PAPER *Special Section on New Generation Network towards Innovative Future Society*

Dynamic and Decentralized Storage Load Balancing with Analogy to Thermal Diffusion for P2P File Sharing*

Masato UCHIDA[†a)], Kei OHNISHI[††b)], Kento ICHIKAWA[†c)], Masato TSURU[††d)], *Members*, *and* Yuji OIE[††e)], *Fellow*

SUMMARY In this paper we propose a file replication scheme inspired by a thermal diffusion phenomenon for storage load balancing in unstructured peer-to-peer (P2P) file sharing networks. The proposed scheme is designed such that the storage utilization ratios of peers will be uniform, in the same way that the temperature in a field becomes uniform in a thermal diffusion phenomenon. The proposed scheme creates replicas of files in peers probabilistically, where the probability is controlled by using parameters that can be used to find the trade-off between storage load balancing and search performance in unstructured P2P file sharing networks. First, we show through theoretical analysis that the statistical behavior of the storage load balancing controlled by the proposed scheme has an analogy with the thermal diffusion phenomenon. We then show through simulation that the proposed scheme not only has superior performance with respect to balancing the storage load among peers (the primary objective of the present proposal) but also allows the performance trade-off to be widely found. Finally, we qualitatively discuss a guideline for setting the parameter values in order to widely find the performance trade-off from the simulation results.

key words: *P2P file sharing networks, storage load balancing, thermal diffusion*

1. Introduction

Recently, peer-to-peer (P2P) network models have attracted a great deal of attention. The concept of the P2P network model is completely different from that of a conventional client-server network model. While a conventional server-client network model explicitly distinguishes hosts providing services (servers) from hosts receiving services (clients), a P2P network model does not assign fixed roles to hosts. Hosts composing P2P networks, referred to as *peers*, can be both servers and clients, so that P2P networks can theoretically function as an autonomous, distributed, and cooperative system.

One of application of P2P networks that is of interest is a distributed storage system for file sharing. A distributed storage system for file sharing provides a large amount of storage by accumulating the unused storage of hosts, which enables large amounts of data to be stored and shared without the need for a costly file server. Although there exist several forms of P2P networks for file sharing [2], in the present study we focus on unstructured P2P networks that do not have a mechanism to manage file locations.

Since unstructured file sharing P2P networks do not have a mechanism to manage file locations, a query distribution mechanism is needed to find requested files. An approach to enhance file search performance of the query distribution mechanism is to make replicas of files so as to increase the total number of files in the network. Although further enhancement of file search performance can be achieved by making replicas of files in specific peers through which search queries pass frequently, such behavior causes the bias of storage load to the specific peers. Also, since storage load balancing in the networks means that several files are not put on specific peers through which search queries pass frequently, and therefore, it leads to the degradation of file search performance. Therefore, there must be a trade-off between storage load balancing and search performance in unstructured file sharing P2P networks. This trade-off should be taken into consideration for controlling unstructured file sharing P2P networks.

The present paper focuses on a file replication scheme. Although an original purpose of the file replication is to improve file search performance, the present paper considers the cancellation of storage load bias caused as a counteraction of the improvement of file search performance. We propose a file replication scheme that considers storage load balancing inspired by nature, in which the order appears to be maintained in an autonomous and distributed manner. The statistical behavior of the proposed file replication scheme has an analogy with the phenomenon of thermal diffusion. Furthermore, the proposed scheme includes parameters to widely find the trade-off between storage load balancing and search performance existing in unstructured P2P file sharing networks. We conduct a theoretical analysis to reveal an analogy between the proposed replication scheme and a thermal diffusion equation, and experimentally examine the ability of the proposed scheme not only to balance the storage load among peers but also to find the performance trade-off. The ability to widely find the trade-off relationship is important because it is impossible to uniquely define the universally best trade-off point due to various user

Manuscript received July 31, 2009.
Manuscript revised November 5, 2009.

[†]The authors are with Network Design Research Center, Kyushu Institute of Technology, Tokyo, 100-0011 Japan.
[††]The authors are with the Dept. of Computer Science & Electronics, Kyushu Institute of Technology, Iizuka-shi, 820-8502 Japan.

*The present paper is an extended version of our previous paper that was presented at Inter-Perf 2006 [1].

a) E-mail: m.uchida@ndrc.kyutech.ac.jp
b) E-mail: ohnishi@cse.kyutech.ac.jp
c) E-mail: ichikawa@ndrc.kyutech.ac.jp
d) E-mail: tsuru@ndrc.kyutech.ac.jp
e) E-mail: oie@cse.kyutech.ac.jp

DOI: 10.1587/transcom.E93.B.525

requirements and operated policies for storage load balancing and search performance.

The present paper is organized as follows. Section 2 briefly describes related research. Section 3 proposes a file replication scheme inspired by thermal diffusion and shows analytically that the proposed scheme has an analogy with a thermal diffusion equation. The proposed scheme is experimentally evaluated in Sects. 4 and 5. Finally, Sect. 6 presents the conclusions obtained through the analysis.

2. Related Studies

Replication schemes have been developed mainly for *unstructured* P2P networks that do not have a means of managing file locations [3], [4]. This is because searching in such networks is blind, and replicas of files distributed over the networks should contribute to quick and reliable file searching. However, these studies focused only on search efficiency and did not consider load imbalance among peers as a trade-off for search efficiency. On the other hand, load balancing schemes have been developed mainly for *structured* P2P networks that have a means of managing file locations such as a distributed hash table (DHT) [5], [6]. This is because fixed file locations provided by P2P using DHT can be the cause of load imbalance among peers, depending on, for example, how the identifier space is constructed, how the subspace is assigned to peers, and differences in popularity between files.

