and memory swap statistics. The monitoring engines periodically inform these statistics to the Honeyguide optimizer. This Honeyguide design is compatible with Sandpiper [17], a framework for migrating VMs.

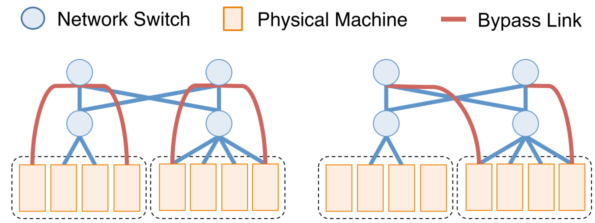
Using these statistics, Honeyguide determines which virtual machine to migrate to which physical machine. First, Honeyguide chooses a physical machine from which the virtual machines are to be off-loaded, and then selects another physical machine to which these virtual machines are to be migrated to. The *original* first-fit algorithm selects the heaviest-loaded physical machine as a source and the lightest-loaded physical machine as the destination. Next, the heaviest-loaded VM in the source is migrated to the destination, if the destination has sufficient capacity to host the incoming VM. This process is repeated until there is no overloaded physical machines. Due to the space limitation, refer to [18] for the details on the first-fit algorithm.

Honeyguide extends the original first-fit algorithm to accommodate honey machines. When selecting a destination machine, Honeyguide chooses a honey machine whenever it is possible. When choosing a destination machine, honey machines are checked beforehand to determine they can host the migrating VM. If there are two or more honey machines, they are checked based on the first-fit algorithm. When a source machine is selected, honey machines are not treated specially. If the load of honey machines are high, the VMs on the honey machines can be migrated to other machines. By doing this, honey machines host as many VMs as possible, leaving less VMs in machines other than the honey machines. As a result, we can increase the possibility that all the machines besides a honey machine become empty and the edge network switches can be turned off thanks to the bypass links.

Honeyguide also extends the original first-fit to take the redundancy requirements into account. When a destination machine is chosen, Honeyguide checks if it violates the redundancy requirement to migrating a VM to the destination. If the requirement is violated, Honeyguide skips this VM and checks if another VM can be migrated to the destination. After the VM placement is determined, the Honeyguide optimizer directs each hypervisor to migrate VMs. Then, all the edge switches with no active machines except for the honey machines are turned off by Honeyguide.

3.3 Over-Subscription

Over-subscription is a well-known technique to utilize network switches effectively. In over-subscribed networks, a network switch is connected to more physical machines than the original tree topology. For example, 2:1 over-subscription means that a network switch is connected to twice the number of physical machines than usual. In the fat tree topology with $k = 4$, while there are usually 2 physical machines under an edge network switch in Fig. 2. An edge network switch is connected to 4 physical machines under 2:1 over-subscription (Fig. 5 (a)).



(a) Handling over-subscription by adding bypass links
(b) Handling over-subscription by switching bypass links

Fig. 5 Handle over-subscription.

The increased number of physical machines in a rack makes it difficult to turn off the edge network switches in over-subscribed networks because a single honey machine cannot consolidate all the VMs in the rack. For example, in 2:1 over-subscription a honey machine must consolidate twice the number of VMs to turn off an edge network switch.

Over-subscription can be handled easily in Honeyguide. Honeyguide allows us to increase the number of honey machines simply by adding some bypass links. Figure 5 (a) shows an example in which one more bypass link is added.

Even if there is no remaining ports in upper-tier network switches, we can increase the number of honey machines in a rack by *sharing* bypass links connected to other racks such as shown in Fig. 5 (b). Although the total number of honey machines are the same, Honeyguide can consolidate more VMs in over-subscribed racks.

4. Experiments

To show the power saving effect of Honeyguide, we conducted simulation-based experiments using a synthetic workload derived from the surveys of real data center traces [19]–[21]. In this paper, we show how Honeyguide reduces the energy consumption of network elements. To show this, we change the number of VMs and the k-value of a fat tree topology, comparing it with the conventional VM consolidation scheme. It performs the original first fit algorithm to select the VMs to be migrated and their destination based on the resource usage, as described in Sect. 3.2. We migrate VMs under the constraint for hardware fault tolerance that the VMs providing the same service are not placed on the same rack. Each rack in the simulation contains six physical machines.

