J Ambient Intell Human Comput (2011) 2:263–270
DOI 10.1007/s12652-010-0044-9

ORIGINAL RESEARCH

An extended power consumption-based algorithm for communication-based applications

Tomoya Enokido · Ailixier Aikebaier · Makoto Takizawa

Received: 25 August 2010 / Accepted: 14 December 2010 / Published online: 9 January 2011

Abstract In information systems, it is critical to reduce the total electrical power consumption of computers and networks in order to realize the eco society. In our previous studies, a pair of PCB (power consumption-based) and TRB (transmission rate-based) algorithms are proposed to select one of servers so that the total power consumption of the servers is reduced for communication-based applications. In addition, the PCB algorithm is shown to be more useful for reducing the total power consumption than the TRB one. However, we consider only how much amount of electric power a server is expected to consume to transmit a file to a new requesting client in the PCB algorithm. Here, the server might be transmitting files to other clients. Therefore, we have to estimate the power consumption to transmit a file to a new client by considering how much power consumption the server would spend to transmit files to other clients. In this paper, we newly propose an extended power consumption-based (EPCB) algorithm by improving the PCB algorithm by taking into consideration how much power a server consumes to transmit files to every requesting client. In the evaluation, we show the total power consumption of servers can be more reduced by the EPCB algorithm than the PCB and TRB algorithms.

Keywords Power consumption model · An extended power consumption-based (EPCB) algorithm · Power consumption-based algorithm · Transmission rate-based algorithm · Communication-based applications

1 Introduction

In the green IT technologies [Green IT, http://www.greenit.net (2010)], the total electric power consumption of computers and networks are required to be reduced. There are various kinds of applications on distributed systems. We classify applications into two types; transaction-based and communication-based network applications. In the papers (Aikebaier et al. 2009; Enokido et al. 2010a, c; Yang et al. 2009), it is discussed how to reduce the power consumption of servers in transaction-based applications like Web applications

In another type of application like the file transfer protocol (FTP), a large volume of data is transmitted by a server to a client. In the papers (Enokido et al. 2009, 2010b), we discussed a power consumption model for transmitting files from servers to clients where the electric power consumption of the server to transmit a file to a client depends on the transmission rate of the server on the experimental results (Enokido et al. 2009). Here, a pair of PCB (*power consumption-based*) and TRB (*transmission rate-based*) algorithms are proposed to select a server in a set of servers so that the total power consumption can be reduced. In the evaluation (Enokido et al. 2010b), the total power consumption and the total transmission time are shown to be reduced in the PCB and TRB algorithms

T. Enokido (✉)
Faculty of Bussiness Administration, Rissho University,
4-2-16 Osaki, Shinagawa-ku, Tokyo 141-8602, Japan
e-mail: eno@ris.ac.jp

A. Aikebaier · M. Takizawa
Department of Computers and Information Science,
Faculty of Science and Technology, Seikei University,
3-3-1 Kichijoji-kitamachi, Musashino-shi,
Tokyo 180-8633, Japan
e-mail: alisher.akber@computer.org

M. Takizawa
e-mail: makoto.takizawa@computer.org

compared with the traditional round-robin (RR) algorithm [Weighted Least Connection (WLC), http://www.linuxvirtualserver.org/docs/scheduling.html (1998); Weighted Round Robin (WRR), http://www.linuxvirtualserver.org/docs/scheduling.html (1998)]. The PCB algorithm is more useful than TRB algorithm since it is difficult to estimate the transmission rates to each client in the TRB algorithm.

Suppose a server s_t transmits a file f to a requesting client c_s at time τ. In the PCB algorithm, it is discussed only how much power the server s_t consumes to transmit a file f to the client c_s. However, there might be some clients receiving files from the server s_t. Therefore, we have to estimate how much amount of power the server s_t consumes to transmit files not only to the new requesting client c_s but also current clients which are receiving files from the server s_t. In this paper, we newly propose an *extended power consumption-based* (EPCB) algorithm by improving the PCB algorithm by taking into consideration how much power a server consumes to transmit a file to every requesting client. We evaluate the EPCB algorithm in terms of the total power consumption and total transmission time of the servers compared with the PCB and TRB algorithms. We show the total power consumption can be more reduced than the PCB and TRB algorithms.

In Sect. 2, we show the system model. In Sect. 3, we show the experimental results of the total power consumption in file transfer applications and the power consumption model. In Sect. 4, we discuss how to select a server for downloading a file to reduce the total power consumption. In Sect. 5, we evaluate the EPCB algorithm compared with the TRB and PCB algorithms.

2 System model

There are a set S of multiple servers $s_1, \ldots, s_n$ ($n \geq 1$), each of which holds a full replica of a file f. There are a set C of clients $c_1, \ldots, c_m$ ($m \geq 1$) which issue file transmission request to servers in S. A client c_s sends a transfer request of the file f to a load balancer K. Then, the load balancer K selects one server s_t in the server set S. The server s_t transmits the file f to the requesting client c_s as shown in Fig. 1.