Our previous study [7] focused on replication schemes in unstructured P2P file sharing networks. Our objective was not only to achieve good search performance, as in previous studies on unstructured P2P networks, but also to achieve storage load balancing, as in previous studies on structured P2P networks. Here, the storage load of a peer is defined as the *storage utilization ratio* which is the ratio of the consumed capacity to the total capacity. These replication schemes were designed mainly to achieve storage load balancing, but by adjusting their parameter values, we found the trade-off between search performance and storage load balancing. In particular, we proposed three different replication schemes in our previous paper: Path Random Replication, Path Adaptive Replication, and Path Adaptive Replication with Priority Level. All these schemes are based on Path Replication that creates replicas of files in peers on the search path (with probability 1). However, these schemes are all probabilistic replication schemes that create replicas of files in peers on the search path as well as Path Replication and differ in how they decide the probability with which file replicas are created, where the search path is formed by random walk-based query forwarding mechanism.

The replication scheme proposed in the present paper also creates file replicas in peers on the search path probabilistically, where the search path is formed by random walk-based query forwarding mechanism. The proposed scheme is mainly intended to achieve storage load balancing, but also to find a wide range of trade-off points between search performance and storage load balancing by adjusting its parameter values. The proposed scheme is based on Path Adaptive Replication (**PAR**). The key concept of PAR is that the probability with which a replica of a requested file is created in a peer at location x on the search path is determined using the storage utilization ratio of the peer at time t, $L(x,t)\,(0 \leq L(x,t) \leq 1)$. This probability is hereinafter referred to as the *file replication ratio*. The file replication ratio is given by the following equation:

$$P(x,t) = 1 - \frac{1 - e^{-\xi L(x,t)}}{1 - e^{-\xi}}, \tag{1}$$

where ξ is a tunable parameter to vary the average value of replication ratios $M \in [0,1]$, and ξ is obtained from M by evaluating $M = \int_0^1 P(x,t)dL$. In addition, $P(x,t)$ satisfies the following conditions:

- When the storage load of the peer is high, the replication ratio becomes low
- When the storage load of the peer is low, the replication ratio becomes high

This scheme allows each peer to independently control storage utilization by means of changing the file replication ratio to achieve fair storage utilization among all of the peers. Parameter ξ is adjusted not only for storage load balancing, but also for better search performance, and therefore for exploring the trade-off between storage load balancing and search performance.

Unlike PAR, the file replication scheme proposed in the present paper is designed to use the storage utilization ratios of peers adjacent to the focus peer, thus achieving network-wide storage load balancing. More specifically, the proposed scheme determines the file replication ratio in the focus peer by using the difference between its storage utilization ratio and those of its neighboring peers. Here, neighboring peers are defined as ones 1-hop away from the focus peer.

Dynamic load balancing schemes with only local peer communication as in the proposed replication system, referred to as diffusive load balancing schemes, have been investigated primarily with respect to distributed computing [8]–[10]. These systems basically rely on a diffusion equation. Load balancing in distributed computing usually results in greater computing efficiency. Since maximum computing efficiency is the only primary objective, no trade-off exists. However, in P2P file sharing networks, better load balancing can cause deterioration in the primary objective of search efficiency, resulting in a trade-off. This suggests that load balancing schemes used in distributed computing, including parameter tuning schemes, might not be suitable for direct application to load balancing in P2P file sharing networks. In addition, while conventional diffusive load balancing schemes ‘diffuse’ the direct causes of load, such as computational tasks, among nodes, the proposed replication scheme does not use such a direct approach to achieve load balancing.

3. Storage Load Balancing Inspired by Thermal Diffusion

3.1 Motivation

Search performance must be enhanced in unstructured P2P file sharing networks, because they have no file location management system. This may be achieved by replicating files over the network, allowing a query forwarding mechanism to find requested files quickly and reliably. This is referred to as a file replication scheme. One of the simplest file replication schemes is Path Replication that creates a replica of a requested file on the current search path provided by a query forwarding method [3], [4].

For query forwarding, random walk-based methods are often employed for unstructured P2P file sharing networks because of their ease of implementation. Considering a random walk on an arbitrary network, the probability with which a random walker stays at a node is shown to be proportional to the degree of the peer [11]. Therefore, random walk-based query forwarding methods are likely to include peers of high degree in the paths that they create. If replication schemes that create a replica of a requested file on the current search path are used together with random walk-based query forwarding methods, more files are replicated in high-degree peers than in low-degree peers. The advantage of Path Replication with random walk-based query forwarding for enhancing search performance has been discussed in [4]. Although this situation is good for enhancing search performance, the storage load would be biased to higher degree peers, and as a result, the P2P network might be unstable and unreliable. Of course, the disadvantage on the storage load bias would be increased when flooding-based query forwarding is used.

Considering into the above mentioned advantage and disadvantage, as in our previous study [7], in the present study, we attempt to find a file replication scheme based on Path Replication that achieves storage load balancing using random walk-based query forwarding method. Here, the definition of load balance in this paper is allocating load uniformly independent of the number of edges a node has. Since file search in unstructured file sharing P2P networks function in an autonomous and distributed manner, it is desirable for storage load balancing to function in the same manner.

With respect to autonomous and distributed storage load balancing, the natural world is a great source of inspiration because nature itself appears to be ordered in this way. We can therefore look at natural physical phenomena, especially the phenomenon of diffusion, and make an analogy between the mechanism of diffusion and storage load balancing. Diffusion is a common physical phenomenon, manifesting itself as heat diffusion through a steel plate or a drop of colored ink spreading through water.

