Evaluation of Effect of Network Energy Consumption in Load Distribution across Data Centers

Harumasa Tada[1], Makoto Imase[2], and Masayuki Murata[2]

[1] Faculty of Education, Kyoto University of Education
[2] Graduate School of Information Science and Technology, Osaka University

Abstract. Recently, the consumption of a considerable amount of energy by data centers has become a serious problem, and there are many researches aiming at the reduction of this energy consumption. However, previous researches intend to reduce only the energy consumed inside data centers. To the best of our knowledge, there are few researches on load distributuion that focus on the network energy consumption arising from the communication across data centers. In this study, we consider the energy consumption of the network as well as that of the data centers in the request distribution across geographically distributed data centers. By using various conditions, we calculate the overall energy consumption of two request distribution policies—one respects the network energy consumption, and the other does not. By comparing these two policies, we examine the condition under which the network energy consumption is worth considering.

Keywords: Cloud Computing, Electricity Cost, Optimization, Simulated Annealing.

1 Introduction

Recently, cloud computing has become popular, and the demand of data centers is increasing rapidly. On the other hand, modern data centers consume considerable amount of energy because of the performance improvement of the servers. The reduction in the electricity cost is one of the great concerns for data centers.

There are many researches on performance control and load distribution intended to reduce a data center's electricity cost. The basic approaches to this cost reduction include turning off redundant servers or using the dynamic voltage/frequency scaling (DVFS) of CPUs, according to the measured current load or the estimated future load [1–5]. These appraches are integrated with the load distribution to the servers [6–8]. Other researches focus on the energy consumption of disks or the memory of servers [9–12]. Moreover, recent researches on load distribution focus on the energy consumption of the cooling equipment in data centers [13–15].

These researches are aimed at reducing the amount of energy consumed in a data center. On the other hand, large organizations such as Google and Yahoo!

I. Tomkos et al. (Eds.): BROADNETS 2010, LNICST 66, pp. 501–517, 2012.

operate multiple data centers. These centers are geographically distributed for the purpose of the provision of service to worldwide customers or the improvement of service availability upon system failure. Data transfer among such data centers involves the energy consumption of network devices such as routers or switches. To the best of our knowledge, there are few researches on load distribution intended to reduce the network energy consumption across data centers.

However, the amount of communication across geographically distributed data centers seems to increase rapidly. For example, Google has many data centers around the world and is performing large-scale distributed processing on self-constructed systems [16–18]. Moreover, many companies including Yahoo! are using Hadoop, the opensource implementation of a large-scale distributed file system. The use of such systems that are premised on data distribution is expected to grow rapidly, and therefore, the network traffic across data centers will increase in the future. Moreover, we consider that in order to reduce the energy cost of data centers, the task transfer to remote data centers has to be carried out frequently. Recently, data centers are often located in the countryside for the reduction of the energy cost. For example, some data centers are located in the cold district in order to reduce the energy consumed for cooling [19]. Other data centers are located near the power plant to reduce the power transmission cost [20]. Moreover, one of the reasons why container data centers [21] are attracting attention is that they can move easily according to the variation in the operation cost, including the energy cost. In general, such data centers are located far away from the place of the computing demand. Therefore, tasks need to be transferred via a network.

There is a research on request distribution intended to reduce the energy cost for geographically distributed multiple data centers. Le *et al.* [22] proposed a method for reducing the electricity cost consumed by servers or the cooling equipments in data centers. Their method exploits the difference in the electricity prices of areas where data centers are located and the variation in the electricity price with respect to time. However, Le's method does not consider the network energy consumption when requests are transferred to data centers. Therefore, it is possible that requests are forwarded to distant data centers whose electricity cost is low, and the overall electricity cost increases.

In this study, we consider the energy consumption of the network as well as that of the data centers in the request distribution across geographically distributed data centers. By using various conditions, we calculate the overall energy consumption of two request distribution policies—one respects the network energy consumption, and the other does not. By comparing these two policies, we examine the condition under which the network energy consumption is worth considering.

The rest of this paper is organized as follows: In section 2, we mention the related works. In section 3, we explain the problem discussed in this paper and the request distribution policies. In section, 4, we compare the two policies—one respects the network energy consumption, and the other does not. We conclude the paper in section 5.

2 Related Works

There have been many researches to reduce the energy consumption of data centers, and various methods have been proposed. These methods control the energy consumption of a data center according to the center's load. The ultimate goal is to make the energy consumption of a data center proportional to its load.

Early researches proposed schemes that turn off redundant servers or use the dynamic voltage/frequency scaling (DVFS) function of CPUs in order to reduce the energy consumption while preserving the throughput [1, 2]. Pinheiro *et al.* [1] proposed a method for changing the number of servers that are turned on (active servers) according to the current load. Elnozahy *et al.* [2] integrated the control of the operating voltage of the CPU and the method proposed in [1]. The drawback of the on/off scheme is the coarse granularity of the control of the energy consumption (in the unit of servers) and the effect of the DVFS is limited to the energy consumption of CPUs.