We assume that a server s_t sends at most one file to each client at same time. We discuss how to select a server in the server set S for a client c_s so that the following constraints are satisfied:

1. The deadline constraint, i.e. the file has to be delivered to the requesting client c_s in some time units.
2. The power consumption of a selected server s_t to transfer the file f has to be minimized.

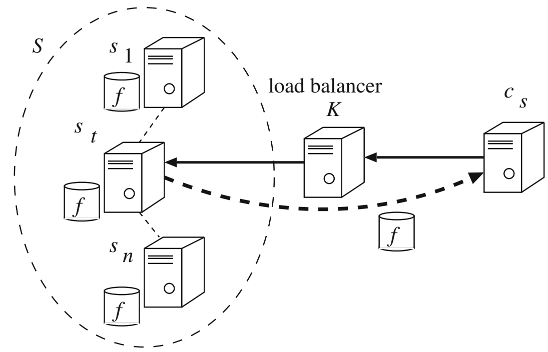

Fig. 1 FTP model

Suppose a server s_t concurrently transmits files $f_1, \ldots, f_m$ to a set C_t of clients $c_1, \ldots, c_m$ at rates $\text{tr}_{t1}(\tau), \ldots, \text{tr}_{tm}(\tau) (m \geq 1)$, respectively, at time τ. Let b_{ts} show the maximum network bandwidth [bps] between a server s_t and a client c_s. Let Maxtr_t be the maximum transmission rate [bps] of the server s_t ($\leq b_{ts}$) which is smaller than the network bandwidth b_{ts}. Here, the total transmission rate $\text{tr}_t(\tau)$ of the server s_t at time τ is given as $\text{tr}_t(\tau) = \text{tr}_{t1}(\tau) + \cdots + \text{tr}_{tm}(\tau)$. Here, $0 \leq \text{tr}_t(\tau) \leq \text{Maxtr}_t$.

Each client c_s receives messages of a file f_s at receipt rate $\text{rr}_s(\tau)$ at time τ. Let Maxrr_s indicate the maximum receipt rate of the client c_s. Here, $\text{rr}_s(\tau) \leq \text{Maxrr}_s$. We assume each client c_s receives a file from at most one server at rate $\text{Maxrr}_s (= \text{rr}_s(\tau))$. The server s_t allocates a client c_s with transmission rate $\text{tr}_{ts}(\tau)$ so that $\text{tr}_{ts}(\tau) \leq \text{Maxrr}_s$ at time τ.

Let T_{ts} be the total transmission time of a file f_s from a server s_t to a client $c_s (s = 1, \ldots, m)$. If the server s_t sends files to other clients concurrently with the client c_s, the transmission time T_{ts} is increased. Let $\text{min}T_{ts}$ show the minimum transmission time $|f_s|/\min(\text{Maxrr}_s, \text{Maxtr}_t)$ [s] of a file f_s from a server s_t to a client c_s where $|f_s|$ indicates the size [bit] of the file f_s. $T_{ts} \geq \text{min}T_{ts}$.

Let $\text{tr}_{ts}(\tau)$ be the transmission rate of a file f_s from the server s_t to the client c_s at time τ. Suppose the server s_t starts and ends transmitting a file f_s to the client c_s at time st and et, respectively. Here, $\int_{st}^{et} \text{tr}_{ts}(\tau) d\tau = |f_s|$ and the transmission time T_{ts} is $(et - st)$. If the server s_t sends only the file f_s to the client c_s at time $\tau, \text{tr}_{ts}(\tau) = \min(\text{Maxtr}_t, \text{Maxrr}_s)$ [bps].

The laxity $l_{ts}(\tau)$ of transmission time is $|f_s| - \int_{\tau}^{et} \text{tr}_{ts}(x) dx$ [bit] at time τ, i.e. how many bits of the file f_s the server s_t still has to transmit to the client c_s at time τ.

There are types of computers with respect to the transmission rate. Let $C_t(\tau)$ be a set of clients $c_1, \ldots, c_m$ to which the server s_t transmits files $f_1, \ldots, f_m$, respectively, at time τ. First, we consider a model where a server s_t satisfies the following properties:

[Server-bound model] If $\text{Maxrr}_1 + \cdots + \text{Maxrr}_m \geq \text{Maxtr}_t$, $\sum_{c_{ts} \in C_t(\tau)} \text{tr}_{ts}(\tau) = d(\tau) \cdot \text{Maxtr}_t$ for every time τ.

Here, $d(\tau)$ (≤ 1) is the degradation factor. In this paper, we assume $d(\tau) = \gamma^{(1-|C_t(\tau)|)} (0 < \gamma \leq 1)$ at time τ. The more number of clients a server s_t transmits files, the longer it takes. Here, the *effective transmission rate* of the server s_t is $d(\tau) \cdot \text{Maxtr}_t$. The more number of clients a server concurrently sends files, the smaller effective transmission rate.