Next, we will examine the phenomenon of thermal diffusion, and consider the analogy between thermal diffusion and the proposed storage load balancing scheme. When heat diffuses through a substance having a thermal gradient, the diffusion is governed only by the local thermal gradient and conductivity, according to a differential equation describing thermal diffusion. This implies that thermal diffusion occurs only through linked local behaviors based on the local thermal gradient. An analogy may be made between thermal diffusion and storage load balancing by file replication in unstructured P2P file sharing networks, by considering the steel plate to be a P2P network, considering a part of the plate to be several peers in the network, and considering heat to be the storage load of the peers. In addition, file replication corresponds to heating of the steel plate. Whereas in thermal diffusion, heat provided by heat sources diffuses through a substance, in the storage load balancing model, storage load ‘diffuses’ through peers on the current search path.

However, there are several differences between file replications for storage load balancing and thermal diffusion. For instance, whereas thermal diffusion is a phenomenon in which the entire steel plate is involved, the file replication locations considered herein are limited to peers on search paths. In the following we will present a method by which to apply the analogy of thermal diffusion to storage load balancing in P2P networks while taking this difference into account.

3.2 Basic Concept

In thermal diffusion, every local non-uniform distribution of temperatures becomes a uniform distribution. Similarly, we suppose that in storage load balancing, every local non-uniform distribution of storage utilization ratios becomes a uniform distribution based on the difference in storage utilization ratios between the focus peer and its neighboring peers.

Suppose that a network simply consists of three peers A, B, and C, and logical links are made between peers B and A and between peers A and C, and are therefore aligned in the order B, A, C. Next, we introduce the following expression, which represents the difference between the storage utilization ratio of peer A and the average storage utilization ratio of its neighboring peers B and C:

$$\frac{L(B,t)+L(C,t)}{2}-L(A,t) \tag{2}$$

Expression (2) suggests that the lower the storage utilization of peer A (second term) compared to the average storage utilization of its neighboring peers (first term), the bigger the value, and vice versa. In addition, Expression (2) takes its largest value 1 when $L(B,t) = L(C,t) = 1$ and $L(A,t) = 0$ and its smallest value -1 when $L(B,t) = L(C,t) = 0$ and $L(A,t) = 1$. Thus, we can measure the relationship between the storage utilization ratio of peer A and the average of the storage utilizations of its neighboring peers B and C by means of Expression (2).

In terms of storage load balancing, the creation of replicas of files in peer A should be actively performed when the value of Expression (2) becomes larger, and vice versa. If we consider only the storage utilization ratios of individual peers, we cannot embed the above storage load balancing strategy in a file replication scheme. Thus, for better storage load balancing, it is desirable that the determination of whether or not replicas of files are created in peers must be done by taking into account the relationship between the storage utilization ratio of a focus peer and its neighboring peers.

Based on the above discussion, we will consider below an actual method for observing the difference between the storage utilization ratio of a focus peer and its neighboring peers.

First, we introduce an operator $\mathbf{D}$ to $L(x,t)$. Although the operator $\mathbf{D}$ is used only with $L(x,t)$ in this paper, the introduction of the operator is helpful for understanding an interesting meaning of the proposed scheme (see Sect. 3.4). The operator $\mathbf{D}$ derives a new function $\mathbf{D} \cdot L(x,t)$ from the function $L(x,t)$, where $\mathbf{D} \cdot L(x,t)$ is defined as the difference between the storage utilization ratio of the peer and the average of its neighboring peers. That is, $\mathbf{D} \cdot L(x,t)$ is defined as the following equation:

$$\mathbf{D} \cdot L(x,t) = \frac{\sum_{i=1}^{d(x)} L(x_i,t)}{d(x)} - L(x,t), \tag{3}$$

where x_i ($i = 1, \ldots, d(x)$) denotes a peer adjacent to peer x and $d(x)$ denotes the degree of peer x. Equation (3) with $d(x) = 2$ is equivalent to Expression (2). As with Expression (2), $\mathbf{D}{\cdot}L(x,t)$ takes (i) its largest value 1 when the storage utilization ratio of peer x is 0 (i.e., $L(x,t) = 0$) and the storage utilization ratios of all of its neighboring peers are 1 (i.e., $L(x_i,t) = 1$) and takes (ii) its smallest value -1 when the storage utilization ratio of peer x is 1 (i.e., $L(x,t) = 1$) and the storage utilization ratios of all of its neighboring peers are 0 (i.e., $L(x_i,t) = 0$). This range of values is represented by the following inequality:

$$-1 \leq \mathbf{D} \cdot L(x,t) \leq 1. \tag{4}$$

In the present paper, we propose a scheme that adjusts the replication ratio $P(x,t)$ of peer x using $\mathbf{D} \cdot L(x,t)$. More specifically, the replication ratio $P(x,t)$ is designed to meet the following conditions:

- When the storage utilization ratio of peer x is lower than its neighboring peers (i.e., the value of $\mathbf{D} \cdot L(x,t)$ is large), the replication ratio $P(x,t)$ becomes higher
- When the storage utilization ratio of peer x is higher than its neighboring peers (i.e., the value of $\mathbf{D} \cdot L(x,t)$ is small), the replication ratio $P(x,t)$ becomes lower

In the next section, we will provide several example definitions of $P(x,t)$ that meet the above conditions.

3.3 Replication Scheme for Storage Load Balancing

As a simple example, we consider a replication ratio $P(x,t)$ that increases linearly with $\mathbf{D} \cdot L(x,t)$. In this case, $P(x,t)$ is represented by the following equation:

$$P(x,t) = \frac{1}{2} + \frac{1}{2}\mathbf{D} \cdot L(x,t), \tag{5}$$

where $0 \leq P(x,t) \leq 1$. Equation (5) suggests that $P(x,t)$ becomes high when the average storage utilization of peers adjacent to peer x is higher than that of peer x, and vice versa. In Sect. 3.5, we will show that the storage distribution described with this replication ratio has basic similarities to thermal diffusion.

