POWER CONSUMPTION-BASED AND TRANSMISSION RATE-BASED ALGORITHMS IN COMMUNICATION-BASED NETWORK APPLICATIONS

Tomoya Enokido
Faculty of Bussiness Administration, Rissho University, Tokyo, Japan

Ailixier Aikebaier, and Makoto Takizawa
Department of Computers and Information Science, Seikei University, Musashino-shi, Tokyo, Japan

Keywords: Green IT, Power consumption, Communication-based systems, Power Consumption-Based (PCB) algorithm, Transmission Rate-Based (TRB) algorithm.

Abstract: In order to realize eco societies, we have to reduce the total electrical power consumption in information systems. We classify network applications into transaction and communication based applications. CPU resources of servers are mainly consumed in the transaction based ones. In this paper, we consider communication based applications where a server transmits a large volume of data to a client like file transfer protocol (FTP). We discuss a power consumption model for communication-based applications. In the model, the total power consumption of a server depends on the total transmission rate and number of clients where the server concurrently transmits files. A client has to select a server in a set of possible servers, each of which holds a file, so that the power consumption of the server is reduced. We newly discuss a pair of PCB (power consumption-based) and TRB (transmission rate-based) algorithms to select a server. In the evaluation, we show the total power consumption can be reduced by the PCB and TRB algorithms compared with the traditional round-robin (RR) algorithm and PCB is more practical than TRB.

1 INTRODUCTION

In the green IT technologies (Green IT, 2010), the total electric power consumption of computers and networks has to be reduced. Various types of hardware technologies like low-power consumption CPUs and storages are now being developed. A cloud computing system (Grossman, 2009; Zhang and Zhou, 2009) is composed of a huge number of server computers like Google file systems (Ghemawat et al., 2003). Biancini *et al*. (Bianchini and Rajamony, 2004) discuss how to reduce the power consumption of a cluster of homogeneous servers by turning off servers which are not required for executing a collection of web requests. Various types of algorithms to find required number of servers in homogeneous and heterogeneous servers are discussed (Heath et al., 2005; Rajamani and Lefurgy, 2003; Aikebaier et al., 2009; Yang et al., 2009b). In wireless sensor networks (Akyildiz and Kasimoglu, 2004; Yang et al., 2009a), routing algorithms (Zhao et al., 2010) to reduce the power consumption of the battery in a sensor node are discussed.

There are transaction-based and communication-based network applications. We discussed how to reduce the power consumption in transaction-based applications like Web applications (Aikebaier et al., 2009; Enokido et al., 2010b; Enokido et al., 2010a; Yang et al., 2009b). Clients issue Web requests to servers. Then the servers encode multimedia contents and send replies with the encoded contents to the clients. We assume the communication bandwidth is infinite, i.e. the communication overhead is so small as to be neglected compared with the processing overhead of servers, mainly for encoding multimedia objects. In another type of application like the file transfer protocol (FTP), a large volume of data is transmitted by a server to a client. According to our experiments, the power consumption of the server to transmit a file to a client depends on the transmission rate of the server. First, a client finds a server which holds a file so that not only the time constraints

Enokido T., Aikebaier A. and Takizawa M. (2010).
POWER CONSUMPTION-BASED AND TRANSMISSION RATE-BASED ALGORITHMS IN COMMUNICATION-BASED NETWORK APPLICATIONS.
In *Proceedings of the Multi-Conference on Innovative Developments in ICT*, pages 13-22
DOI: 10.5220/0003034800130022

are satisfied but also the power consumption of the server is reduced. In this paper, we discuss a power consumption model for transmitting files based on the experimental results. We newly discuss a pair of PCB (*power consumption-based*) and TRB (*transmission rate-based*) algorithms to select a server in a set of servers so that the total power consumption can be reduced. We evaluate the PCB and TRB algorithms in terms of the total power consumption and the total transmission time compared with the traditional round-robin (RR) algorithm (Weighted Least Connection (WLC), 1998; Weighted Round Robin (WRR), 1998). We show the total power consumption and the total transmission time can be reduced in the PCB and TRB algorithms. The TRB algorithm is based on the transmission rate but it is difficult to estimate the bandwidth since the transmission rate is in reality changed in the networks. Hence, the PCB algorithm is more useful than the others since the transmission rate is not considered.

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

2 FILE TRANSFER MODEL

Suppose there are a collection $S = \{s_1, ..., s_n\}$ of servers, where each server s_t holds a full replica of a file f. A client c_s selects one server s_t in the server set S and issues a transmission request to the server s_t. Then, the server s_t transmits the file f to the client c_s as shown in Figure 1.

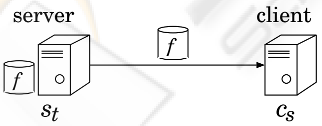

Figure 1: File transfer model.

Suppose a server s_t concurrently sends files $f_1, ..., f_m$ to a set C_t of clients $c_1, ..., c_m$ at rates $tr_{t1}(\tau), ..., tr_{tm}(\tau)$ ($m \geq 1$), respectively, at time τ. b_{ts} shows the maximum 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 bandwidth b_{ts} of the network. Here, the total transmission rate $tr_t(\tau)$ of the server s_t at time τ is given as $tr_t(\tau) = tr_{t1}(\tau) + \cdots + tr_{tm}(\tau)$. Here, $0 \leq tr_t(\tau) \leq Maxtr_t$.