The following researches consider the service level agreement (SLA) or quality of service (QoS) and propose methods that aim at ensuring that the processing time of the requests meets the deadline as well as preserving the throughput [3–5]. Sharma *et al.* [3] proposed a method that for controlling the frequency of the CPU in order to maintain synthetic utilization that is defined to process the requests before the deadline. Rusu *et al.* [4] proposed a method for controlling the CPU frequency of each server on the basis of prediction of the processing time made by keeping a track of the processing time of the processed requests. Chen *et al.* [5] proposed two methods and compared them. One determines the CPU frequency of each server by solving the optimization problem whose constraint is the processing time estimated using the queuing model. The other changes the CPU frequency of each server dynamically using the feedback control based on the measured processing time. Although the deadline is essential in most of the real world services, meeting the deadline causes some inefficiency in the energy consumption.

Recent researches consider a relatively detailed model and introduce load distribution to servers [6–8]. Heath *et al.* [6] proposed the request distribution method for heterogeneous data centers, including different types of servers. Their method uses the model that reflects the variations in the characteristics of the CPUs and disks of different servers. Rusu *et al.* [7] proposed a method for heterogeneous data centers. Their method determines the number of active servers and the request distribution to these servers on the basis of the measured value of energy consumption. Chen *et al.* [8] proposed a method for data centers that provide connection-intensive services. Their method determines the number of active servers and the request distribution to these servers in order to reduce the energy consumption while avoiding the service not available (SNA) error and server-initiated disconnection (SID). Elaborate control based on realistic models contributes to the reduction of energy consumption. However, the trade off between its effect and its implementation cost need to be considered.

As other approaches, some methods that focus on the server components other than the CPU are proposed [9–12]. In a Google data center, at its peak time, the

DRAM occupies 30% of the total energy consumption and the disk occupies 10% while the CPU occupies 33% [23]. This fact implies that the methods focusing on memories and disks are promising for reducing the energy consumption of data centers. Gurumurthi *et al.* [9] proposed a method that modulates the disk rotation speed to reduce the energy consumption. Zhu *et al.* [10] proposed a cache replacement algorithm to reduce the energy consumption of disks. Ganesh *et al.* [11] argue that log-structured file systems can extend the time for which the disks stop in order to reduce the energy consumption. Li *et al.* [12] proposed a method that switches the operation modes of memories and disks in order to reduce the energy consumption. In order to make the server energy consumption proportional to the server load, it is essential to consider components other than the CPUs. However, this approach often results in an ad hoc method depending on a particular implementation of servers.

Recent researches have focused on the load distribution to servers in order to reduce the energy consumption by the cooling equipment [13–15]. Since servers that consume a considerable amount of energy produce a large amount of heat, the cooling equipment also requires considerable energy to cool the servers. Moreover, recently, because of a high density implementation of servers, the heat density of data centers is increasing and the energy required for cooling is rapidly increasing. Therefore, it will be important to reduce the energy required for cooling the servers in the future. Moore *et al.* [13] introduced a metric called heat recirculation factor (HRF) which expresses the amount of heat recirculation in a data center, and proposed a load distribution method based on HRF. Bash *et al.* [14] proposed a method that allocates a high load to servers placed in an area where the cooling efficiency is high in a data center in order to reduce the energy consumption for cooling. Tang *et al.* [15] proposed a task scheduling method that allocates tasks to equalize the inlet temperature of servers using a detailed model of the heat recirculation in a data center. Since the cooling is responsible for a significant portion of the energy consumption in today's data center [23], the importance of researches on cooling is increasing.

3 Request Distribution That Respects Network Energy Consumption

3.1 System Model

We extend the model introduced by Le *et al.* [22]. Figure 1 shows our system model. The system includes several front-ends, several data centers, and a single scheduler.

The transfer of a request is as follows: First, a client sends a request to a front-end. The front-end that receives the request selects a data center to forward the request. The data center that receives the request processes it and returns the result to the front-end. Then, the front-end forwards the result to the client.

The front-ends distribute requests to data centers according to certain fractions. These request fractions are updated periodically according to the variation

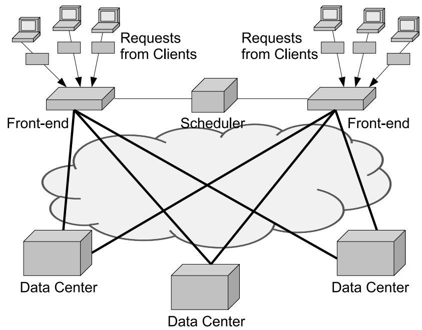

Fig. 1. System Model

in the request arrival rate. The period during which the request fractions are fixed is called an *epoch*. At the beginning of an epoch, the scheduler determines the request fractions for each front-end. In order to calculate the request fractions, the scheduler can use the information about the network and all the data centers. Moreover, the scheduler is periodically notified of the expected request arrival ratio by the front-ends. The details of the calculation of the request fractions are mentioned in the following sections.