Equation (5) gives a linear variation of $P(x,t)$ for all values of $\mathbf{D} \cdot L(x,t)$. However, from a practical point of view, $P(x,t)$ should be more flexible, such that the variation of $P(x,t)$ is more sensitive to the variation of $\mathbf{D} \cdot L(x,t)$ over certain areas of its range. For example, Eq. (5) can be modified as follows:

$$P(x,t) = \frac{1}{2} + \frac{1}{2}\tanh\left(\mu + \lambda \tanh^{-1} \mathbf{D} \cdot L(x,t)\right), \tag{6}$$

where Eq. (6) with $\lambda = 1$ and $\mu = 0$ is equal to Eq. (5). Graphs of Eq. (6) with various sets of values of (μ, λ), which will be used for simulation experiments in Sect. 4, are shown in Fig. 1. As shown in Fig. 1, λ is a parameter that can be used to adjust the sensitivity of $P(x,t)$ to $\mathbf{D} \cdot L(x,t)$. In addition, μ is a parameter used to determine the replication ratio of peers not only after perfect storage load balancing is achieved, but also when $\lambda = 0$, because from Eq. (6) the replication ratio ρ for the case in which $\mathbf{D} \cdot L(x,t) = 0$ for all of the peers or $\lambda = 0$ is as follows:

$$\rho = \frac{1}{2} + \frac{1}{2}\tanh(\mu) \tag{7}$$

In particular, when $\lambda = 0$, the proposed replication scheme is equivalent to Path Random Replication with a replication ratio ρ, where Path Random Replication creates a replica of a requested file in peers on the current search path with a fixed replication ratio.

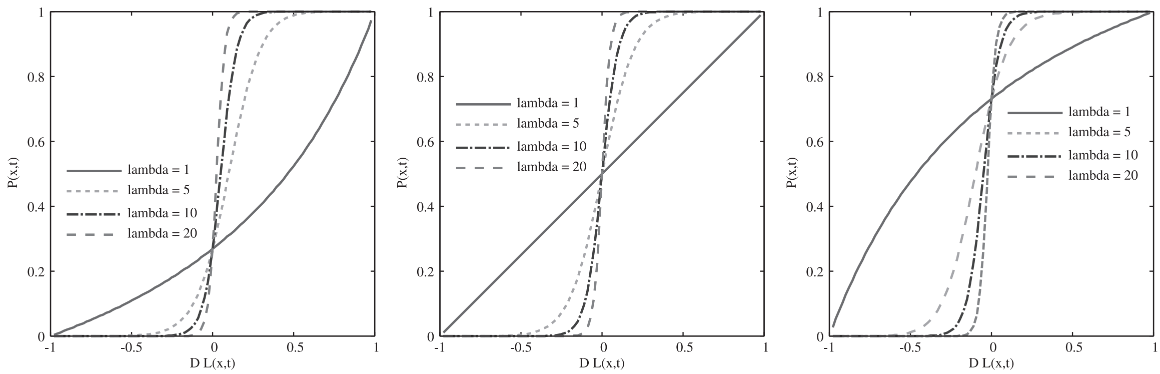
3.4 Relationship between Operator D and Laplacian

Here, we will show that the operator $\mathbf{D}$ can be regarded as a sort of Laplacian, which is a second-order differential. This parallel is fundamental in discussions concerning the analogy between the proposed file replication scheme and thermal diffusion.

We assume the same simple network as that shown in Sect. 3.2. Equation (2) can be transformed into the following equation:

$$\begin{aligned} &\frac{L(B,t) + L(C,t)}{2} - L(A,t) \\ &\quad = \frac{1}{2}\{(L(B,t) - L(A,t)) - (L(A,t) - L(C,t))\} \end{aligned} \tag{8}$$

This equation suggests that Eq. (2) corresponds to a (discrete) second-order derivative of $L(x,t)$ on x when we assume a row of peers A, B, and C to be an axis. The important point is that the three peers A, B, and C are aligned, and



(a) $\mu = -0.5$. (b) $\mu = 0.0$. (c) $\mu = 0.5$.

Fig. 1 $P(x,t)$ with a variety of sets of values of (μ, λ), as defined by Eq. (6).

because of this Eq. (2) can be regarded as a second-order derivative on the axis including peers A, B, and C.

Next, we assume more general P2P networks, in which connection between peers is not limited to a series connection. In this case, $\mathbf{D} \cdot L(x,t)$ is represented by the following equation:

$$\begin{aligned}\mathbf{D} \cdot L(x,t) &\\ &= \frac{\sum_{i=1}^{d(x)} L(x_i)}{d(x)} - L(x) \\ &= \frac{1}{d(x)(d(x)-1)} \sum_{i<j} \{(L(x_i) - L(x)) + (L(x_j) - L(x))\}\end{aligned} \tag{9}$$

This equation suggests that when we consider a row of x_i, x, and x_j (x_i-x-x_j) with respect to all of the combinations of i and j ($i < j$) to be an axis, $\mathbf{D} \cdot L(x,t)$ represents the sum of (discrete) second-order derivatives on $x_{i,j}$ in $L(x,t)$, where $x_{i,j}$ corresponds to the axis of x_i-x-x_j. This means that the continuous representation of operator $\mathbf{D}$ is given by the $d(x)(d(x)-1)$-th-order Laplacian:

$$\sum_{i<j} \frac{\partial^2}{\partial x_{i,j}^2}. \tag{10}$$

Note that the replication ratio $P(x,t)$ defined in the previous section becomes high when $\mathbf{D} \cdot L(x,t) > 0$ and low when $\mathbf{D} \cdot L(x,t) < 0$. In other words, the replication ratio $P(x,t)$ becomes high when the storage utilization ratio on a P2P network is convex at peer x, compared to its neighboring peers, and becomes low when the storage utilization ratio on a P2P network is concave at peer x, compared to its neighboring peers.