Each client c_s receives messages at receipt rate $rr_s(\tau)$ at time τ. Let $Maxrr_s$ indicate the maximum receipt rate of the client c_s. Here, $tr_{ts}(\tau) \leq Maxrr_s$. We assume each client c_s receives a file from at most one server at rate $Maxrr_s$ (= $rr_s(\tau)$). The server s_t allocates each client c_s with transmission rate $tr_{ts}(\tau)$ so that $tr_{ts}(\tau) \leq Maxrr_s$ 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. If the server s_t sends files to other clients concurrently with the client c_s, the transmission time T_{ts} is increased. Let $minT_{ts}$ show the minimum transmission time $|f_s|$ / $min(Maxrr_s, Maxtr_t)$ [sec] 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. $T_{ts} \geq minT_{ts}$.

The *average transmission rate* (*ATR*) A_{ts} of the server s_t to the client c_s is defined as 1 / T_{ts} [1/sec]. Let $maxA_{ts}$ be 1 / $minT_{ts}$. $maxA_s = max(maxA_{1s}, ..., maxA_{ns})$ and $minA_s = min(maxA_{1s}, ..., maxA_{ns})$.

Let $tr_{ts}(\tau)$ show 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, $\int_{st}^{et} tr_{ts}(\tau)\, d\tau = |f_s|$ 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 τ, $tr_{ts}(\tau) = min(Maxtr_t, Maxrr_s)$ [bps].

The laxity $l_{fts}(\tau)$ is $|f_s| - \int_{\tau}^{et} tr_{ts}(x)\, dx$ [bit] at time τ, i.e. how many bits of a 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 normalized transmission rate (NTR). Let $\mathbf{F_t}(\tau)$ be a set of current files which the server s_t is transmitting to clients at time τ. Let $C_t(\tau)$ be a set of clients $c_{t1}, ..., c_{tm}$ to which the server s_t transmits files $f_1, ..., f_m$ in $\mathbf{F_t}(\tau)$, respectively, at time τ. First, we consider a model where a server s_t satisfies the following properties:

[Server-bound Model]. If $Maxrr_1 + \cdots + Maxrr_m \geq Maxtr_t$, for every time τ, $\sum_{c_{ts} \in C_t(\tau)} A_{ts}(\tau) = d(\tau) \cdot maxA_t$.
Here, $d(\tau)$ (≤ 1) shows the degradation factor $\gamma^{(1-|C_t(\tau)|)}$ ($0 < \gamma \leq 1$) at time t. Here, the *effective transmission rate* of the server s_t is $d(\tau) \cdot maxA_t$. 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 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. $et_{t1} = st_{t2}$ and $et_{t2} = st_{t3}$ as shown in Figure 2. Here, the transmission time T_t is $et_{t3} - st_{t1} = minT_{tf_1} + minT_{tf_2} + minT_{tf_3}$. 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 Figure 2 (2). Here, since three

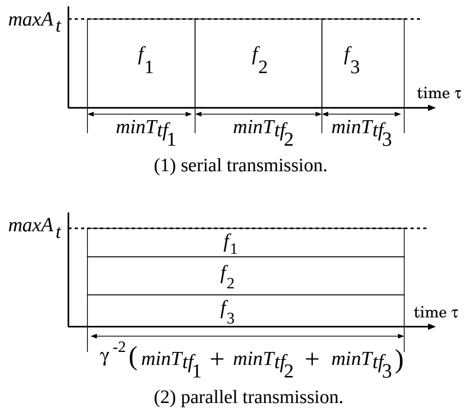

Figure 2: Transmission time.

files are concurrently transmitted, $C_t(t) = 3$ and $\gamma^{-2}T_t = minT_{tf_1} + minT_{tf_2} + minT_{tf_3}$. 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 rate $Maxtr_t$, i.e. $Maxrr_s < Maxtr_t$. Hence, the transmission rate tr_{ts} of the server s_t to a client c_s is $Maxrr_s$.

[Client-bound model]. If $Maxrr_1 + \cdots + Maxrr_m \leq Maxtr_t$, $\sum_{c_{ts} \in C_t(\tau)} A_{ts}(\tau) = maxA_t \cdot (Maxrr_1 + \cdots + Maxrr_m) / Maxtr_t$.

Even if every client c_{ts} receives a file at maximum rate $Maxrr_s$, the effective transmission rate is not degraded.

3 EXPERIMENTAL RESULTS AND POWER CONSUMPTION MODEL

3.1 Environment

We measure how much electric power a computer spends to transfer files to other computers by using the power meter Watts up?.Net (Watts up? .Net, 2009) where the power consumption of each computer can be measured every one second. As shown in Figure 3, a pair of server computers s_1 and s_2 are interconnected with a pair of client computers c_1 and c_2 in 1Gbps networks. Table 1 summarizes the specifications of the servers s_1 and s_2. The server s_1 is equipped with a one-core CPU. The server s_2 is composed of a pair of two-core CPUs. That is, the bandwidth b_{ts} from a server s_t to a client c_s is 1Gbps (t = 1, 2). Each client c_s downloads a file f from one of the servers. The size of the file f is 43,051,806 bytes long. Here, we measure the total power consumption of the servers s_1 and s_2.