Let us consider three files f_1, f_2, and f_3 which a server s_t sends to clients c_1, c_2, and c_3, respectively, as an example. First, suppose that the server s_t serially sends the files f_1, f_2, and f_3 to the clients c_1, c_2, and c_3, i.e. $et_1 = st_2$ and $et_2 = st_3$ as shown in Fig. 2. Here, the transmission time T_t is $et_3 - st_1 = \min T_{t1} + \min T_{t2} + \min T_{t3}$. Next, suppose the server s_t starts transmitting three files f_1, f_2, and f_3 at time st and terminates at time et as shown in Fig. 2(2). Here, since three files are concurrently transmitted, $C_t(t) = 3$ and $\gamma^{-2} \cdot T_t = \min T_{t1} + \min T_{t2} + \min T_{t3}$. For $\gamma = 0.98$, it takes about 1.4% longer time than the serial transmission.

On the other hand, we consider another environment where a client c_s cannot receive a file from a server s_t at the maximum transmission rate Maxtr_t, i.e. $\text{Maxrr}_s < \text{Maxtr}_t$. Hence, the transmission rate tr_{ts} of the server s_t to a client c_s is the maximum receipt rate Maxrr_s of the client c_s.

[Client-bound model] If $\text{Maxrr}_1 + \cdots + \text{Maxrr}_m \leq \text{Maxtr}_t$, $\sum_{c_{ts} \in C_t(\tau)} \text{tr}_{ts}(\tau) = \text{Maxtr}_t \cdot (\text{Maxrr}_1 + \cdots + \text{Maxrr}_m) / \text{Maxtr}_t$ for every time τ.

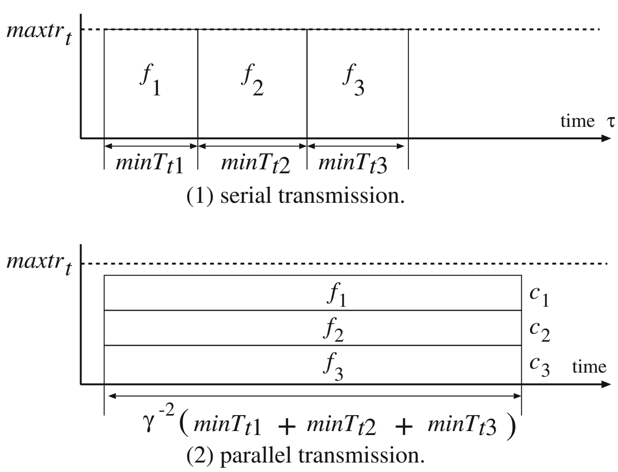

Fig. 2 Transmission time

Even if every client c_s receives a file at the maximum rate Maxrr_s, the effective transmission rate is not degraded, i.e. $d_t(\tau) = 1$.

3 Power consumption model

The amount of electric power a server consumes to transfer files to client is measured by using the power meter Watts up?.Net [Watts up? Net, http://www.wattsupmeters.com/secure/products.php?pn=0 (2009)] in the paper (Enokido et al. 2010b). Here, the power consumption of each server can be measured every one second. Two servers s_1 and s_2 are measured as shown in Table 1. For each server s_t, two types of experimentation environments, one-client ($1C_t$) and two-client ($2C_t$) environments ($t = 1, 2$) are considered. In the $1C_t$ environment, one client, say c_1 downloads the file f from the server s_t. In the $2C_t$ environment, a pair of the clients c_1 and c_2 concurrently download the file f from the server s_t. A server s_t consumes the electric power to transmit files to clients while clients consume less amount of electric power. The power consumption rate shows the electric power consumption for a second (W/s). In the $1C_1$ environment, the server s_1 transmits a file f to one client, say c_1 at rate tr_{11}. Here, the server s_1 is composed of one one-core CPU. The maximum transmission rate Maxtr_1 is 160 Mbps in the network of bandwidth $b_{11} = 1G$ [bps]. In the $2C_1$ environment, the server s_1 concurrently transmits the file f to a couple of clients c_1 and c_2. Here, $\text{tr}_1 = \text{tr}_{11} + \text{tr}_{12}$. At the higher rate tr_1 the server s_1 transmits the file f, the larger amount of power consumption the server s_1 consumes. We obtain the approximated formula $PC_1(tr)$ to show the power consumption rate of a server s_1 for total transmission rate tr [Mbps] by using the least-squares method to the experimental results:

$$PC_t(tr) = \beta_t(m) \cdot \alpha_t \cdot tr + \min E_t. \quad (1)$$

Here, α_t is the power consumption to transmit one Mbits [W/Mb] for the $1C_t$ environment. α_t depends on a server type s_t. In Table 2, parameters obtained by the experiments are shown. m shows the number of clients in a server. $\beta_t(m)$ shows how much power consumption is increased for the number m of clients, $\beta_t(m) \geq 1$ and $\beta_t(m) \geq \beta_t(m-1)$. There is a fixed point $\max m_t$ such that $\beta_t(\max m_t - 1)$