3.5 Analogy to Thermal Diffusion

In this section, we will analytically show that the proposed replication scheme is analogous to a thermal diffusion equation. The analogy is validated by the fact that an essential term in a thermal diffusion equation appears in a statistical expression for the proposed replication scheme. Furthermore, a term representing a continuously active heat source also appears in the statistical expression. The heat source term indicates that the total amount of storage load in a network increases steadily. We will discuss these points in detail later.

We will define several symbols prior to the analysis. First, let $\mathbb{E}[X]$ be an expectation value for random variable X, and let $\mathbb{E}[X|Y]$ be a conditional expectation value for random variable X when random variable Y is given.

We will define further random variables by assuming in the following analysis that file search in a P2P network is stochastic (non-deterministic). Let $L(x,t)$ be a random variable that represents the storage utilization ratio of peer x at time t. The analysis is valid only when $0 \leq L(x,t) \leq 1$. In addition, let $R(f)$ be a random variable that represents the size of file f, which can be the target of a search. Furthermore, let $I(x,t)$ be a random variable that indicates whether peer x is on the current search path at time t. That is, $I(x,t)$ is 1 when it is on the current search path, and 0 otherwise.

Therefore, the product of two random variables $R(f)$ and $I(x,t)$ can be interpreted as the rise of the storage utilization of peer x at time t for the case in which a stochastic file search is performed in a certain network. The product value depends on the network topology and the file search path (depending on the file search method). Finally, let $C(x)$ be a random variable that represents the storage capacity of peer x.

The analysis begins with calculation of the storage utilization ratio at time $t + \Delta t$, i.e., a unit of time Δt after time t. The result is as follows:

$$\begin{aligned}&\mathbb{E}[L(x, t+\Delta t)|L(x,t), R(f), I(x,t), C(x)] \\ &\quad = L(x,t) + P(x,t)\frac{R(f)I(x,t)\Delta t}{C(x)}\end{aligned} \tag{11}$$

This equation holds for any $P(x,t)$. Substituting Eq. (5) for $P(x,t)$ in Eq. (11), we obtain the following equation:

$$\frac{\mathbb{E}[L(x,t+\Delta t)|L(x,t), R(f), I(x,t), C(x)] - L(x,t)}{\Delta t}$$

$$= \frac{R(f)I(x,t)}{2C(x)} + \frac{R(f)I(x,t)}{2C(x)}\mathbf{D} \cdot L(x,t). \quad (12)$$

The simplest representation of a thermal diffusion equation when we let $T(x,t)$ be the temperature of position x at time t is as follows:

$$\frac{\partial T(x,t)}{\partial t} = K\frac{\partial^2}{\partial x^2}T(x,t), \quad (13)$$

where K is a positive constant value called diffusion coefficient. The right-hand side of this equation has the same form as the second term in the right-hand side of Eq. (12), which implies that the proposed replication scheme is able to balance the storage load among peers. In addition, the first term in Eq. (12) implies that an effect similar to a constant heat source exists in the P2P network.

If we substitute Eq. (6), in place of Eq. (5), for $P(x,t)$ in Eq. (11), and apply a Taylor expansion to the substituted Eq. (11) for $\mathbf{D} \cdot L(x,t) \sim 0$ and $\mu \sim 0$, we obtain the first-order approximation of the substituted Eq. (11), as follows:

$$\frac{\mathbb{E}[L(x,t+\Delta t)|L(x,t),R(f),I(x,t)] - L(x,t)}{\Delta t}$$
$$= \frac{R(f)I(x,t)(1+\mu)}{2C(x)} + \frac{\lambda R(f)I(x,t)}{2C(x)}\mathbf{D} \cdot L(x,t). \quad (14)$$

Since the second term on the right-hand side of Eq. (14) corresponds to the diffusion of heat, we can expect that increasing the value of λ improves the performance for storage load balancing. In addition, since the first term on the right-hand side of Eq. (14) corresponds to a heat source, we can expect that increasing the value of μ improves the search performance.

4. Evaluation

In the previous section, we showed analytically that the proposed replication scheme can be expected to achieve network-wide storage load balancing because it is strongly analogous to thermal diffusion. Here, we will evaluate the proposed replication scheme through simulation, with respect to not only storage load balancing but also search performance. The replication ratio used in the evaluation is given by Eq. (6).

In addition, we will evaluate whether the proposed replication scheme has a better ability to find a wider range of trade-off points between storage load balancing and search performance. This is because there are various possible demands for the network performance according to the satisfaction of users and the policies of network operators. For example, certain network operators/users might attach importance to the search performance, while another network operators/users might attach importance to the load-balancing performance. Thus, it is important to show that the proposed replication scheme has a better ability to achieve various possible demands for the network performance.

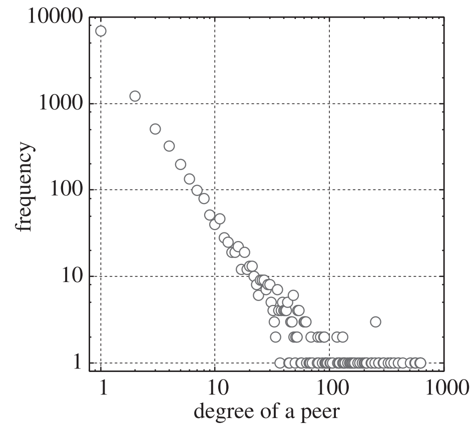


Fig. 2 Distribution of the degree of peers in the simulated network. This distribution follows a power law.