For each server s_t, we consider two types of experimentations, one-client ($1C_t$) and two-client ($2C_t$) environments (t = 1, 2). 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.

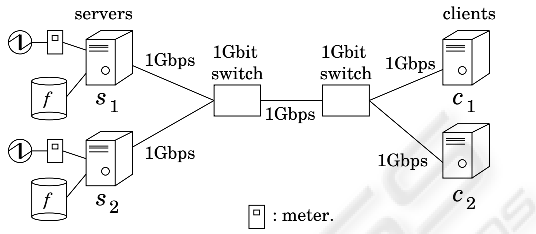

Figure 3: Experimental environment.

3.2 Power Consumption

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/sec]. 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, $tr_1 = tr_{11} + tr_{12}$. Figure 4 shows the power consumption rate of the server s_1 for the total transmission rate tr_1. 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. In Figure 4, the bold dotted line shows the approximated power consumption of the server s_1 where one client downloads the file f from the server s_1. The dotted line shows the approximated power consumption of the server s_1 where a pair of clients c_1 and c_2 concurrently download the file f from the server s_1. Let $PC_1^1(tr)$ and $PC_1^2(tr)$ be the power consumption rates in the $1C_1$ and $2C_1$ environments, respectively, at total rate tr.

$1C_1$: $PC_1^1(tr) = 0.11tr + 4.15$ [W/sec].

$2C_1$: $PC_1^2(tr) = 0.12tr + 4.43$ [W/sec].

In a single-CPU server s_t, the power consumption rate $PC_t(tr)$ is proportional to the total transmission rate tr.

Next, we consider another server s_2 which is composed of a pair of two-core CPUs. Here, the maxi-

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 10000rpm x 2 RAID1
NIC	Broadcom Gbit Ether (1Gbps)	Nvidia Ether Controler (1Gbps)

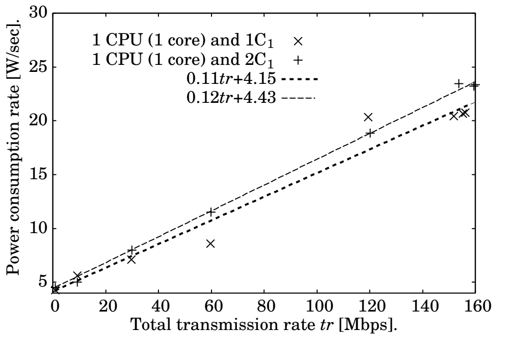

Figure 4: One-CPU : Power consumption rate [W/sec].

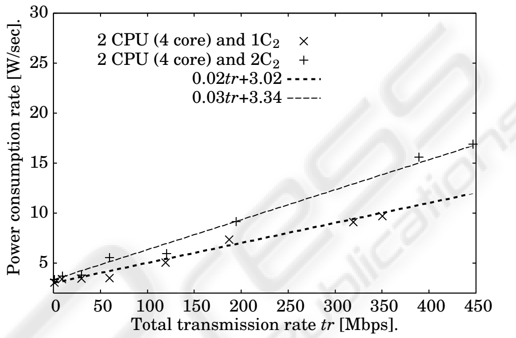

Figure 5: Two-CPU : Power consumption rate [W/sec].

mum transmission rate $Maxtr_2$ of the server s_2 is 450 [Mbps]. We measure the power consumption rate for the total transmission rate tr_2 for $1C_2$ and $2C_2$. Figure 5 shows the power consumption rate [W/sec] of the server s_2 for the total transmission rate tr. Following Figure 5, the power consumption rate of the server s_2 also depends on the total transmission rate tr_2 like $1C_1$. At the higher rate the server s_2 transmits, the larger power consumption s_2 consumes. The approximated formulas $PC_2^1(tr)$ and $PC_2^2(tr)$ of the power consumption rate of the server s_2 for total transmission rate tr [Mbps] are given in the $1C_2$ and $2C_2$ environments as follows:

$1C_2$: $PC_2^1(tr) = 0.02tr + 3.02$ [W/sec].

$2C_2$: $PC_2^2(tr) = 0.03tr + 3.34$ [W/sec].

The increase rate of the power consumption of the server s_2 in $2C_2$ is about 1.5 times larger than $1C_2$. Compared with the one-CPU case $1C_t$, the power consumption rate is not so much increased for the increase of transmission rate in the two-CPU case $2C_t$.