3.2 Problem Overview

The problem that we deal with in this study is the determination of the fraction of requests that should be directed to each data center in order to minimize the total energy consumption of the data centers and the network. The energy consumption is measured on a daily basis. That is, the *accounting period* is one day. In the calculation of the request fraction, the following constraints must be taken into account. First, the peak request arrival rate of each data center cannot exceed the capacity of the data center. The *capacity* of a data center means the maximum number of requests that it can process in 1s. Second, the service has a single SLA that should be satisfied. The SLA is expressed as (L, P) which means that ratio of requests that are processed within time L to the total number of requests that arrive in a day should be more than P. The request processing time, which is measured at the front-end, is the time from forwarding a request to receiving its result. This implies that the communication delay between a client and a front-end does not affect the request processing time.

3.3 Assumptions about Energy Consumption

The energy consumption of a data center is the sum of the energy consumption of the servers, local network devices, cooling equipment, and so on. As mentioned in section 2, there are many methods to reduce the energy consumption of data centers. By applying these methods, we can control the energy consumption of the data center according to its load. In this study, we consider the ideal situation in which the energy consumption of a data center is proportional to its load. However, we must take into account that a data center consumes a fixed amount of energy irrespective of its load; we call this energy the *base energy*.

The energy consumption of a network is a sum of the energy consumption of network devices such as routers and switches, and devices for signal transfer such as amplifiers. The reduction of the network energy consumption has attracted attention recently, and there is a pioneering research [24]. However, the relation between the load and the energy consumption is still unclear. In this study, we consider the ideal situation in which the network energy consumption is proportional to the amount of traffic. We do not consider the network base energy because there is traffic that is unrelated to our system on the network.

On the basis of the above observations, we make the following assumptions.

1. A data center consumes a fixed amount of energy irrespective of the number of processed requests.
2. In addition to (1), a data center consumes the energy proportional to the number of processed requests per unit time.
3. The average amount of traffic between a front-end and a data center for requests is proportional to the number of forwarded requests per unit time.
4. The network energy consumption between a front-end and a data center is proportional to the average amount of traffic of (3). That is, we do not consider the energy consumption of the traffic that is unrelated to our system.

3.4 Problem Formulation

We formulate the problem mentioned in section 3.2 as an optimization problem.

Table 1 shows the parameters used in the formulation. In the table, t represents an epoch. Most parameters are the same as those used in [22]. We have replaced the parameters of monetary energy costs ($) with those of energy consumptions (kWh) and introduced the variable h that represents a front-end. Figure 2 depicts an example of the parameters.

$$OverallEC = \sum_t \sum_h \sum_i (f_{hi}(t) LT_h(t) EC_{hi}) \tag{1}$$

$$\forall t \forall h \forall i f_{hi}(t) \geq 0 \tag{2}$$

$$\forall t \forall h \sum_i f_{hi}(t) = 1 \tag{3}$$

$$\forall t \forall i \sum_h (f_{hi}(t) LR_h(t)) \leq LC_i \tag{4}$$

$$\frac{\sum_t \sum_h \sum_i (f_{hi}(t) LT_j(t) CDF_i(L, \mathit{offered}_i(t))}{\sum_t \sum_h LT_h(t)} \geq P \tag{5}$$

Table 1. Parameters (EC stands for "energy consumption")

Symbol	Meaning
$f_{hi}(t)$	Ratio of requests to be forwarded from frontend h to center i
$OverallEC$	Total EC (kWh)
EC_{hi}	Avg. EC (kWh) of a request forwarded from frontend h to center i
a_i	Avg. EC (kWh) of a request at center i
b_{hi}	Avg. network EC (kWh) of a request forwarded from frontend h to center i
LC_i	Load capacity (reqs/s) of center i
$LR_h(t)$	Expected peak service rate (reqs/s) at frontend h
$LT_h(t)$	Expected total service load (#reqs) at frontend h
$\mathit{offered}_i(t)$	$\sum_h (LR_h(t) \times f_{hi}(t))$ (reqs/s)
$CDF_i(L, \mathit{offered}_i)$	Expected ratio of requests that complete within L time, given $\mathit{offered}_i$ load

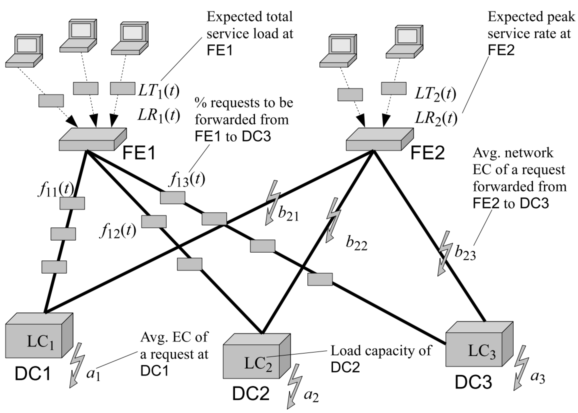

Fig. 2. Example of Parameters

The objective function to minimize is the total energy consumed for an entire day (formula (1)). However, the base energy is not included in formula (1) because it is not changed by the request distribution. If the monetary energy cost is used as the objective function instead of the energy consumption, formula (1) can be easily modified.