We calculate the network energy savings of Honeyguide through the simulations in the same way as Heller et al. [7]. The network energy savings are computed as:

$$= 100 - \frac{\text{Network Power Consumption of Honeyguide} \times 100}{\text{Network Power Consumption of the Simple Consolidation}}$$

The numerator is the network energy consumption of Honeyguide, while the denominator represents the network energy consumption of the conventional VM consolidation scheme. The percentage gives us an accurate idea of the

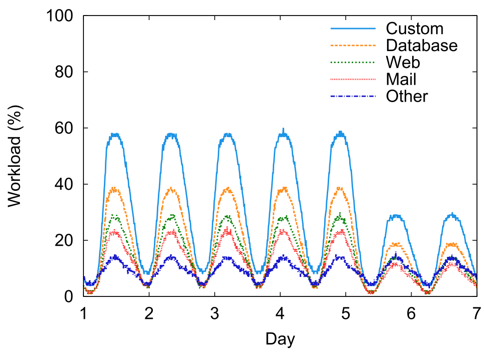


Fig. 6 Experimental workloads.

over-all energy savings by Honeyguide.

The network energy consumption is the sum of the energy consumed by the network switches. The energy consumption of each network switch is computed as (Base power + Port power × The number of active ports) × The total hours. The base power is the power that a network switch consumes during idle periods. The port powers are the powers that are consumed when the links connected to the ports are active. The total hours value is the time for which the network switches are turned on. We set the base power and each port power to 100 W and 2 W respectively, based on our experimental results in which we measured the energy consumption of CISCO Catalyst 3750G [8].

4.1 Experimental Workload

Data centers of various sizes (different k values) are simulated in our experiments. We have changed the scale of workloads according to k (i.e., the total number of physical machines). If the total number of physical machines is larger, the total workload is set to be larger. In all simulated data centers, we assume the average CPU usage of VMs changes as shown in Fig. 6. Note that a single VM runs only one service and the number of VMs is not dynamically changed. By doing this, we can simulate larger workloads in larger data centers since the aggregated workload becomes larger. In our simulation, we have used CPU usage as a representative resource because the network traffic is not the bottleneck in most data centers. According to [22]–[26], the average usage of network is less than 5%, while the average usage of CPU is 10–50%.

We generated a synthetic workload based on the surveys of real data center traces [19]–[21]. The workload is shown in Fig. 6. It is widely known that the workload in data centers almost regularly varies over time and its shape is similar to that of a sine curve; the requests increase during the day time and decrease at night. In addition, guided by the studies in [19]–[21], we prepare workloads for five types of services running on VMs; Custom (C), Database (D), Web (W), Mail (M), and Others (O). Custom is the custom home-brewed applications. Database represents the database servers, Web represents the Web applications, and Mail is for the mail servers. Others includes the other servers such as authentication servers like the LDAP servers.

Table 1 No. of VMs providing same service.

	base	VM-6	VM-12	VM-24	VM-36	VM-72
C	0	6	12	24	36	72
D	0	3	6	12	18	36
W	0	3	6	12	18	36
M	0	0	3	6	9	18
O	0	0	3	6	9	18

We set the scales of each workload based on the surveys [19], [20]. We borrow the Database, Web, and Mail scales from the real-world data center traces [19], [20]. For example, the peak of Database is a 40% usage of the resources and the average is a 5% usage, according to the real traces from DB2 and SQL2000 [19]. The web's peak and average are a 30% and 5% usage of the resources respectively, which is shown in the Apache trace [19]. The mail scale is set by the Hotmail trace, which shows that the peak is 20–30% and the load stays at 5% when the access is not heavy [20]. Although we do not have the scale data for Custom and Others, we set them based on the above values by taking into consideration the ratio of the network trace data [21].