Following the experiments, the power consumption rate $PC_t(tr)$ of a server s_t is lineally increased for transmission rate tr ($0 \leq tr \leq Maxtr_t$) as follows:

$$PC_t(tr) = \beta_t(m) \cdot \alpha_t \cdot tr + minE_t. \qquad (1)$$

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. As shown in Figures 4 and 5, the more number of clients, the more amount of electric power is consumed. $\beta_t(m)$ 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 $maxm_t$ such that $\beta_t(maxm_t - 1) \leq \beta_t(maxm_t) = \beta_t(maxm_t + h)$ for $h > 0$. $minE_t$ gives the minimum power consumption rate of the server s_t where no file is transmitted. $\beta_t(maxm_t) \cdot \alpha_t \cdot Maxtr_t + minE_t$ gives the maximum power consumption rate $maxE_t$ of the server s_t.

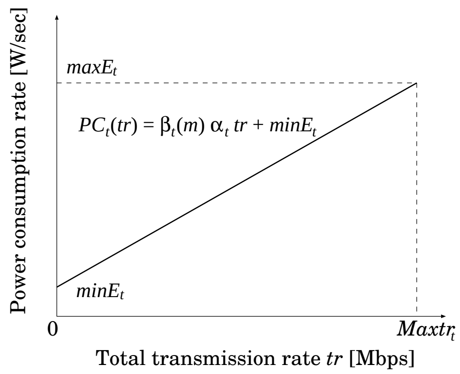

Figure 6: Power consumption rate of server s_t [W/sec].

3.3 Power Consumption Model

We would like to discuss how much electrical power a server s_t consumes to transfer a file to a client c_s. Suppose there are n (≥ 1) servers $s_1, \ldots, s_n$, each of which holds a file f. Let $E_t(\tau)$ show the electric power consumption rate of a server s_t at time τ [W/sec] ($\tau = 1, \ldots, n$). $maxE_t$ and $minE_t$ indicate the maximum and minimum electric power consumption of a server s_t, respectively. Here, $minE_t$ shows the power consumption of a server s_t which is in idle state. That is, $minE_t \leq E_t(\tau) \leq maxE_t$. $maxE$ and $minE$ show $max(maxE_1, ..., maxE_n)$ and $min(minE_1, ..., minE_n)$, respectively.

In this paper, we assume that only file transfer applications are performed on each server. The electric power consumption rate $E_t(\tau)$ of a server s_t at time τ is given as follows:

$$E_t(\tau) = PC_t(tr_t(\tau)). \quad (2)$$

As discussed in the preceding section, $E_t(\tau)$ is given in a linear function (1). $E_t(\tau) = \beta_t(|C_t(\tau)|) \cdot \alpha_t \cdot tr_t(\tau) + minE_t$. Here, $C_t(\tau)$ indicates a set of clients to which a server s_t sends files at time τ.

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

$$TPC_t(\tau_1, \tau_2) = \int_{\tau_1}^{\tau_2} E_t(\tau) d\tau. \quad (3)$$

4 SELECTION ALGORITHMS OF SERVERS

4.1 System Model

There are a set S of multiple servers $s_1, ..., s_n$, each of which holds a full replica of a file f. 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 set S. The server s_t transmits the file f to the client c_s. We discuss how to select a server in the set S for a client c_s so that the following constraints are satisfied:

1. The file f has to be transmitted to the client so as to satisfy the deadline constraint.
2. The power consumption of a selected server s_t to transfer the file f has to be minimized.

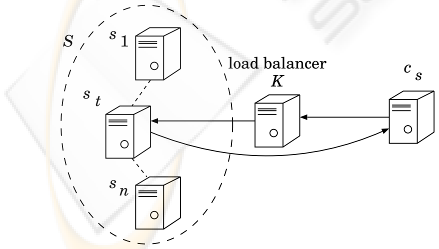

Figure 7: FTP model.

4.2 Round-robin Algorithms

In a load balancer K, types of round-robin algorithms are widely used. In the basic round-robin (RR) algorithm, the servers $s_1, ..., s_n$ in the server set S are totally ordered. A request is first issued to the first server s_1 in the ordered set. If s_1 is overloaded, a request is sent to the second server s_2. Thus, if servers $s_1, ..., s_i$ are overloaded, a request is issued to a server s_{i+1} $(i < n)$.

We further consider weighted round robin (WRR) (Weighted Round Robin (WRR), 1998) and weighted least connection (WLC) (Weighted Least Connection (WLC), 1998) algorithms. For each of the WRR and WLC algorithms, we consider two cases, *Per* (performance) and *Pow* (power). In *Per*, the weight is given in terms of the performance ratio of the servers. That is, the higher performance a server supports, the more number of processes are allocated to the server. On the other hand, the weight is defined in terms of the power consumption rate of the servers in *Pow*. The smaller power a server consumes, the more number of processes are allocated to the server.