Constraints are shown as formulas (2)–(5). Formula (2) shows that the request ratio from any front-end to any data center should not be negative. Formula (3) shows that all requests that arrive at the front-end should be allocated to some data center. Formula (4) shows that the peak request arrival ratio of a data center should not exceed its capacity. Formula (5) shows that the SLA (see section 3.2) should be satisfied, that is, the ratio of requests that are processed within time L to the total number of requests that arrive in a day should be more than P.

3.5 Optimization Policy

In this study, we consider two optimization policies. Policy DC+Net respects the energy consumption of data centers and the network. Policy DConly respects the energy consumption of only the data centers. The difference in these policies is the definition of EC_{hi} included in objective function (formula (1)), which is shown as formulas (6) and (7).

$$\textit{Policy}\ \text{DC+Net:}\ \ EC_{hi} = a_i + b_{hi} \tag{6}$$

$$\textit{Policy}\ \text{DConly:}\ \ EC_{hi} = a_i \tag{7}$$

3.6 Instantiation of Parameters

The parameters of the optimization problem of section 3.4 are given as follows:

On the basis of assumptions stated in section 3.3, it can be inferred that the energy consumption per request at data center i and the network energy consumption per request for a pair of front-end h and data center i, denoted by a_i and b_{hi} respectively, are constants.

The request arrival rate differs for each front-end and varies with time. Therefore, the peak request arrival rate and the total service load of front-end h in epoch t, denoted by $LR_h(t)$ and $LT_h(t)$ respectively, are given as functions of epoch t.

We define $CDF_i(L, \mathit{offered}_i)$ as follows:

$$CDF_i(L, \mathit{offered}_i) = \begin{cases} 1\ (\mathit{offered}_i \leq LC_i) \\ 0\ (\mathit{offered}_i > LC_i) \end{cases} \tag{8}$$

Formula (8) shows that unless the load of data center i exceeds its capacity, all requests arrived that arrive at i are processed before their deadline, including the communication delay. This implies that as long as the constraint shown as formula (4) is satisfied, the constraint shown as formula (5) is also satisfied.

4 Evaluation

In order to examine the condition under which the network energy consumption should be considered, we applied two policies DC+Net and DConly mentioned in section 3.5 to various conditions and compared the energy consumption.

4.1 Methodology

We consider a system having three front-ends $FE1 \sim FE3$ and five data centers $DC1 \sim DC5$.

By using various parameter settings, we calculated and compared the total energy consumption of DC+Net and DConly. The term "the total energy consumption" refers to the total amount of energy consumed by all data centers and the network except for the base energy. The total energy consumption is calculated as follows: For each policy, we solve the optimization problem mentioned in 3.4 to determine the request fractions for all front-ends. Then, we calculate the whole-day energy consumption of each policy according to the request fractions determined above by assuming that the requests arrive at each front-end at the predicted rate. The calculated value is the lower limit of the energy consumption for each policy.

4.2 Solving the Optimization Problem

As in [22], we use the simulated annealing (SA) technique to solve the optimization problem. We use 1-hour epochs and divide a day into 24 epochs.

A state in SA, denoted by s, is defined as follows:

$$s = \{f_{hi}(t) \mid h \in FE, i \in DC, t \in EP\}$$

FE, DC, and EP represent the sets of all front-ends, all data centers, and all epochs, respectively.

Table 2 shows the pseudocode of SA that we used. $exp(x)$ denotes the exponential function. $Energy(s)$ is a function that returns the energy consumption in state s according to formula (1) in section 3.4. $Rand()$ is a function that returns a random actual number from 0 to 1. $Neighbor(s)$ is a function that returns a neighbor of state s. A neighbor of s is calculated as follows: First, $x \in FE$, $y, z \in DC$, and $u \in EP$ are selected randomly such that $f_{xy}(u) \in s$, $f_{xz}(u) \in s$, and $f_{xy}(u) \geq 0.1$. Then, a neighbor of s is made from s by subtracting 0.1 from $f_{xy}(u)$ and adding 0.1 to $f_{xz}(u)$. Table 3 shows the parameter setting of SA.

4.3 Parameter Settings

Request Arrival Rate. The request arrival rate of a front-end varies with time. In this study, we use the request arrival rate used in [22], which is shown in figure 3.

We use the same request arrival rate for three front-ends but shift −3 h, 0, and +6 h, respectively, reflecting different time zones in which front-ends are placed.