We stress each VM based on this workload, which provides the VMs resource usage. Honeyguide monitors the resource usage of the VMs and migrates them in the first-fit way as described in Sect. 3.2

4.2 Changing Ratio Between Service and Replica VMs

To demonstrate how Honeyguide reduces the network energy consumption, we first simply change the ratio between the service and replica VMs, fixing the total number of running VMs. The service VM is a VM that performs its own service and can be migrated to every rack. The replica VM is a VM that performs the same service a service VM performs. The placement constraint is imposed on them, which means that a replica VM cannot be migrated to the rack where the VMs execute the same service. We set up a fat tree with $k = 12$. This parameter is borrowed from [7]. The fat tree consists of 432 physical machines and 72 racks in total. We ran 864 VMs and ran 432 Custom, 144 Database, 144 Web, 72 Mail, and 72 Other VMs. We did not run replica VMs at first. We refer to this configuration as the base, and then, we varied the ratio between the service and replica VMs, taking into consideration the balance of each service workload. The details of this are listed in Table 1. We also varied the number of honey machines, 12, 24, 36, and 72.

The number of VMs is derived from various literature [11], [17], [25], [27]. The total number of VMs is derived from [11], [17]. In [17], 35 VMs are prepared for 16 physical machines; i.e., the number of VMs is more than twice of physical machines. In [11], the number of VMs is changed from 200 to 400 for 200 physical machines; i.e, the number of VMs is less than the twice of physical VMs. So, we have decided to change the number of VMs around the twice of physical machines. The number of service VMs

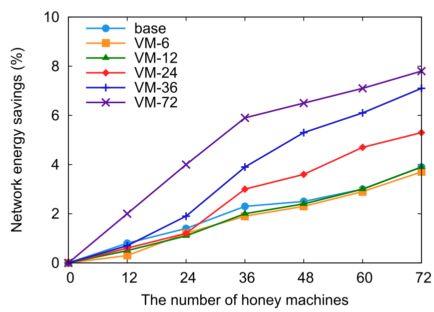


Fig. 7 Power savings of Honeyguide when ratio between service and replica VMs is changed.

is set to be proportional to network traffic reported in [21]. The number of replica VMs is set from 0 to the number of the racks since a replica is usually placed in another rack.

The initial placement of service VMs is determined as follows. Using the first-fit algorithm, Honeyguide determines which physical machine hosts which virtual machine. Honeyguide prepares one list of physical machines (the order is random) and another list of service VMs (the order is random). Honeyguide picks up a service VM from the list and sequentially searches for a physical machine in the list that can host the service VM. Since Honeyguide is based on the first-fit, it places the service VM on the matching physical machine. Thereafter, the placement of service VMs is dynamically changed according to the first-fit-based algorithm shown in Sect. 3.2.

For the placement of honey machines, the fundamental principle is to distribute honey machines uniformly in the pods. If the number of pods is 12 and the number of honey machines is also 12, each pod has one honey machine. If the number of pods is 12 and the number of honey machines is 24, each pod has 2 honey machines. Two honey machines in each pod are uniformly distributed over racks.

The experimental results are shown in Fig. 7. The x-axis represents the number of honey machines, and the y-axis is the network energy savings. The figure shows that Honeyguide better reduces the network energy consumption compared to the conventional VM consolidation scheme. We can also see that Honeyguide reduces energy consumption more when more replica VMs and honey machines are used. The increased number of honey machines gives greater flexibility in the placement of replica VMs. Thus, the more honey machines are, the more we can reduce energy consumption. This trend becomes clearer if the number of replica VMs is increased because the constraint on the replica placement becomes severer if the number of replica VMs increases. The energy savings of Honeyguide are 0.8–3.9% in the base, while the energy savings are 0.5–3.9% in VM-12. Honeyguide saves 2.0–7.8% in network power in VM-72.