Table 1 Servers

Server	s_1	s_2
Number of CPUs	1	2
Number of cores/CPU	1	2
CPU	AMD Athlon 1648B (2.7GHz)	AMD Opteron 270 (2GHz)
Memory	4,096MB	4096MB
DISK	150GB 7200rpm	74GB 10,000 rpm x 2 RAID1
NIC	Broadcom Gbit Ether (1Gbps)	Nvidia Ether Controler (1Gbps)

Table 2 Parameters

t	1		2	
m	1	2	1	2
α_t	0.11	0.12	0.02	0.03
$\min E_t$	4.15	4.43	3.02	3.34
$\beta_t(m)$	1	1.09	1	1.5

$\leq \beta_t(\max m_t) = \beta_t(\max m_t + h)$ for $h > 0$. $\min E_t$ gives the minimum power consumption rate of the server s_t where no file is transmitted (Fig. 3). $\beta_t(\max m_t) \cdot \alpha_t \cdot \mathrm{Maxtr}_t + \min E_t$ gives the maximum power consumption rate $maxE_t$ of the server s_t.

Let $E_t(\tau)$ show the electric power consumption rate of a server s_t at time τ (W/s) ($t = 1, \ldots, n$). $maxE_t$ and $\min E_t$ indicate the maximum and minimum electric power consumption rates of a server s_t, respectively. Here, $\min E_t$ shows the power consumption rate of a server s_t which is in idle state. That is, $\min E_t \leq E_t(\tau) \leq \max E_t$ for every time τ. Let $\max E$ and $\min E$ show $\max(\max E_1, \ldots, \max E_n)$ and $\min(\min E_1, \ldots, \min E_n)$, respectively. In this paper, we assume that only file transfer applications are performed on each server and each client issues at most one file transfer request at a time. The electric power consumption rate $E_t(\tau)$ of a server s_t at time τ is given as follows:

$$E_t(\tau) = PC_t(\mathrm{tr}_t(\tau)) = \beta_t(|C_t(\tau)|) \cdot \alpha_t \cdot \mathrm{tr}_t(\tau) + \min E_t. \tag{2}$$

Here, $C_t(\tau)$ indicates a set of clients to which a server s_t sends files at time τ.

The power consumption $\mathrm{TPC}_t(\tau_1, \tau_2)$ [W] of a server s_t from time τ_1 to time τ_2 is given as follows:

$$\mathrm{TPC}_t(\tau_1, \tau_2) = \int_{\tau_1}^{\tau_2} E_t(\tau) d\tau. \tag{3}$$

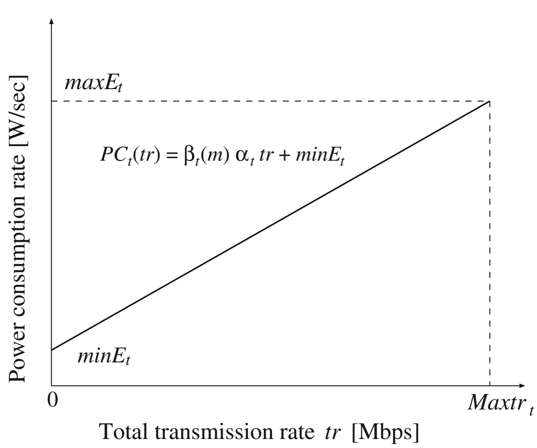

Fig. 3 Power consumption rate of server s_t (W/s)

4 Selection algorithms of servers

4.1 Transmission rates

At time τ, the maximum transmission rate $\mathrm{maxtr}_t(\tau)$ of a server s_t depends on the degradation factor $d_t(\tau)$ of the server s_t, i.e. the number of requesting clients to which the server s_t concurrently transmits files at time τ. Each time a new request is issued by a client c_s and a current request for a client c_s is terminated at time τ, $C_t(\tau) = C_t(\tau) \cup \{c_s\}$ and $C_t(\tau) = C_t(\tau) - \{c_s\}$, respectively. Here, the maximum transmission rate $\mathrm{maxtr}_t(\tau)$ of a server s_t at time τ is calculated as $\gamma^{1-|C_t(\tau)|} \cdot \mathrm{Maxtr}_t$. Here, $0 < \gamma \leq 1$. In the fair allocation algorithm, the transmission rate $\mathrm{tr}_{ts}(\tau)$ for each client c_s in $C_t(\tau)$ is the same, i.e. $\mathrm{tr}_{ts}(\tau) = \mathrm{maxtr}_t(\tau)/|C_t(\tau)|$. The server s_t can transmit messages to a client c_s at rate $\mathrm{maxtr}_t(\tau)/|C_t(\tau)|$. However, the maximum receipt rate Maxrr_s of the client c_s might be smaller than $\mathrm{maxtr}_t(\tau)/|C_t(\tau)|$. Here, the rate $(\mathrm{maxtr}_t(\tau)/|C_t(\tau)| - \mathrm{Maxrr}_s)$ is not used. In order to efficiently use the total transmission rate $\mathrm{maxtr}_t(\tau)$, the higher receipt rate a client c_s has, the higher transmission rate a server s_t allocates to c_s.