Capacity of Data Centers. We set the same capacities (200 reqs/s) for all data centers. Note that the total amount of capacities of five data centers is larger than the sum of peak request arrival rates of three front-ends. This implies that data centers are never overloaded as long as requests are distributed appropriately. This is a necessary condition for obtaining a solution of the optimization problem that meets the constraint of formula (4) in section 3.4.

Table 2. Pseudocode of SA

$s \leftarrow S_{init}$; $e \leftarrow Energy(s)$; $tp \leftarrow TP_{init}$
$s_{best} \leftarrow s$; $e_{best} \leftarrow e$
for $l = 1$ to L
  $s_{new} \leftarrow Neighbor(s)$
  $e_{new} \leftarrow Energy(s_{new})$
  if $e_{new} < e_{best}$ **then**
    $s_{best} \leftarrow s_{new}$; $e_{best} \leftarrow e_{new}$
  if $e_{new} < e$ **then**
    $s \leftarrow s_{new}$; $e \leftarrow e_{new}$
  else if $Rand() < exp((e - e_{new})/tp)$ **then**
    $s \leftarrow s_{new}$; $e \leftarrow e_{new}$
  $tp \leftarrow tp * C$
return s_{best}

Table 3. Parameter Setting of SA

Symbol	Meaning	Value
S_{init}	Initial state	$f_{hi}(t) = 0.2$ for any h, i, t
TP_{init}	Initial temperature	1000
C	Parameter to control temperature drop	0.999
L	Number of iterations	10000

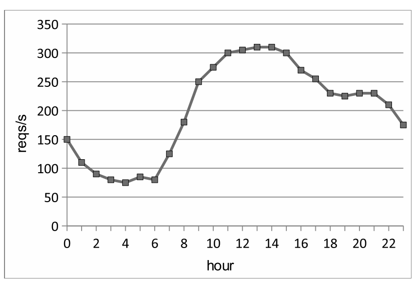

Fig. 3. Request Arrival Rate

Variations of Energy Consumption of Data Centers. In this study, we consider a situation in which there is variation in the energy consumption per request of data centers.

Table 4 shows the result of SPECpower_ssj2008 [25], a benchmark test of the power efficiency of servers. This tells us that there is a great difference between the power efficiency of a new server and that of an old one. In this study, as the metric of the variation in multiple values, we use the ratio of the maximum value and the minimum value, which we call the *max-min ratio*. For example, the max-min ratio in table 4 is approximately 51.2.

Table 4. SPECpower_ssj2008 Results

Server name	ssj_ops/W	Hardware Availability
HP ProLiant DL380 G4	45.2	Sep-2004
HP ProLiant SL2x170z G6	2316	Oct-2009

We used four settings, denoted by $d0 \sim d3$, of the energy consumption of data centers. The ratio of the energy consumption values of the data centers in each setting is shown in table 5. $d0$ is the case in which the energy consumption values of all data centers are the same and the max-min ratio is 1. In the case of $d1$, $d2$, and $d3$, the max-min ratios are 2.6, 5, and 25 respectively. Note that the sum of the values of the five data centers is the same for all settings. This implies that when the requests are distributed uniformly to all data centers, the total energy consumption is the same at any setting.

Table 5. Variation in Data Center Energy Consumption

ID/DC	DC1	DC2	DC3	DC4	DC5
d0	9	9	9	9	9
d1	5	7	9	11	13
d2	3	6	9	12	15
d3	1	2	5	12	25

Variation in Energy Consumption of Network. The energy consumption of the network depends on the location of the front-end and the data center. The energy consumption is low if they are close and high if they are far apart. In order to reflect this fact, we set different values of energy consumption per request for each pair of a front-end and a data center.

In order to determine the variation in the energy consumption of the network, we referred to the variation in the hop counts in a real network. We believe that the hop count is related to the energy consumption of the network since it represents the number of network devices that the packets pass through. Fei *et al.* [26] measured the hop counts from a host at UCLA to various sites on the Internet and reported that the maximum hop count was 27. On the basis of this fact, we used four settings, denoted by $n0 \sim n3$, of the energy consumption of the network. They are shown in table 6. $n0$ is the case with no variation, and its max-min ratio is 1. In the case of $n1$,$n2$, and $n3$, the max-min ratios are 2.6, 5, and 25 respectively.

Ratio of Energy Consumptions of a Data Center and the Network. The result of the comparison between DC+Net and DConly is significantly affected by the ratio of the energy consumption between a data center and the network. Of course, it is desirable that this ratio is close to the ratio in the real world. However, it is very difficult to measure the energy consumption of a