4.3 Changing Total Number of VMs

To show the effectiveness of Honeyguide under various VM consolidation degrees, we change the number of service and replica VMs. Due to the limitation of our simulator's scalability, we scale down the k-value and the size of the racks. In particular, the k value is four and each rack contains two physical machines. The topology consists of 20 network switches, 16 physical machines, and 8 racks. We also varied the number of honey machines, 2, 4, 6, and 8.

We first set up the same number of service VMs as physical machines, namely 16 service VMs. The VMs include 8 Custom, 3 DB, 3 Web, 1 Mail, and 1 Other VMs. We refer to this configuration as the base. After simulating our base configuration, we first increase the number of service VMs. We ran two, three, four, and five times the number of each VM. Second, we increase the number of replica VMs by the preparing two, four, six, and eight times number of each VM. Last, we increase both types of VMs. We prepare four configurations: (1) one more VM in each service and each VM has 1 replica, (2) two more VMs in each service and each VM has 3 replicas, (3) three more VMs in each service and each VM has 5 replicas, and (4) four more VMs in each service and each VM has 7 replicas.

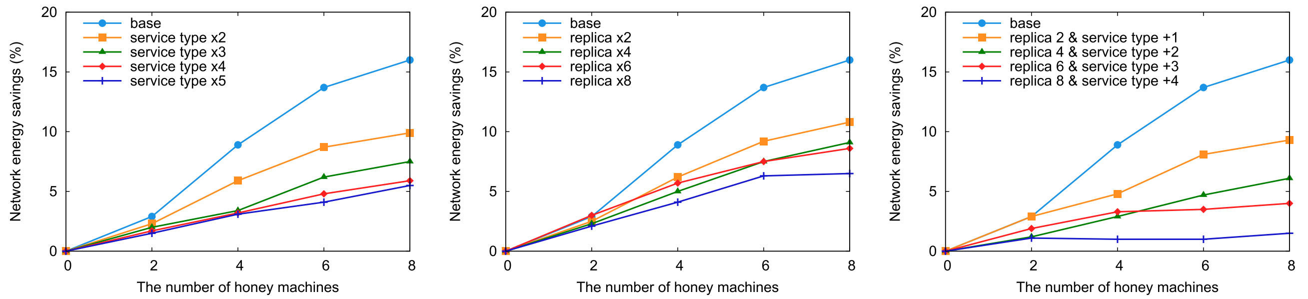
Figure 8 shows the results. The figure shows that Honeyguide can save more network energy than the simple VM consolidation scheme in all cases. When the number of service VMs is five times, Honeyguide's power savings are 1.5–5.5%. Its power savings are 2.1–6.5% when the number of replica VMs is eight times. The power savings of Honeyguide are 1.1–1.5% under the (4) configuration. The results also show that the network savings of Honeyguide are lower as the number of VMs is increased. This is because consolidating the VMs into one physical machine is more difficult when the number of VMs are increased. In fact, Honeyguide is the most effective in the base configuration where the number of VMs is the smallest.

4.4 Changing Network Topology Size

To demonstrate how Honeyguide works under different network topology sizes, we measure the network energy savings, varying the k-value of the fat tree topology. We change the k-value to 16, 20, and 24. The number of running VMs is twice the number of physical machines in each topology. Similar to the experiment described in Sect. 4.2, we change the ratio between the service and replica VMs. We ran $\frac{k^3}{8}$ Custom, $\frac{k^2(k-4)}{16}$ DB, $\frac{k^2(k-4)}{16}$ Web, $\frac{k^2}{4}$ Mail, and $\frac{k^2}{4}$ Other VMs. Table 2 lists the configurations of service and replica VMs we set up. We also varied the number of honey machines, k, $2k$, $3k$, $\frac{k^2}{4}$, and $\frac{k^2}{2}$. We can prepare more honey machines as the topology gets bigger, since a big topology consists of