4.1 Simulation Model

The configuration of the P2P network simulation model is described as follows. The total number of peers present in the network is 10,000, and the total number of links between peers is 20,000. The distribution of the degree of peers in the network is shown in Fig. 2. The topology of the P2P network in the simulation model used is generated by the algorithm described in [12]. The topology follows a power law [13]–[15] with respect to the distribution of degree. Since the Gnutella network is a representative unstructured network that follows an approximate power law with respect to the distribution of the degree of peers [16], it is valid to use the network topology following a power law for the investigation of unstructured P2P networks. The present paper considers static network topology.

The maximum storage capacity of every peer is 40. One file consumes one unit of storage, so that the maximum number of files that a peer can hold is 40. When the storage capacity of a peer is full and a request for a replica of a requested file arrives at the peer, the oldest file is replaced by the requested file (i.e., a FIFO replacement method is used).

The query forwarding method used in the present paper is a 16-walker random walk [4]. The walk uses 16 walkers, each with a query, which randomly walk around peers starting at the peer making the query for a requested file. The smallest number of hops among the number of hops that 16 walkers need in order to find the requested file is taken as the number of hops in the current file search. All 16 walkers are allowed up to 100 hops in one file search, and it is possible for a walker to revisit the same peer more than once. In each run of the simulation, the search for a requested file is individually repeated 50,000 times, and the peer that makes the query is chosen randomly. Furthermore, the file requested is randomly determined each time. This means that all types of files in the network have an equal chance of being chosen.

The total number of file types in the network is 100, with 10 files initially allocated over the network for each type of file. This initial distribution of 1,000 files is random, but is determined in the same way each time the simulation model is run.

4.2 Evaluation Criteria

Using random walk-based query forwarding methods, high-degree peers are more likely to be exposed to high load than low-degree peers, so the load imbalance among groups of peers of the same degree should be evaluated. Let $s(d)$ be the average storage utilization ratio of peers of degree d in the network. The evaluation criterion for storage load balancing is defined by the standard deviation of $s(d)$:

$$\sqrt{\sum_{d=1}^{d_{\max}} \frac{(s(d) - \bar{s})^2}{d_{\max}}}, \tag{15}$$

where $d_{\max}$ represents the maximum degree of the network and $\bar{s}$ is the average of $s(d)$, defined as $\sum_{d=1}^{d_{\max}} s(d)/d_{\max}$. Here, we use the standard deviation of $s(d)$ at the moment at which the average storage utilization ratio of all of the peers in the network, called $\bar{H}$, has just exceeded 0.025, 0.05, and 0.1, denoted by s_1, s_2, and s_3, respectively. Here, a smaller standard deviation represents better storage load balancing ability.

The evaluation criterion for search performance is defined as the number of hops needed to find the files requested at the moment at which $\bar{H}$ has just exceeded 0.025, 0.05, and 0.1, denoted by h_1, h_2, and h_3, respectively. Here, a smaller number of hops represents better search performance.

Three pairs of coordinates $\langle s_1, h_1\rangle$, $\langle s_2, h_2\rangle$, and $\langle s_3, h_3\rangle$ are obtained after running the simulation model. The model is run 200 times for each scenario, and the average values of those three points over 200 runs, $\langle \bar{s}_1, \bar{h}_1\rangle$, $\langle \bar{s}_2, \bar{h}_2\rangle$, and $\langle \bar{s}_3, \bar{h}_3\rangle$, are plotted. The plotted graph may show the ability of the selected load balancing scheme in exploring the trade-off between storage load balancing and search performance.

4.3 Results

The proposed replication scheme has two parameters, μ and λ, as shown in Eq. (6). Tuning the values of the two parameters is necessary in order to find the trade-off between storage load balancing and search performance sufficiently. While search performance depends on μ, storage load balancing ability depends on λ. Therefore, we test 12 sets of (μ, λ), being all combinations of $\mu = \{-0.5, 0, 0.5\}$ and $\lambda = \{1, 5, 10, 20\}$.

We use two replication schemes proposed in our preceding work [7] for comparison with the proposed replication scheme. One scheme is Path Random Replication (**PRR**), which makes a replica of a requested file only in peers on the current search path with fixed probability. The fixed probability is denoted by M, which is the same as in Path Adaptive Replication (PAR), which is explained in detail in Sect. 2. The second scheme is PAR. These two schemes have only one parameter, which is $M \in [0, 1]$. As values for M, $0.1, 0.2, \cdots$, and 0.9 are used.

Plotting $\langle \bar{s}_1, \bar{h}_1\rangle$, $\langle \bar{s}_2, \bar{h}_2\rangle$, and $\langle \bar{s}_3, \bar{h}_3\rangle$ for all three schemes used gives the graphs shown in Fig. 3.