Table 6. Variation in Network Energy Consumption

ID	FE/DC	DC1	DC2	DC3	DC4	DC5
n0	FE1	9	9	9	9	9
	FE2	9	9	9	9	9
	FE3	9	9	9	9	9
n1	FE1	5	7	9	11	13
	FE2	12	8	5	8	12
	FE3	13	11	9	7	5
n2	FE1	3	6	9	12	15
	FE2	13	8	3	8	13
	FE3	15	12	9	6	3
n3	FE1	1	2	5	12	25
	FE2	18	4	1	4	18
	FE3	25	12	5	2	1

real network. Instead, we determined this ratio on the basis of the ratio of the energy consumption between a server and a router. According to the report of Principled Technologies Inc. [27], a server with Intel Xeon 5160 CPU processed 40461 requests/s in WebBench benchmark [28]. Unfortunately, neither the model name nor the energy consumption of the server is found in this report. Instead, we use the specification of Hitachi HA8000/130 server that has the same CPU, whose maximum energy consumption is 611 W. Using these value, we calculated the energy consumption per request to be approximately 15 mJ. On the other hand, Chabarek *et al.* [29] performed experiments using routers and measured the relation of the number of processed packets and the energy consumption. They reported that Cisco GSR 12008 router consumed 770 W when it processed 540,000 packets/s. That is, the energy consumption per packet was 1.4 mJ. Considering that the processing of a single request requires multiple packets and a packet passes multiple routers to reach the destination, we conclude that the difference in the energy consumption per request between a server and the network is not considerable in the case of WWW services. Based on this observation, we used three patterns of the ratio (2:1, 1:1, and 1:2) of the average energy consumption of a data center and the network.

4.4 Results

In this section, we show the calculated total energy consumption of DC+Net and DConly. In the following graphs, the x-axis shows the setting of the energy consumption of the data centers ($d0 \sim d3$) and the network ($n0 \sim n3$), which are shown in tables 5 and 6. The y-axis shows the total energy consumption of each policy, which is divided by the total energy consumption when the requests are distributed uniformly to all data centers, which is denoted by EQUAL.

Figure 4 shows the result when the ratio of the energy consumption of the data centers and the network is 2:1. Unless the data center energy consumption values are the same (setting $d0$), the difference in the total energy consumption

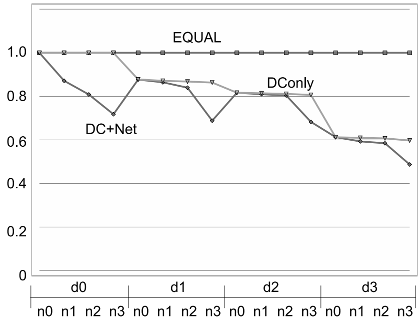

Fig. 4. Data Center:Network=2:1

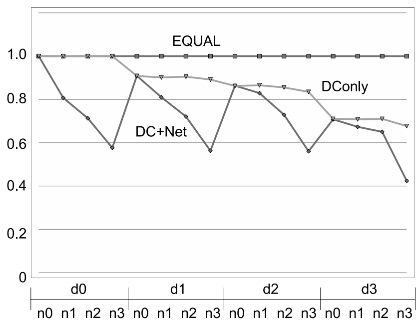

Fig. 5. Data Center:Network=1:1

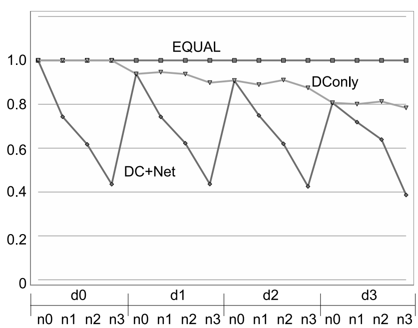

Fig. 6. Data Center:Network=1:2

between DC+Net and DConly is very small in the case of settings $n1$ and $n2$. In this case, since the data center energy consumption is dominant, the effect of considering the network energy consumption is relatively small. When the network energy setting is $n3$, DC+Net outperforms DConly by more than 15% in the case of settings $d1 \sim d3$. Since in the case of setting $n3$, the variation

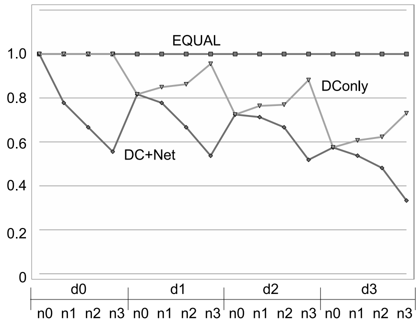

Fig. 7. Data Center:Network=1:1(half load)

in the network energy consumption is significantly large, the network energy consumption is worth considering even if the data center energy consumption is relatively large.

Figure 5 shows the result when the ratio of the energy consumption of the data centers and the network is 1:1. Unless the network energy setting is $n0$, DC+Net outperforms DConly significantly. This result shows that the network energy consumption should be considered in this case.

When the data center setting is $d3$, DC+Net outperforms DConly with more than 10% only in the case of setting $n3$. On the other hand, when the data center setting is $d1$, DC+Net outperforms DConly by more than 10% in the case of settings $n1 \sim n3$. This result shows that the network energy consumption should be considered when the variation in the energy consumption of the data centers is small. This is because when the variation in the energy consumption of the data centers is small, the choice of a data center makes little difference to the energy consumption of the data centers, and therefore, the effect of the network energy consumption becomes relatively large.