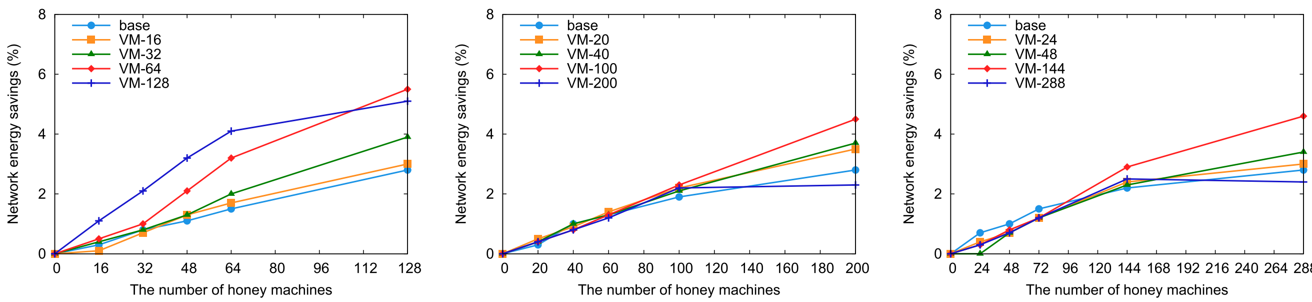
Table 2 No. of VMs providing same service.

	base	VM-k	VM-$2k$	VM-$k^2/4$	VM-$k^2/2$
C	0	k	$2k$	$k^2/4$	$k^2/2$
D	0	$k/2$	k	$k^2/8$	$k^2/4$
W	0	$k/2$	k	$k^2/8$	$k^2/4$
M	0	$k/4$	$k/2$	$k^2/16$	$k^2/8$
O	0	$k/4$	$k/2$	$k^2/16$	$k^2/8$



(a) Service VMs (b) Replica VMs (c) Service and Replica VMs

Fig. 8 Power savings when No. of VMs is increased.



(a) k = 16 (b) k = 20 (c) k = 24

Fig. 9 Power savings when k-value is varied.

many physical machines and network switches.

The experimental results are shown in Fig. 9. We can see that Honeyguide effectively reduces the network power under different topology sizes. Honeyguide reduces the network power consumption by 0.3–7.8% at $k = 12$, while it reduces the power consumption by 0.3–4.6% at $k = 24$. We can also see that Honeyguide reduces the power consumption even more when k is smaller. If k is smaller, the number of physical machines in a rack also becomes smaller. This increases the possibility that a honey machine can consolidate all VMs in a rack to turn off a network switch. When the ratio is $1{:}\frac{k^2}{2}$, the power savings at k = 20 and 24 are worse than the base.

4.5 Over-Subscription

To confirm Honeyguide works well under over-subscription, we simulate 2:1 and 4:1 over-subscription. Each rack contains 12 and 24 physical machines under 2:1 and 4:1 over-subscription, respectively. We run the twice number of VMs as physical machines. Also, we vary the number of honey machines, 12, 24, 36, and 72.

If Honeyguide does not take over-subscription into account, the effect of network energy savings is decreased. Figure 10 shows the result. The higher the over-subscription rate is, the lower the power saving in Honeyguide is. Honeyguide saves 2.0 to 7.8% power without over-subscription, while it does 0.7 to 2.1% and 0 to 0.2% under 2:1 and 4:1 over-subscription, respectively. When the over-subscription rate is higher, edge switches are unlikely to be turned off because a honey machine must consolidate many VMs.

As described in Sect. 3.3, this problem can be mitigated by a slight extension to Honeyguide. To demonstrate Honeyguide can handle over-subscription, we change the placement of honey machines. By sharing some bypass links, we increase the number of honey machines in each rack without changing the total number of honey machines. More concretely, we place a honey machine in each rack, then two honey machines in the half of racks, finally three honey machines in one third of the racks.