In another proportional way, the transmission rate $\mathrm{tr}_{ts}(\tau)$ for each client c_s is proportional to the maximum receipt rate Maxrr_s. Here, $\mathrm{tr}_{ts}(\tau) = \mathrm{Maxtr}_t \cdot \mathrm{Maxrr}_s / (\sum_{c_s t(\tau)} \mathrm{Maxrr}_s)$. Here, a slower client has to receive messages at slower rate.

In this paper, we consider a novel algorithm where each client c_s is guaranteed to be allocated with $\mathrm{maxtr}_t(\tau)/|C_t(\tau)|$ if needed. However, a client c_s may not use the transmission rate $u_{ts}(\tau) = \mathrm{maxtr}_t(\tau)/(|C_t(\tau)| - \mathrm{Maxrr}_s)$ if the allocated transmission rate $\mathrm{maxtr}_t(\tau)/|C_t(\tau)|$ is larger than the maximum receipt rate Maxrr_s. The unused transmission rate $u_{ts}(\tau)$ is allocated to other clients which need the higher rate. The transmission rate $\mathrm{tr}_{ts}(\tau)$ is decided so that the following conditions hold:

$$\text{minimize}\left(\mathrm{Maxtr}_t - \sum\nolimits_{c_s \in C_t(\tau)} \mathrm{tr}_{ts}(\tau)\right).$$

$$\text{minimize} \sum_{c_s \in C_t(\tau)} (\mathrm{Maxrr}_s - \mathrm{tr}_{ts}(\tau)).$$

In the papers (Enokido et al. 2009, 2010b), the following algorithm to allocate each client c_s with the transmission rate $\mathrm{tr}_{ts}(\tau)$ is discussed for each time τ:

1. $V = 0$; $R = 0$; $TS = \mathrm{maxtr}_t/|C_t(\tau)|$;
2. For each client $c_s, \mathrm{tr}_t(\tau) = TS$ and $R = R + (TS - \mathrm{Maxrr}_s)$ if $\mathrm{Maxrr}_s \leq TS$. Otherwise, $\mathrm{tr}_t(\tau) = \mathrm{Maxrr}_s$ and $V = V + (\mathrm{Maxrr}_s - TS)$.
3. For each client $c_s, \mathrm{tr}_{ts}(\tau) = \mathrm{tr}_{ts}(\tau) + V \cdot (\mathrm{Maxrr}_s - \mathrm{tr}_{ts}(\tau))/R$ if $\mathrm{tr}_{ts}(\tau) < \mathrm{Maxrr}_s$.

Suppose a server s_t is selected by three clients c_1, c_2, c_3 ($C_t(\tau) = \{c_1, c_2, c_3\}$) and the maximum transmission rate

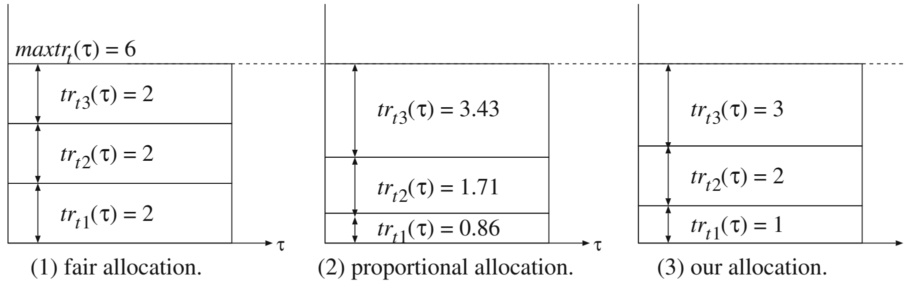

Fig. 4 Transmission rate allocation

$\text{maxtr}_t(\tau)$ of the server s_t is 6 Mbps at time τ as shown in Fig. 4. Suppose $\text{Maxrr}_1 = 1$ Mbps, $\text{Maxrr}_2 = 2$ Mbps, and $\text{Maxrr}_3 = 4$ Mbps. In the basic fair allocation algorithms, each client c_s is allocated with the same transmission rate $\text{tr}_{ts}(\tau) = \text{maxtr}_t(\tau)/|C_t(\tau)| = 6/3 = 2$ Mbps as shown in Fig. 4(1). Here, the transmission rate $2 - 1 = 1$ Mbps is not used for the client c_1. In addition, the client c_3 cannot use the maximum receipt rate Maxrr_3 (= 4 Mbps). In the proportional way, the client c_1, c_2, and c_3 are allocated with $6/7 = 0.86$, $12/7 = 1.71$, and $24/7 = 3.43$ Mbps, respectively. In the algorithm discussed here, the unused transmission rate of the client c_1 (= 1 Mbps) can be used for the client c_3. That is, the clients c_1, c_2, and c_3 receive files at 1, 2, and 3 Mbps, respectively.