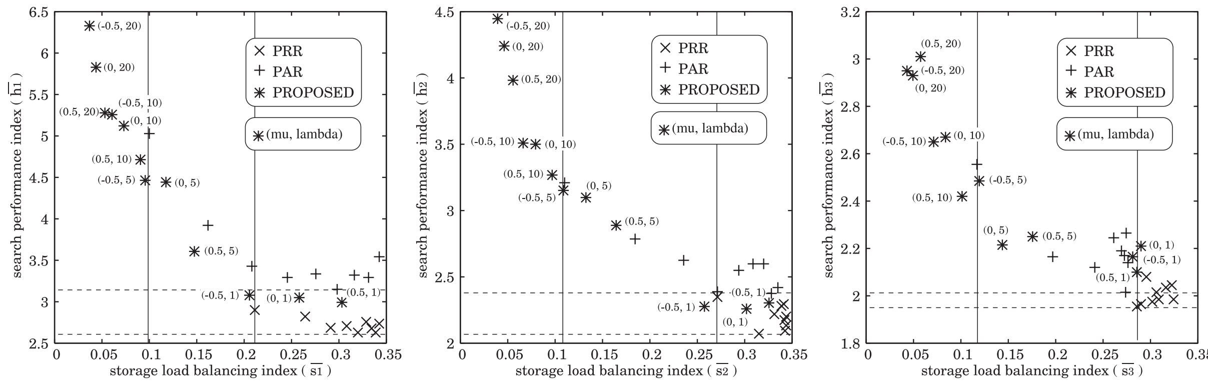
4.4 Discussion

4.4.1 Fundamental Performance

With respect to storage load balancing ability, Fig. 3 shows that, depending on set of values of (μ, λ), the proposed replication scheme gives better performance than PRR or PAR for any parameters and for every observation snapshot, that is, $\bar{H} = 0.025$, $\bar{H} = 0.05$, and $\bar{H} = 0.1$. Therefore, our primary objective, which is to achieve better ability in balancing the storage load among peers in an autonomous and distributed manner, could be achieved. However, the search performance of the proposed scheme is almost the same as or slightly worse than PRR or PAR. These results suggest that the trade-off between storage load balancing and search performance is similar in the three replication schemes used. Therefore, we need to determine how widely the three schemes are able to find this trade-off.

Figure 3 indicates that the proposed scheme shows a wider spread of plotted points (or trade-off points) along both axes (load balancing and search performance) than PRR or PAR. That is, by changing the parameter values, the proposed method can find almost identical trade-off points to PRR and PAR, as well as trade-off points that PRR and PAR cannot find. Therefore, the proposed scheme is the best for exploring a wide range of trade-off points.

The scale of the horizontal axes (i.e., the variation range of storage load balancing performance) of the three graphs in Fig. 3 is the same regardless of the value of $\bar{H}$. The reason for this can be explained as follows. First, the probability that a random walker (i.e., query) stays at a peer is shown to be proportional to the degree of the peer [11]. In addition, the peer that makes the query and the file that is requested by the peer are chosen randomly each time in the simulation. This indicates the frequency of requests for the creation of replicas arrive at peers each time does not depend on $\bar{H}$. Second, the storage load imbalance among peers that has appeared each time is determined by the frequency of requests for the creation of replicas, which does not depend on $\bar{H}$ as shown in the above, and the file replication ratio (i.e., the acceptance ratio of the request for the creation of replica). The proposed file replication scheme attempts to uniform this storage load imbalance among peers by controlling the file replication ratio. The file replication ratio in a peer is uniquely defined by the storage load imbalance among peers, which is defined by the difference between its storage utilization ratio and those of its neighboring peers (i.e., $\mathbf{D} \cdot L(x, t)$), as shown in Eq. (6). The differences in the storage utilization ratio among peers does not depend on $\bar{H}$ by definition unless $\bar{H}$ is close to 0 or 1 as explained in the end of this paragraph. Therefore, the file replication ratio does not depend on $\bar{H}$, nor does the behavior of the proposed file replication scheme. This indicates the storage load imbalance among peers realized by the proposed file



(a) $\bar{H} = 0.025$. (b) $\bar{H} = 0.05$. (c) $\bar{H} = 0.1$.

Fig. 3 Simulation results. The solid lines indicate the smallest standard deviations of storage utilization ratios that PRR and PAR could achieve. The broken lines indicate the smallest numbers of hops that PRR and PAR could achieve.

replication scheme does not depend on $\bar{H}$. Finally, the storage load balancing performance is evaluated based on the standard deviation of $s(d)$ of which the definition does not depend on $\bar{H}$, as shown in Eq. (15). The above three remarks indicate that the storage load balancing performance, which is evaluated based on the standard deviation of $s(d)$, does not depend on $\bar{H}$. However, the storage load balancing performance must depend on $\bar{H}$ when most of the peers experience $L(x, t) \sim 0$ (resp. 1) (i.e., $\bar{H} \sim 0$ (resp. 1)). That is, if $\bar{H} \sim 0$ (resp. 1), then the standard deviation of $s(d)$ becomes close to 0.

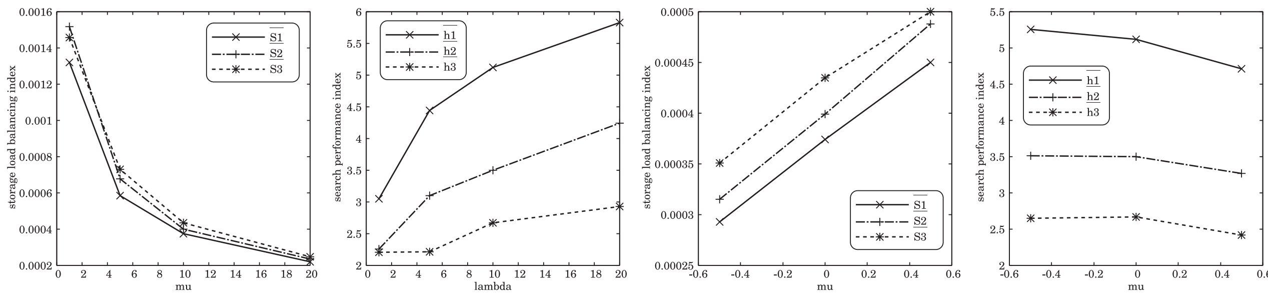
On the other hand, the scale of the vertical axes (i.e., the variation range of the search performance) of the three graphs in Fig. 3 is different. That is, the scale decreases and narrows as the value of $\bar{H}$ increases. This is because the number of hops needed to find the requested files decreases as the value of $\bar{H}$ increases since the search performance improves as the number of files in the network increases.