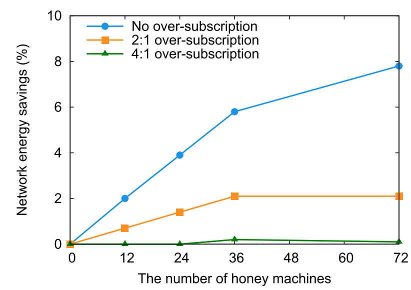
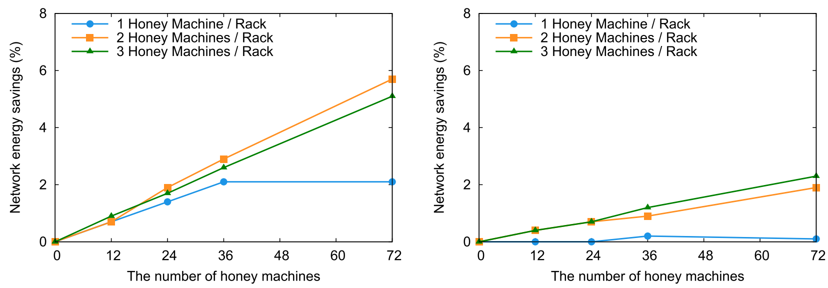


Fig. 10 Network energy savings in over-subscription.

Figure 11 shows the experimental result. From the result, we can see that our extension saves network power even under over-subscription. By adding more honey machines in each rack, the network power saving is higher because the VMs can be easily consolidated to honey machines. For example, Honeyguide can save 0.7 to 2.1% under 2:1 over-subscription. By increasing bypass links, Honeyguide can save 0.7 to 5.7% and 0.9 to 5.1% in network power with 2 honey machines and 3 honey machines in racks, respectively. Moreover, Honeyguide can save 0 to 0.2% under 4:1 over-subscription. Honeyguide can save 0.4 to 1.9% and 0.4 to 2.3% in network power under 4:1 over-subscription by



(a) 2:1 over-subscription (b) 4:1 over-subscription

Fig. 11 Network energy savings in over-subscription when the number of honey machines in a rack is changed.

increasing the number of honey machines in a rack.

Allocating too many honey machines has bad side-effects since the number of racks which contain no honey machines increases. The power savings of 3 honey machines under 2:1 over-subscription are less than those of 1 or 2 honey machines when the total number of honey machines is 24, 36, and 72. This is because the total number of racks containing honey machines is smaller than the other cases. This makes it difficult to turn off some network switches to which no honey machines are connected.

5. Related Work

Some work discusses a new network topology and the existing network topology modifications to save network power. Abts et al. [23] argue that a flattened butterfly topology [14], which is a multi-dimensional direct network, is more power efficient than the other commonly proposed topologies. However, to deploy a flattened butterfly topology, we have to change all the cabling of the existing network topologies. At worst, we have to reconstruct a data center. On the other hand, Honeyguide can be introduced by adding some additional links in a tree-based topology.

ElasticTree [7] saves energy in data center networks at the expense of network redundancy. It continuously monitors the data center traffic conditions using intelligent switches such as OpenFlow, and chooses the set of network elements including the links and switches that must stay active to meet the performance and fault tolerance goals; the unchosen elements are turned off for power saving. Honeyguide does not decrease the network redundancy for network power saving. This approach can also be used complementary to ours if the network redundancy can be decreased.

Several researches focus on VM migration to reduce the power consumption in data centers. pMapper [12] is a VM placement controller that dynamically migrates VMs to minimize the power consumption while meeting the performance guarantees. It models and speculates the migration time, the VM performance after the migration, and the power usage of the physical machines. And then it performs a first fit algorithm to consolidate the VMs. Entropy [11] consolidates the VMs based on the constraint programming, which allows it to better find the VM placement than those found by heuristics based on the local optimization. The migration scheme proposed by Meng et al. [25] moves to the same or close physical machine VMs that communicate with each other to improve the scalability of data center networks. Wang et al. [28] propose an online packing algorithm that allows us to consolidate VMs under dynamic network bandwidth requirements. These works focus on reducing the number of active physical machines to reduce the energy consumption, while ours focuses on network power saving.