4.2 TRB and PCB algorithms

Next, we discuss how a load balancer K selects a file server s_t for a client c_s in the server set S. In the paper (Enokido et al. 2009), two types of server selection algorithms, *transmission rate-based* (*TRB*) and *power consumption-based* (*PCB*) algorithms to select a server for a client are proposed. In addition, we evaluated the PCB and TRB algorithms (Enokido et al. 2010b) in terms of the total power consumption and the total transmission time compared with the basic round-robin (RR) algorithm [Weighted Least Connection (WLC), http://www.linuxvirtualserver.org/docs/scheduling.html (1998); Weighted Round Robin (WRR), http://www.linuxvirtualserver.org/docs/scheduling.html (1998)]. In the TRB algorithm, a server s_t is selected for a client c_s where the transmission rate $\text{tr}_{ts}(\tau)$ of the server s_t to transmit a file f to a client c_s is the largest. On the other hand, in the PCB algorithm, a server s_t is selected for the client c_s where the power consumption is the smallest. Here, $|f|/\text{tr}_{ts}(\tau)$ is an estimated transmission time at time τ when a server s_t starts transmitting a file f to a client c_s with a transmission rate $\text{tr}_{ts}(\tau)$. The power consumption rate $E_{ts}(\tau)$ of each server s_t at time τ is $\beta_t(|C_t(\tau)|) \cdot \alpha_t \cdot \text{tr}_{ts}(\tau)$ as discussed in the preceding section. It is not easy to estimate how much electric power the server s_t consumes to transmit a file f to the client c_s since there might be other clients which receive files. Here, the estimated change of power consumption $EE_{ts}(\tau)$ [W] of a server s_t for transmitting a file f to a client c_s at time τ when the server s_t starts transmitting the file f is $(|f|/\text{tr}_{ts}(\tau)) \cdot \beta_t(|C_t(\tau)|) \cdot \alpha_t \cdot \text{tr}_{ts}(\tau) = |f| \cdot \beta_t(|C_t(\tau)|) \cdot \alpha_t$. Here, a server s_t whose $EE_{ts}(\tau)$ is minimum is selected for a client c_s.

According to the evaluation results (Enokido et al. 2010b), the total power consumption and the total transmission time can be reduced in the PCB and TRB algorithms compared with the basic RR algorithm. In the PCB algorithm, the total power consumption can be more reduced than the TRB algorithm and the difference of the total transmission time between PCB and TRB is almost neglectable. In reality, the transmission rate between a server s_t and a client c_s is dynamically changed in the network since the transmission rate of a server s_t is dynamically changed based on the number of clients. It is not easy to estimate the transmission rate of the server s_t to a client c_s from the practical point of view. In addition, a server s_t for a client c_s can be selected without considering the transmission rate between the server s_t and the client c_s in the PCB algorithm. Therefore, the PCB algorithm is simpler and more useful than the TRB and RR algorithms.

4.3 Extended power consumption-based (EPCB) algorithm

In this paper, we newly propose an *extended power consumption-based* (EPCB) algorithm by improving the PCB algorithm. Suppose a server s_t is selected for a client c_s and starts transmitting a file to the client c_s at time τ. In the PCB algorithm, the estimated change of power consumption $EE_{ts}(\tau) = |f| \cdot \beta_t(|C_t(\tau)|) \cdot \alpha_t$ is considered to select a server s_t. This means, only the power consumption for a server s_t to transmit a file f_s to a client c_s at time τ is considered. The server s_t might be transmitting the file f to other clients at time τ when starting transmitting the file f to the client c_s. The power consumption to be consumed by a server s_t to transmit files to other clients has to be considered for obtaining the more correct estimation of the power consumption. Here, $l_{ts}(\tau)$ shows laxity of a file f_s where a server s_t has to transmit the file f to a client c_s from time τ as presented in the preceding section. Here, the estimated change $EE_{ts}(\tau)$ [W] of power consumption of a server s_t for transmitting a file f_s to a client c_s at time τ

when the server s_t starts transmitting the file f is defined as follows:

$$EE_{ts}(\tau) = \Sigma_{c_s \in C_t(\tau)} l_{ts}(\tau) \cdot \beta_t(|C_t(\tau)|) \cdot \alpha_t. \quad (4)$$

Here, it is noted $l_{ts}(\tau)$ is $|f_s|$ at time τ when the server s_t starts transmitting a file f_s to a client c_s. A server s_t is selected for a client c_s in the EPCB algorithm by using $EE_{ts}(\tau)$ at time τ as follows:

```
EPCB(c_s, τ) {
  server = φ;  EPC = 0;
  for each s_t in S, EPC_ts(τ) = Σ_{c_s∈C_t(τ)} |l_ts(τ)| · β_t(|C_t(τ)|) · α_t;
  server = s_t in S such that EPC_ts(τ) is minimum;
  return(server);
}
```