4.3 Algorithm for Allocating Transmission Rates

At time τ, the maximum transmission rate $maxtr_t(\tau)$ of a server s_t depends on the degradation factor $d_t(\tau)$ of the server s_t, i.e. the number of 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, $C_t(\tau) = C_t(\tau) + \{c_s\}$ and $C_t(\tau) = C_t(\tau) - \{c_s\}$, respectively. Here, the maximum transmission rate $maxtr_t(\tau)$ of a server s_t at time τ is calculated as $\gamma^{1-|C_t(\tau)|} \cdot Maxtr_t$. Here, $0 < \gamma \leq 1$. The transmission rate $tr_{ts}(\tau)$ of a server s_t for a client c_s at time τ is calculated as follows:

CalcMAXTR_TS(s_t, c_s, τ) {
  check = **False**;
  $maxtr_t(\tau) = \gamma^{1-|C_t(\tau)|} \cdot Maxtr_t$;
  $nc = |C_t(\tau)| + \{c_s\}$;
  /*$C_t(\tau)$ is sorted in ascending order of $Maxrr_s$.*/
  SORT($C_t(\tau)$);
  for each c_i **in** $C_t(\tau)$ {
  /*take a client c_i in the ascending order.*/
    if $Maxrr_i \leq maxtr_t(\tau) / nc$, {
      if $c_i = c_s$, {
        $tr_{ts}(\tau) = Maxrr_i$;
        $maxtr_t(\tau) = maxtr_t(\tau) - tr_{ts}(\tau)$;
        check = **True**;
        break;
      }

$tr_{ts}(\tau) = maxtr_t(\tau)$ - $Maxrr_i$;
$maxtr_t(\tau) = maxtr_t(\tau) - tr_{ts}(\tau)$;
$nc = nc - 1$;
}
} /* for end */
if *check* = **False**, {
$tr_{ts}(\tau) = maxtr_t(\tau) / nc$;
break;
}
return($tr_{ts}(\tau)$);
}

In the procedure **CalcMAXTR_TS**(), each server s_t can transmit a file at least $tr_{ts}(\tau) = maxtr_t(\tau) / |C_t(\tau)|$ [Mbps] to a client c_s in the set $C_t(\tau)$. Here, if the maximum receipt rate $Maxrr_s(\tau)$ of a client c_s is larger than the maximum transmission rate $maxtr_t(\tau)$ allocated for a client c_s, the server s_t transmits a file to the client c_s at rate $tr_{ts}(\tau)$ at time τ. Otherwise, the server s_t transmits at rate $maxrr_s(\tau)$. Here, the unused part of the maximum transmission rate of the server s_t for the client c_s (= $tr_{ts}(\tau)$ - $maxrr_s(\tau)$) can be used for other clients.

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 $maxtr_t(\tau)$ of the server s_t is 6 [Mbps] at time τ as shown in Figure 8. Suppose $Maxrr_1$ = 1 [Mbps], $Maxrr_2$ = 2 [Mbps], and $Maxrr_3$ = 3 [Mbps]. In the basic fair allocation algorithms, each client c_s is allocated with the same transmission rate $tr_{ts}(\tau) = maxtr_t(\tau) / |C_t(\tau)| = 6 / 3 = 2$ [Mbps] as shown in Figure 8 (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 $Maxrr_3$ (= 3 [Mbps]). On the other hand, the unused transmission rate of the client c_1 (= 1 [Mbps]) can be used for the client c_3 in the procedure **CalcMAXTR_TS**(). Then, each client c_s (s = 1, 2, 3) can download files from the server s_t at the maximum receipt rate $Maxrr_s$ at time τ.

4.4 Selection Algorithms

Next, we discuss how a load balancer K selects a server s_t for a client c_s in the server set S. In this paper, we propose two novel allocation algorithms, *transmission rate-based* (*TRB*) and *power consumption-based* (*PCB*) algorithms to select a server for a client. In the TRB algorithm, a server s_t is selected for a client c_s where the transmission rate $tr_{ts}(\tau)$ of the server s_t to transmit a file f to a client c_s is the largest. The TRB algorithm is shown as follows:

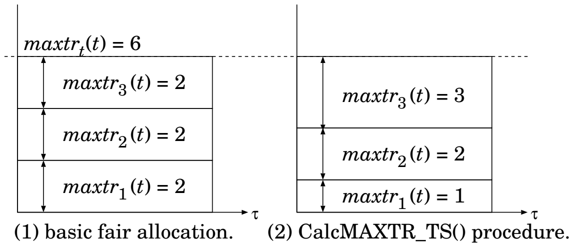

Figure 8: Transmission rate allocation.

TRB(c_s, τ) {
server = ϕ; $MAXTR = 0$;
for each s_t **in** S {
$tr_{ts}(\tau)$ = **CalcMAXTR_TS**(s_t, c_s, τ);
if *server* = ϕ, {
server = s_t;
$MAXTR = tr_{ts}(\tau)$;}
else {
if $MAXTR < tr_{ts}(\tau)$, {
$MAXTR = tr_{ts}(\tau)$;
server = s_t;
}
}
}
return(*server*);
}

In the PCB algorithm, a server s_t is selected for the client c_s where the power consumption to transmit a file f to a client c_s is the smallest. Here, $|f| / tr_{ts}(\tau)$ 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 $tr_{ts}(\tau)$. The power consumption rate $E_{ts}(\tau)$ of each server s_t at time τ is $\beta_t(|C_t(\tau)|) \cdot \alpha_t \cdot tr_{ts}(\tau)$ 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 s_t starts transmitting f is defined as follows:

$$\begin{aligned} EE_{ts}(\tau) & \\ &= (|f| \,/\, tr_{ts}(\tau)) \cdot \beta_t(|C_t(\tau)|) \cdot \alpha_t \cdot tr_{ts}(\tau) \qquad (4) \\ &= |f| \cdot \beta_t(|C_t(\tau)|) \cdot \alpha_t. \end{aligned}$$