Some techniques for controlling the network switches to reduce their power consumption have been explored. Guranatne et al. [3] studied the necessary policies to control the adaptive link rate mechanism that reduces the energy consumption of Ethernet links by adaptively varying the link data rate in response to the utilization. Nedevschi et al. [5] proposed novel schemes to put the network elements to sleep during idle times and adapt the rate of network operations to the offered workload. These techniques are complementary to Honeyguide to save even more network power.

6. Conclusion

This paper presented *Honeyguide*, an energy-efficient network topology for reducing the energy consumption in data center networks under severe redundancy requirements. Honeyguide combines two techniques: 1) VM and traffic consolidation, and 2) bypass links. Our simulation-based experiments show that Honeyguide can better reduce the amount of network power consumption compared with the conventional VM migration scheme, and its savings are up to 7.8% in a fat tree with $k = 12$. One of our future directions is to apply the Honeyguide philosophy to other network topologies such as mesh and torus. Honeyguide is a slight extension of the tree-based network topologies. We believe that the two techniques used in Honeyguide can be effective at reducing save the energy consumption of networks in other topologies.

Acknowledgements

This work is partially supported by “R&D for Construction of a Cloud Infrastructure that supports wide-area disaster (Environment-Related Network Signaling Technology)” project of Ministry of Internal Affairs and Communication

(MIC) of Japan.