Let us consider an example of a pair of servers s_1 and s_2. The power consumption coefficients α_1 and α_2 to transmit 1 Mbit for one client of servers s_1 and s_2 are 0.09 and 0.07, respectively. The server s_1 is selected by a client c_1 ($C_1(\tau) = \{c_1\}$) and the server s_2 is selected by a client $c_2(C_2(\tau) = \{c_2\})$ at time τ, respectively. Suppose a client c_3 issues a new request to transmit a file f whose size is one Gbytes to a load balancer K at time τ. Here, a pair of laxity $l_{11}(\tau)$ and $l_{21}(\tau)$ are 0.1 and 0.9 GByte, respectively, at time τ. In the PCB algorithm, a server s_t which has the minimum value of the formula $|f| \cdot \beta_t(|C_t(\tau)|) \cdot \alpha_t$ is selected in the PCB algorithm. Here, sets $C_1(\tau)$ and $C_2(\tau)$ of current clients of servers s_1 and s_2 include two clients, $\{c_1, c_3\}$ and $\{c_2, c_3\}$, respectively. Suppose the increasing rates $\beta_1(2)$ and $\beta_2(2)$ of the power consumption of the servers s_1 and s_2 are 1.2 and 1.1, respectively. Here, $|f| \cdot \beta_1(2) \cdot \alpha_1 = 1 \cdot 1.2 \cdot 0.09 = 0.108.|f| \cdot \beta_2(2) \cdot \alpha_2 = 1 \cdot 1.1 \cdot 0.07 = 0.077$. Therefore, the server s_2 is selected for a client c_3 in the PCB algorithm. On the other hand, a server s_t which has the minimum value of the formula $\Sigma_{c_s \in C_t(\tau)} l_{st}(\tau) \cdot \beta_t(|C_t(\tau)|) \cdot \alpha_t$ is selected in the EPCB algorithm. Here, $\Sigma_{c_s \in C_1(\tau)} l_{1s}(\tau) \cdot \beta_1(2) \cdot \alpha_1 = (1 + 0.1) \cdot 1.2 \cdot 0.09 = 0.119$. $\Sigma_{c_s \in C_2(\tau)} l_{2s}(\tau) \cdot \beta_2(2) \cdot \alpha_2 = (1 + 0.9) \cdot 1.1 \cdot 0.07 = 0.146$. Therefore, the server s_1 is selected for a client c_3 in the EPCB algorithm.

5 Evaluation

5.1 Evaluation environment

We evaluate the EPCB algorithm in terms of the total amount of power consumption and total transmission time of files compared with the TRB and PCB algorithms through the simulation. In the evaluation, there are five servers s_1, s_2, s_3, s_4, and s_5 as shown in Table 3, $S = \{s_1, s_2, s_3, s_4, s_5\}$. The power consumption coefficient α_t to transmit one Mbits for one client of each server s_t is randomly selected between 0.02 and 0.11 W/Mb based on

Table 3 Types of servers

Servers	α	$\beta(m)$	min$E$ (W)	Maxtr (Mbps)
s_1	0.03	1.259	3.39	406
s_2	0.05	1.195	3.17	401
s_3	0.03	1.285	3.12	249
s_4	0.09	1.117	3.90	231
s_5	0.02	1.162	3.02	171

the experimental results. The increasing rate of the power consumption $\beta_t(m)$ for the number m of clients is randomly selected between 1.09 and 1.5. The minimum power consumption rate minE_t of each server s_t is randomly selected between 3 and 4 W. The maximum transmission rate Maxtr$_t$ of each server s_t is randomly selected between 150 and 450 Mbps. Each server s_t has a full replica of a file f. The size $|f|$ of the file f is one giga-byte.

Totally 100 clients download the file f from one server s_t in the server set S. The maximum receipt rate Maxrr$_s$ of each client c_s is randomly selected between 1 and 100 Mbps. Each client c_s issues a transfer request of the file f to a load balancer K at time st_s. Here, the starting time st_s of each client c_s is randomly selected between 1 and 3,600 s at the simulation time. Each client c_s issues one file transfer request at time st_s in the simulation. In the simulations of the EPCB, TRB, and PCB algorithms, the starting time st_s of the file transmission to each client c_s is the same.

5.2 Total power consumption

Figure 5 shows the total power consumption rates (W/s) of the servers $s_1, \ldots, s_5$ at each time. Table 4 shows the total power consumptions of the servers in the EPCB, PCB, and TRB algorithms. The total power consumptions of the EPCB, PCB, and TRB algorithms are 204,973, 209,831, and 250,176 W, respectively. In the TBR algorithm, a

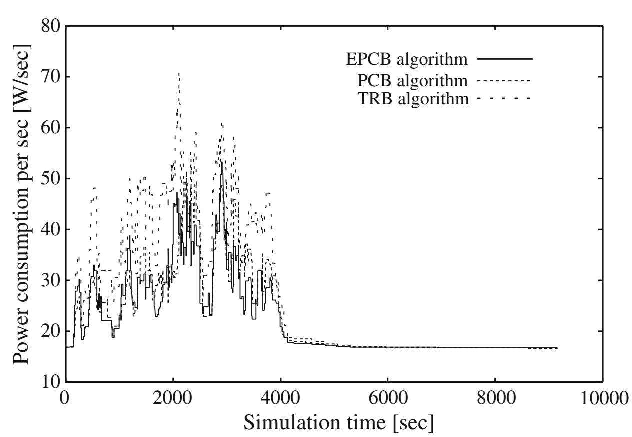

Fig. 5 Total power consumption rate

Table 4 Total amount of power consumption (W)