Here, a server s_t is selected for a client c_s in the PCB algorithm by using $EE_{ts}(\tau)$ at time τ as follows:

PCB(c_s, τ) {
server = ϕ; $EPC = 0$;
for each s_t **in** S {
$EPC_{ts}(\tau) = |f| \cdot \beta_t(|C_t(\tau)|) \cdot \alpha_t$
if *server* = ϕ, {
server = s_t;

```
      EPC = EE_ts(τ);}
    else {
      if EPC > EE_ts(τ), {
        EPC = EE_ts(τ);  server = s_t;
      }
    }
  return(server);
}
```

For example, there are a pair of servers s_1 and s_2. The maximum transmission rates of the servers s_1 and s_2 are 7 [Mbps] and 6 [Mbps], respectively, i.e. $Maxtr_1$ = 7 [Mbps] and $Maxtr_2$ = 6 [Mbps]. The power consumption coefficients α_1 and α_2 to transmit one [Mbit] for one client of servers s_1 and s_2 are 0.10 and 0.03, respectively. A server s_1 is selected by two clients c_{11} and c_{12} ($C_1(\tau) = \{c_{11}, c_{12}\}$) and another server s_2 is selected by two clients c_{21} and c_{22} ($C_2(\tau) = \{c_{21}, c_{22}\}$) at time τ, respectively. The maximum receipt rates of clients c_{11} and c_{21} are the same 1 [Mbps] ($Maxrr_{11} = Maxrr_{21}$ = 1 [Mbps]). The maximum receipt rates of clients c_{12} and c_{22} are the same 2 [Mbps] ($Maxrr_{12} = Maxrr_{22}$ = 2 [Mbps]). Suppose a client c_3 issues a new request to transmit a file f whose size is ten Mbytes to a load balancer K at time τ. Here, the maximum receipt rate $Maxrr_3$ for the file f on the client c_3 is 4 [Mbps]. According to the procedure **CalcMAXTR_TS**$(\cdots)$, the unused transmission rates of the servers s_1 and s_2 are 4 [Mbps] and 3 [Mbps], respectively. The servers s_1 and s_2 can allocate transmission rates 4 [Mbps] and 3 [Mbps] for the client c_3, respectively. In the TRB algorithm, a server s_t which can allocate the maximum transmission rate to a client c_3 is selected. Therefore, the server s_1 is selected for a client c_3. On the other hand, 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, i.e. a server which can mostly save the power consumption is selected at time τ. Here, sets $C_1(\tau)$ and $C_2(\tau)$ of current clients of servers s_1 and s_2 include three clients, respectively. Suppose the increasing rates $\beta_1(3)$ and $\beta_2(3)$ of the power consumption of the servers s_1 and s_2 are 1.2 and 1.09, respectively. Here, $|f|\cdot\beta_1(3)\cdot\alpha_1 = 10 \cdot 1.2 \cdot 0.10 = 1.2$. $|f|\cdot\beta_2(3)\cdot\alpha_2 = 10 \cdot 1.09 \cdot 0.03 = 0.327$. Therefore, the server s_2 is selected for a client c_3 in the PCB algorithm.

5 EVALUATION

5.1 Evaluation Environment

We evaluate the TRB and PCB algorithms in terms of the total amount of power consumption and total transmission time of files compared with the basic RR algorithm 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 2, $S = \{s_1, s_2, s_3, s_4, s_5\}$. 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 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 replica of a file f. The size 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 0.1 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 [sec] at the simulation time. Each client c_s issues one request at time st_s in the simulation. In the simulations of the TRB, PCB, and RR algorithms, the starting time st_s of the file transmission to each client c_s is the same.

Table 2: Types of servers.

Servers	α	$\beta(m)$	$minE$ [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

5.2 Total Power Consumption

Figure 9 shows the total power consumption rate [W/sec] of the servers s_1, ..., s_5 at each time. Table 3 shows the total power consumptions of the servers in the PCB, TRB, and RR algorithms. The total power consumptions of the PCB, TRB, and RR algorithms are 546,186 [W], 654,161 [W], and 1,073,914 [W], respectively. In the PCB algorithm, the total amount of power consumption is the smallest because a server s_t is selected for a client c_s, whose the power consumption is the smallest to transmit a file f to the client c_s. In the TBR algorithm, a 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 PCB algorithm. On the other hand, the total amount of power consumption of the TRB algorithm is smaller than the RR algorithm.

Table 3: Total amount of power consumption.