By comparing the case in which the network energy consumption is uniform and the data center energy consumption varies widely (setting $d3$ with $n0$) and the case in which the data center energy consumption is uniform and the network energy consumption varies widely (setting $d0$ with $n3$), we observe that DC+Net shows low energy consumption in the latter case. As shown in tables 5 and 6, the max-min ratios of the energy consumption of settings $d3$ and $n3$ is the same. Therefore, considering that the average energy consumption values of the data centers and the network are the same, we conclude that the effect of the variation in the network energy consumption is larger than that of the data center energy consumption. This is because the optimization of the network energy consumption has more options than that of the data center energy consumption. The data center energy consumption of a request is the same irrespective of a front-end that sends the request. In terms of the data center energy consumption, there are some data centers to which no requests should be forwarded. However, when the total load is heavy, some requests have to be forwarded to such data centers and the total energy consumption worsens. On the other hand, the network energy consumption of a request varies depending on the front-end that sends the request. In terms of the network energy consumption, every data center can receive requests from "near" front-ends. Therefore, even when the total load is heavy, the total energy consumption can be improved by the optimization considering the network energy consumption.

Figure 6 shows the result when the ratio of the energy consumption of the data centers and the network is 1:2. Irrespective of the variation in the energy consumption of data centers, in the case of all settings $n0 \sim n3$, DC+Net outperforms DConly by more than 10%. This result implies that it is essential to consider the network energy consumption in this case. In particular, in the case of setting $n3$, DC+Net outperforms DConly by more than 50% in the case of settings $d0 \sim d3$. This result shows that the effect of the network energy consumption is dominant.

Figure 7 shows the result when the ratio of the energy consumption of the data centers and the network is 1:1 and the total load is set to half of the default value. In the case of setting $n0$, in which the network energy consumption is uniform, the total energy consumption values of both DC+Net and DConly improve by 10%–20% as compared to those shown in figure 5. This is because when the load is considerably less than the data centers' capacity, many options are possible with respect to the choice of data centers; this enhances the effect of optimization. On the other hand, in the case of settings $d1 \sim d3$, the total energy consumption of DC+Net at setting $n3$ is 30%–40% lower than at setting $n0$ while that of DConly at setting $n3$ is 15%–30% higher than at setting $n0$. This result shows that when there are many options with respect to the choice of data centers, the penalty of ignoring the network energy consumption during the request distribution becomes relatively large.

On the basis of the above results, we argue that the network energy consumption should be considered in optimization if at least one of the following conditions is satisfied. Note that if some conditions are satisfied simultaneously, the effect of the network energy consumption will increase even if each condition is not sufficiently satisfied.

1. The average energy consumption of the network is equal to or larger than that of the data centers.
2. The max-min ratio of the energy consumption of the network is larger than 20.
3. The max-min ratio of the energy consumption of data centers is smaller than 3.
4. The total load is smaller than 50% of the total capacity of data centers.

5 Conclusions

The reduction of the energy consumption of data centers has been an important research topic. Recently, the energy consumption of the network is attracting considerable attention. In this study, we focused on the request distribution across geographically distributed data centers. We compared two optimization policies—one respects the network energy consumption, and the other does not—and examined the condition under which the network energy consumption is worth considering.

Our evaluations are based on the total energy consumption values calculated in the case of optimization. Since these values are estimated under an ideal condition, they differ from the real energy consumption. In real systems, the policy that is good for optimization does not necessarily show good performance because of unexpected events, e.g., the unexpected increase in the number of requests. We do not consider such unexpected conditions because the scope of this study is the effectiveness of considering the network energy consumption. Nevertheless, robustness, the property to maintain a good performance even under unexpected conditions, is one of the important metrics for request distribution methods. Evaluation in a more realistic environment is our future work.