EPCB	PCB	TRB
204,973	209,831	250,176

Table 5 Total transmission time (s) of the files

PCB	TRB	RR
9,134	9,160	9,087

server s_t is selected for a client c_s, whose transmission rate for the client c_s is the largest. Then, the total amount of power consumption is larger than the EPCB and PCB algorithms. In the EPCB and PCB algorithms, a server s_t is selected for a client c_s, whose power consumption is the smallest to transmit a file *f* to the client c_s. The total amount of power consumption of the EPCB algorithm is smaller than the PCB algorithm. The total power consumption in the EPCB algorithm is reduced about 2% than the PCB algorithm.

5.3 Total transmission time

Table 5 shows the total transmission time of the files to the 100 clients in the PCB, TRB, and RR algorithms. The total transmission time are 28,614, 28,594, and 43,744 s in the PCB, TRB, and RR algorithms, respectively. The total transmission time of the TRB algorithm is smaller than the PCB and RR algorithms. However, the difference of the total transmission time between TRB and PCB is neglectable. In the TRB algorithm, a server s_t is selected, which can supply the maximum transmission rate. Therefore, the difference of the transmission time between PCB and TRB is so small as to be neglected in this simulation.

In the EPCB and PCB algorithms, a server s_t is selected for a client c_s without considering the transmission rate between the server s_t and the client c_s. On the other hand, a server s_t is selected for a client c_s based on the estimated transmission rate in the TRB algorithm. From the evaluation results, we consider the total power consumption can be more reduced in the EPCB algorithm than the PCB and TRB algorithms and the difference of the total transmission time between the EPCB, PCB and TRB algorithms is neglectable. In reality, the transmission rate between a server s_t and a client c_s is dynamically changed in the network since the transmission rate of a server s_t is dynamically changed based on the number of clients. It is not easy to estimate the transmission rate of the server s_t to a client c_s from the practical point of view. A server s_t for a client c_s can be selected without considering the transmission rate between the server s_t and the client c_s in the EPCB and PCB algorithms. In addition, The power consumption can be more reduced in the EPCB algorithm than the PCB algorithm. Therefore, the EPCB algorithm is simpler and more useful than the TRB and PCB algorithms.

6 Concluding remarks

In this paper, we proposed the EPCB algorithm to select one file transfer server for a requesting client by improving the PCB algorithm. The PCB algorithm is more useful than the TRB and traditional RR algorithms since the transmission rate is not considered. However, only the power consumption of a server s_t to transmit a file *f* to a new requesting client c_s is considered to estimate the total power consumption of the server s_t for transmitting the file *f*. There might be some clients receiving files from the server s_t on receipt of a file transfer request from the client c_s. Therefore, the power consumption of the server to transmit files to other clients has to be considered for estimating the total power consumption of the server s_t. In this paper, we improved the estimation of the power consumption of a server for transmitting a file. We evaluated the EPCB algorithm in terms of the total power consumption and the total transmission time compared with the TRB and PCB algorithms through simulation. According to the evaluation results, the total power consumption can be reduced in the EPCB algorithm compared with the TRB and PCB algorithms. The difference of the total transmission time among the EPCB, PCB and TRB algorithms is almost neglectable. Therefore, the EPCB algorithm is more useful for reducing the total power consumption in the communication-based applications.

Acknowledgments This research was partially supported by the strategy research project of Seikei University and MEXT, Grant in Aid for Building Strategy Research Information.

References

Aikebaier A, Enokido T, Takizawa M (2009) Energy-efficient computation models for distributed systems. In: Proceedings of the 12th international conference on network-based information systems (NBiS2009), pp 29–36

Akyildiz IF, Kasimoglu IH (2004) Wireless sensor and actor networks: research challenges. Ad Hoc Netw J 2:351–367

Enokido T, Suzuki K, Aikebaier A, Takizawa M (2009) Algorithms for reducing the total power consumption in data communication-based applications. In: Proceedings of 24th IEEE international conference on advanced information networking and applications (AINA-2010), pp 142–149

Enokido T, Aikebaier A, Takizawa M (2010a) A model for reducing power consumption in peer-to-peer systems. IEEE Syst J 4(2): 221–229

Enokido T, Aikebaier A, Takizawa M (2010b) Power consumption-based server selection algorithms for communication-based systems. In: Proceedings of the 13th international conference on network-based information systems (NBiS2010) (2010, accepted)
Enokido T, Suzuki K, Aikebaier A, Takizawa M (2010c) Laxity based algorithm for reducing power consumption in distributed systems. In: Proceedings of the 4th international conference on complex, intelligent and software intensive systems (CISIS2010), pp 321–328
Yang Y, Xiong N, Aikebaier A, Enokido T, Takizawa M (2009) Minimizing power consumption with performance efficiency constraint in web server clusters. In: Proceedings of the 12th international conference on network-based information systems (NBiS2009), pp 45–51