PCB	TRB	RR
546,186 [W]	654,161 [W]	1,073,914 [W]

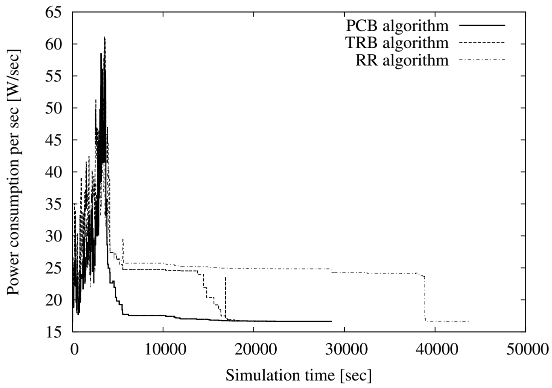

Figure 9: Total power consumption rate.

5.3 Total Transmission Time

Table 4 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 [sec], 28,594 [sec], and 43,744 [sec] 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.

Table 4: Total transmission time of the files.

PCB	TRB	RR
28,614 [sec]	28,594 [sec]	43,744 [sec]

In the PCB algorithm, 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 PCB algorithm than the TRB algorithm and the difference of the total transmission time between the 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. 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.

5.4 File Size

We measured the total transmission time [sec] and the total power consumption of the PCB, TRB, and RR algorithms to transmit five types of files whose sizes are 1, 2, 3, 4, and 5 [GB], respectively. Totally the 100 clients download the file f from one s_t of the servers in the server set S (= $\{s_1, s_2, s_3, s_4, s_5\}$). Table 5 and Figure 10 show the total transmission time in the PCB, TRB, and RR algorithms for each file size. The total transmission time of the RR algorithm is longer than the PCB and TRB algorithms.

Figure 11 shows the total transmission time in the PCB and TRB algorithms for file size. The total transmission time of the TRB algorithm is smaller than the PCB and RR algorithms. The difference of the total transmission time between the PCB algorithm and the TRB algorithm is almost neglectable.

Table 5: Total transmission time [sec].

	PCB	TRB	RR
1GB	28,614	28,594	43,744
2GB	55,599	55,521	430,248
3GB	82,585	82,354	1,722,061
4GB	109,570	108,958	4,570,858
5GB	136,556	135,036	7,276,956

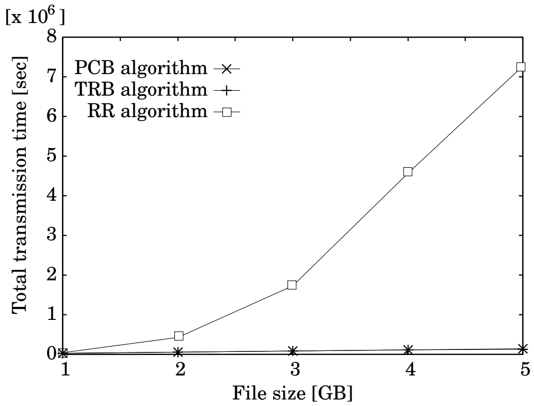

Figure 10: Total transmission time (PCB, TRB, and RR).

Next, we measured the total power consumption [W] of the five servers s_1, ..., s_5 in the PCB, TRB, and RR algorithms. Table 6 and Figure 12 show the total

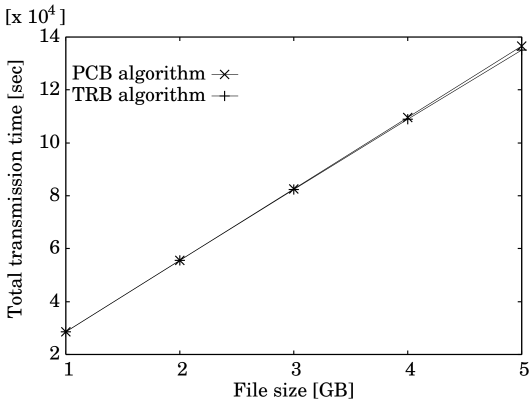

Figure 11: Total transmission time (PCB and TRB).

amount of power consumption in the PCB, TRB, and RR algorithms for each file size. Figure 13 shows the total amount of power consumption of the servers in the PCB and TRB algorithms. The total amount of power consumption of the PCB algorithm is smaller than the TRB and RR algorithms. The total amount of power consumption of the TRB algorithm is smaller than the RR algorithm. The PCB algorithm is better than the TRB and RR algorithms for any file size.

Table 6: Total power consumption [W].

	PCB	TRB	RR
1GB	546,186	654,161	1,073,914
2GB	1,209,621	1,405,971	10,647,775
3GB	1,971,767	2,215,575	42,670,592
4GB	2,835,375	3,131,209	113,334,809
5GB	3,998,291	4,211,034	180,500,814

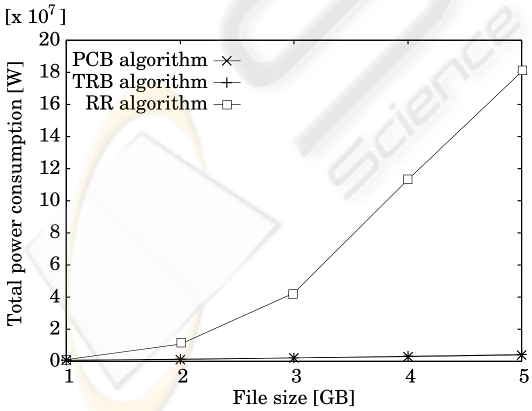

Figure 12: Total power consumption (PCB, TRB, and RR).