References

1. Pinheiro, E., Bianchini, R., Carrera, E., Heath, T.: Load balancing and unbalancing for power and performance in cluster-based systems. In: Workshop on Compilers and Operating Systems for Low Power, vol. 180, pp. 182–195 (2001)
2. Elnozahy, E., Kistler, M., Rajamony, R.: Energy-Efficient Server Clusters. In: Falsafi, B., VijayKumar, T.N. (eds.) PACS 2002. LNCS, vol. 2325, pp. 179–196. Springer, Heidelberg (2003)
3. Sharma, V., Thomas, A., Abdelzaher, T., Skadron, K., Lu, Z.: Power-aware QoS management in web servers. In: Proceedings of the 24th IEEE International Real-Time Systems Symposium, p. 63 (2003)
4. Rusu, C., Xu, R., Melhem, R., Mosse, D.: Energy-efficient policies for request-driven soft real-time systems. In: Euromicro Conference on Real-Time Systems, ECRTS 2004 (2004)
5. Chen, Y., Das, A., Qin, W., Sivasubramaniam, A., Wang, Q., Gautam, N.: Managing server energy and operational costs in hosting centers. ACM SIGMETRICS Performance Evaluation Review 33(1), 303–314 (2005)
6. Heath, T., Diniz, B., Carrera, E., et al.: Energy conservation in heterogeneous server clusters. In: Proceedings of the Tenth ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming, p. 195. ACM (2005)
7. Rusu, C., Ferreira, A., Scordino, C., Watson, A., Melhem, R., Mossé, D.: Energy-efficient real-time heterogeneous server clusters. In: Proceedings of RTAS, pp. 418–428 (2006)
8. Chen, G., He, W., Liu, J., Nath, S., Rigas, L., Xiao, L., Zhao, F.: Energy-aware server provisioning and load dispatching for connection-intensive internet services. In: Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation, pp. 337–350. USENIX Association (2008)
9. Gurumurthi, S., Sivasubramaniam, A., Kandemir, M., Franke, H.: DRPM: dynamic speed control for power management in server class disks. In: Proceedings of the 30th Annual International Symposium on Computer Architecture, p. 181. ACM (2003)
10. Zhu, Q., David, F., Devaraj, C., Li, Z., Zhou, Y., Cao, P.: Reducing energy consumption of disk storage using power-aware cache management (2004)
11. Ganesh, L., Weatherspoon, H., Balakrishnan, M., Birman, K.: Optimizing Power Consumption in Large Scale Storage Systems. In: Proceedings of the 11th USENIX Workshop on Hot Topics in Operating Systems, HotOS 2007 (2007)

12. Li, X., Li, Z., Zhou, Y., Adve, S.: Performance directed energy management for main memory and disks. ACM Transactions on Storage (TOS) 1(3), 380 (2005)
13. Moore, J., Chase, J., Ranganathan, P., Sharma, R.: Making scheduling gcool h: Temperature-aware workload placement in data centers. In: Proceedings of the USENIX Annual Technical Conference, pp. 61–75 (2005)
14. Bash, C., Forman, G.: Cool job allocation: Measuring the power savings of placing jobs at cooling-efficient locations in the data center. In: 2007 USENIX Annual Technical Conference on Proceedings of the USENIX Annual Technical Conference, pp. 1–6. USENIX Association (2007)
15. Tang, Q., Gupta, S., Varsamopoulos, G.: Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: a cyber-physical approach. IEEE Transactions on Parallel and Distributed Systems, 1458–1472 (2008)
16. Ghemawat, S., Gobioff, H., Leung, S.: The Google file system. ACM SIGOPS Operating Systems Review 37(5), 43 (2003)
17. Chang, F., Dean, J., Ghemawat, S., Hsieh, W., Wallach, D., Burrows, M., Chandra, T., Fikes, A., Gruber, R.: Bigtable: A distributed storage system for structured data. In: Proceedings of the 7th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2006 (2006)
18. Dean, J., Ghemawat, S.: Map Reduce: Simplified data processing on large clusters. Communications of the ACM-Association for Computing Machinery-CACM 51(1), 107–114 (2008)
19. Data Center Knowledge, Google's Chiller-less Data Center (2009), http://www.datacenterknowledge.com/archives/2009/07/15/googles-chiller-less-data-center/
20. Information Week, Google In Oregon: Mother Nature Meets The Data Center (2007), http://www.informationweek.com/blog/main/archives/2007/08/google_in_orego.html
21. Hamilton, J.: Architecture for modular data centers, Arxiv preprint cs/0612110 (2006)
22. Le, K., Bianchini, R., Martonosi, M., Nguyen, T.: Cost-and Energy-Aware Load Distribution Across Data Centers. In: Workshop on Power Aware Computing and Systems, HotPower 2009 (2009)
23. Barroso, L., Hölzle, U.: The datacenter as a computer: An introduction to the design of warehouse-scale machines. Synthesis Lectures on Computer Architecture 4(1), 1–108 (2009)
24. Mahadevan, P., Sharma, P., Banerjee, S., Ranganathan, P.: A Power Benchmarking Framework for Network Devices. In: Fratta, L., Schulzrinne, H., Takahashi, Y., Spaniol, O. (eds.) NETWORKING 2009. LNCS, vol. 5550, pp. 795–808. Springer, Heidelberg (2009)
25. Standard Performance Evaluation Corporation, SPECpower_ssj (2008), http://www.spec.org/power_ssj2008/
26. Fei, A., Pei, G., Liu, R., Zhang, L.: Measurements on delay and hop-count of the internet. In: IEEE GLOBECOM 1998-Internet Mini-Conference (1998)
27. Principled Technologies, WebBench performance on quad-core and dual-core dual-processor servers (2006), http://www.principledtechnologies.com/clients/reports/Intel/X5355WebBench1106.pdf
28. WebBench, http://cs.uccs.edu/~cs526/webbench/webbench.htm
29. Chabarek, J., Sommers, J., Barford, P., Estan, C., Tsiang, D., Wright, S.: Power awareness in network design and routing. In: IEEE INFOCOM (2008)