References

[1] A. Greenberg, J. Hamilton, D.A. Maltz, and P. Patel, "The cost of a cloud: research problems in data center networks," SIGCOMM Comput. Commun. Rev., pp.68–73, 2008.
[2] R. Brown, E. Masanet, B. Nordman, B. Tschudi, A. Shehabi, J. Stanley, J. Koomey, D. Sartor, and P. Chan, "Report to congress on server and data center energy efficiency: Public law 109-431," Tech. Rep., Ernest Orlando Lawrence Berkeley National Laboratory, Berkeley, CA (US), Aug. 2007.
[3] C. Gunaratne, K. Christensen, B. Nordman, and S. Suen, "Reducing the energy consumption of ethernet with adaptive link rate (alr)," IEEE Trans. Comput., vol.57, pp.448–461, 2008.
[4] M. Gupta and S. Singh, "Using low-power modes for energy conservation in ethernet lans," 26th IEEE INFOCOM, pp.2451–2455, 2007.
[5] S. Nedevschi, L. Popa, G. Iannaccone, S. Ratnasamy, and D. Wetherall, "Reducing network energy consumption via sleeping and rate-adaptation," Proc. 5th USENIX Symp. on Networked Systems Design and Implementation, pp.323–336, 2008.
[6] "IEEE 802.3az."
[7] B. Heller, S. Seetharaman, P. Mahadevan, Y. Yiakoumis, P. Sharma, S. Banerjee, and N. McKeown, "Elastictree: saving energy in data center networks," Proc. 7th USENIX Symp. on Networked systems design and implementation, pp.249–264, 2010.
[8] "Cisco catalyst 3750 series switches."
[9] CIO Research, "Virtualization in the enterprise survey: Your virtualized state in 2008," 2008.
[10] A.V. Mohammad Al-Fares, A. Loukissas, "A scalable, commodity data center network architecture," Proc. ACM SIGCOMM, pp.63–74, 2008.
[11] F. Hermenier, X. Lorca, J.M. Menaud, G. Muller, and J. Lawall, "Entropy: A consolidation manager for clusters," Proc. 2009 ACM int'l conf. on Virtual execution environments, pp.41–50, 2009.
[12] A. Verma, P. Ahuja, and A. Neogi, "pmapper: power and migration cost aware application placement in virtualized systems," Proc. 9th ACM/IFIP/USENIX Int'l Conf. on Middleware, pp.243–264, 2008.
[13] A. Greenberg, J.R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D.A. Maltz, P. Patel, and S. Sengupta, "Vl2: A scalable and flexible data center network," Proc. ACM SIGCOMM, pp.51–62, 2009.
[14] J. Kim, W.J. Dally, and D. Abts, "Flattened butterfly: a cost-efficient topology for high-radix networks," Proc. 34th annual int'l symp. on Computer architecture, pp.126–137, 2007.
[15] J. Kim, W.J. Dally, B. Towles, and A.K. Gupta, "Microarchitecture of a high-radix router," Proc. 32nd annual int'l symp. on Computer Architecture, pp.420–431, 2005.
[16] S. Scott, D. Abts, J. Kim, and W.J. Dally, "The blackwidow high-radix clos network," Proc. 33rd annual int'l symp. on Computer Architecture, pp.16–28, 2006.
[17] T. Wood, P. Shenoy, A. Venkataramani, and M. Yousif, "Black-box and gray-box strategies for virtual machine migration," Proc. 4th USENIX Symp. on Networked systems design and implementation, pp.229–242, 2007.
[18] M. Yue, "A simple proof of the inequality ffd $(l) \leq 11/9$ opt $(l) + 1, \forall l$ for the ffd bin-packing algorithm," Acta Mathematicae Applicatae Sinica (English Series), pp.321–331, 1991.
[19] V. Soundararajan and J.M. Anderson, "The impact of management operations on the virtualized datacenter," Proc. 37th annual int'l symp. on Computer architecture, pp.326–337, 2010.
[20] M. Lin, A. Wierman, L. Andrew, and E. Thereska, "Dynamic right-sizing for power-proportional data centers," 30th IEEE INFOCOM, pp.1098–1106, 2011.
[21] T. Benson, A. Akella, and D.A. Maltz, "Network traffic characteristics of data centers in the wild," Proc. 10th annual conference on Internet measurement, pp.267–280, 2010.
[22] W. Vogels, "Beyond server consolidation," Queue, vol.6, no.1, pp.20–26, Jan. 2008.
[23] D. Abts, M.R. Marty, P.M. Wells, P. Klausler, and H. Liu, "Energy proportional datacenter networks," Proc. 37th annual int'l symp. on Computer architecture, pp.338–347, 2010.
[24] D. Gupta, S. Lee, M. Vrable, S. Savage, A.C. Snoeren, G. Varghese, G.M. Voelker, and A. Vahdat, "Difference engine: harnessing memory redundancy in virtual machines," Commun. ACM, vol.53, no.10, pp.85–93, Oct. 2010.
[25] X. Meng, V. Pappas, and L. Zhang, "Improving the scalability of data center networks with traffic-aware virtual machine placement," 29th IEEE INFOCOM, pp.1–9, 2010.
[26] VMware Inc., "The role of memory in vmware esx server 3," 2006.
[27] S. Kumar, V. Talwar, V. Kumar, P. Ranganathan, and K. Schwan, "vmanage: loosely coupled platform and virtualization management in data centers," Proceedings of the 6th international conference on Autonomic computing, pp.127–136, 2009.
[28] M. Wang, X. Meng, and L. Zhang, "Consolidating virtual machines with dynamic bandwidth demand in data centers," 30th IEEE INFOCOM, pp.71–75, 2011.

Hiroki Shirayanagi was born in 1988. He received his B.E. degrees from Keio University in 2011. He is currently a M.E. candidate in School of Science for Open and Environmental Systems at Keio University. His research interests include virtualization and cloud computing.

Hiroshi Yamada received his B.E. and M.E. degrees from the University of Electro-communications in 2004 and 2006, respectively. He received his Ph.D. degree from Keio University in 2009. He is currently an associate professor of the Division of Advanced Information Technology and Computer Science at Tokyo University of Agriculture and Technology. His research interests include operating systems, virtualization, dependable systems, and cloud computing. He is a member of ACM, USENIX and IEEE/CS.

Kenji Kono received the BSc degree in 1993, MSc degree in 1995, and Ph.D. degree in 2000, all in computer science from the University of Tokyo. He is an associate professor of the Department of Information and Computer Science at Keio University. His research interests include operating systems, system software, and Internet security. He is a member of the IEEE/CS, ACM and USENIX.