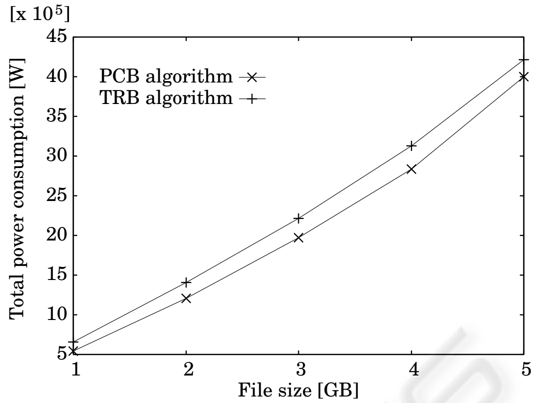

Figure 13: Total power consumption (PCB and TRB).

6 CONCLUDING REMARKS

In this paper, we discussed how much electric power a server consumes to transfer a file to a client. A server consumes the electric power proportional to the transmission rate. Through the experiments, we obtained approximate linear functions showing how much a server computer consumes the electric power to transmit files to clients for transmission rate. We proposed the PCB and TRB algorithms to select a server so that the total power consumption of the servers is reduced. We evaluated the PCB and TRB algorithms in terms of the total power consumption and the total transmission time compared with the basic RR algorithm through simulation. According to the evaluation results, 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. It is not necessary to estimate the transmission rate between a server and a client in the PCB algorithm. In addition, the total power consumption and total transmission time are no increased in the PCB and TRB algorithms compared with the RR algorithm even if the file size is increased. Therefore, the PCB algorithm is more useful for reducing the total power consumption in the communication-based applications.

REFERENCES

Aikebaier, A., Enokido, T., and Takizawa, M. (2009). Energy-efficient computation models for distributed systems. In *Proceedings of the 12th International*

Conference on Network-Based Information Systems (NBiS2009), pages 29–36. IEEE CS Press.

Akyildiz, I. F. and Kasimoglu, I. H. (2004). Wireless sensor and actor networks: Research challenges. In *Ad Hoc Networks journal*, volume 2, pages 351–367. Elsevier.

Bianchini, R. and Rajamony, R. (2004). Power and energy management for server systems. In *IEEE Computer*, volume 37(11), pages 68–76. IEEE CS Press.

Enokido, T., Aikebaier, A., and Takizawa, M. (2010a). A model for reducing power consumption in peer-to-peer systems. In *accepted at IEEE Systems Journal*. IEEE CS Press.

Enokido, T., Suzuki, K., Aikebaier, A., and Takizawa, M. (2010b). 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)*, pages 321–328. IEEE CS Press.

Ghemawat, S., Gobioff, H., and Leung, S. (2003). The google file system. In *Proceedings of the 19th ACM symposium on Operating systems principles (SOSP'03)*, pages 29–43. ACM.

Green IT (2010). http://www.greenit.net.

Grossman, R. L. (2009). The case for cloud computing. In *IT Professional*, pages 23–27. IEEE CS Press.

Heath, T., Diniz, B., Carrera, E. V., Meira, W., and Bianchini, R. (2005). Energy conservation in heterogeneous server clusters. In *PPoPP '05: Proceedings of the tenth ACM SIGPLAN symposium on Principles and Practice of Parallel Programming*, pages 186–195. ACM.

Rajamani, K. and Lefurgy, C. (2003). On evaluating request-distribution schemes for saving energy in server clusters. In *Proceedings of the 2003 IEEE International Symposium on Performance Analysis of Systems and Software*, pages 111–122. IEEE CS Press.

Watts up? .Net (2009). https://www.wattsupmeters.com/secure/products.php ?pn=0.

Weighted Least Connection (WLC) (1998). http://www.linuxvirtualserver.org/docs/scheduling.html.

Weighted Round Robin (WRR) (1998). http://www.linuxvirtualserver.org/docs/scheduling.html.

Yang, T., Barolli, L., Ikeda, M., Marco, G. D., Xhafa, F., and Miho, R. (2009a). Performance evaluation of a wireless sensor network considering mobile event. In *Proceedings of the 3rd International Conference on Complex, Intelligent and Software Intensive Systems (CISIS2009)*, pages 1169–1174. IEEE CS Press.

Yang, Y., Xiong, N., Aikebaier, A., Enokido, T., and Takizawa, M. (2009b). Minimizing power consumption with performance efficiency constraint in web server clusters. In *Proceedings of the 12th International Conference on Network-Based Information Systems (NBiS2009)*, pages 45–51. IEEE CS Press.

Zhang, L. and Zhou, Q. (2009). Ccoa: Cloud computing open architecture. In *2009 IEEE International Conference on Web Services*, pages 607–616. IEEE CS Press.

Zhao, G., Xuan, K., Rahayu, W., Taniar, D., Safar, M., Gavrilova, M., and Srinivasan, B. (2010). Voronoi-based continuous k nearest neighbor search in mobile navigation. *accepted for publication in IEEE Trans. on Industrial Electronics*.