4.4.2 Guideline on Settings of Parameter Values

The proposed replication scheme has two parameters μ and λ. Therefore, the trade-off point realized between storage load balancing and search performance would vary based on the tuning of the two parameters. In the following, we present a qualitative discussion about the relationship between the values of parameters and the performance of the proposed scheme using the simulation results. Although qualitative guideline of parameter tuning does not offer a solid method that can ensure an exact achievement of a specific trade-off point, it is still useful as a first step to realize a parameter tuning method with trial-and-error manner.

In the simulation experiments, a random walk-based query forwarding method was used. According to an existing theory on random walk in a network [11], the probability with which a walker with a query approaches a node is proportional to the degree of the node. Therefore, we can consider that a request for the creation of a replica of a file arrives (via a walker) more frequently at peers of higher degree. In addition, since the initial load states of all of the peers were approximately the same in the experiments, the peers are considered to have experienced $\mathbf{D} \cdot L(x, t) \sim 0$, at least during the early stage of the file searches. Considering these two features collectively, high-degree peers would mostly experience $\mathbf{D} \cdot L(x, t) \sim 0$ and $\mathbf{D} \cdot L(x, t) < 0$, and low-degree peers would mostly experience $\mathbf{D} \cdot L(x, t) \sim 0$ and $\mathbf{D} \cdot L(x, t) > 0$, at least during the early stage of the file searches.

First, suppose that we fix the value of μ and vary the value of λ. Since the second term in the right-hand side of Eq. (14) corresponds to heat diffusion, increasing the value of λ would improve the storage load balancing performance. At the same time, improving storage load balancing by increasing λ would cause a degradation of search performance, because at high-degree peers (at which search queries frequently arrive), the number of files becomes smaller. Next, suppose that we vary the value of μ and fix the value of λ. Since the first term in the right-hand side of Eq. (14) corresponds to a heat source, increasing the value of μ would improve the search performance. From Fig. 1, we can see that $P(x, t)$ for $\mathbf{D} \cdot L(x, t) \sim 0$ and $\mathbf{D} \cdot L(x, t) < 0$, which higher degree peers mostly experience, becomes larger as μ increases relative to $P(x, t)$ for $\mathbf{D} \cdot L(x, t) \sim 0$ and $\mathbf{D} \cdot L(x, t) > 0$, which lower degree peers mostly experience. That is, we consider that the ratio between a replication ratio of a lower degree peer (P_L) and that of a higher degree peer (P_H), P_H/P_L, increases with the value of μ. The probability of a query arriving at a higher degree peer (Q_H) is higher than that of a query arriving at a lower degree peer (Q_L), i.e., $Q_H > Q_L$. Coupling this observation with the previous observation, we can consider that the ratio between the storage utilization ratio of the lower degree peer ($U_L \propto Q_L \times P_L$) and that of the higher degree peer ($U_H \propto Q_H \times P_H$), U_H/U_L, becomes higher with the value of μ, and consequently the storage load balancing performance becomes worse. In summary, we can hypothesize as follows:



(a) Storage load balancing performance index ($\bar{s}_1$, $\bar{s}_2$, $\bar{s}_3$) versus λ with $\mu = 0$.

(b) Search performance index ($\bar{h}_1$, $\bar{h}_2$, $\bar{h}_3$) versus λ with $\mu = 0$.

(c) Storage load balancing performance index ($\bar{s}_1$, $\bar{s}_2$, $\bar{s}_3$) versus μ with $\lambda = 10$.

(d) Search performance index ($\bar{h}_1$, $\bar{h}_2$, $\bar{h}_3$) versus μ with $\lambda = 10$.

Fig. 4 Relationship between values of μ and λ and performance of proposed scheme.

- With the value of μ fixed, the bigger the value of λ, the better the storage load balancing performance and the worse the search performance.
- With the value of λ fixed, the bigger the value of μ, the worse the storage load balancing performance and the better the search performance.

The simulation results do support the idea that with the value of μ fixed, the load balancing performance improves with the value of λ, while the search performance deteriorates. Typical examples are shown in Figs. 4(a) and (b). On the other hand, the results partially support the idea that with the value of λ fixed, the load balancing performance deteriorates with the value of μ, while the search performance improves. Typical examples are shown in Figs. 4(c) and (d). This hypothesis seems to be satisfied only when $\bar{H} = 0.025$ and $\bar{H} = 0.05$. This could be because the assumption that the peers stay around $\mathbf{D} \cdot L(x, t) \sim 0$ is correct only in these situations.

5. Advanced Evaluation

In the evaluation of Sect. 4, the factors to cause load bias among peers were network topology and a query forwarding method. In this section, we will examine a stronger load bias among peers by using a query forwarding method that differs from the method used in Sect. 4 and examine whether the proposed replication scheme can balance the storage load even in such a case.

The query forwarding method used in this section is the degree proportional-based k-walker random walk. In the degree proportional-based k-walker random walk, the next hop peer of each walker is selected from the neighboring peers by the probability that is proportional to the degree of the peers, while the original k-walker random walk uses a random selection policy for the next hop peer. The probability for selecting peer x_i as the next hop peer of peer x in the degree proportional-based k-walker random walk is given by

$$\frac{d(x_i)}{\sum_{i=1}^{d(x)} d(x_i)}, \tag{16}$$

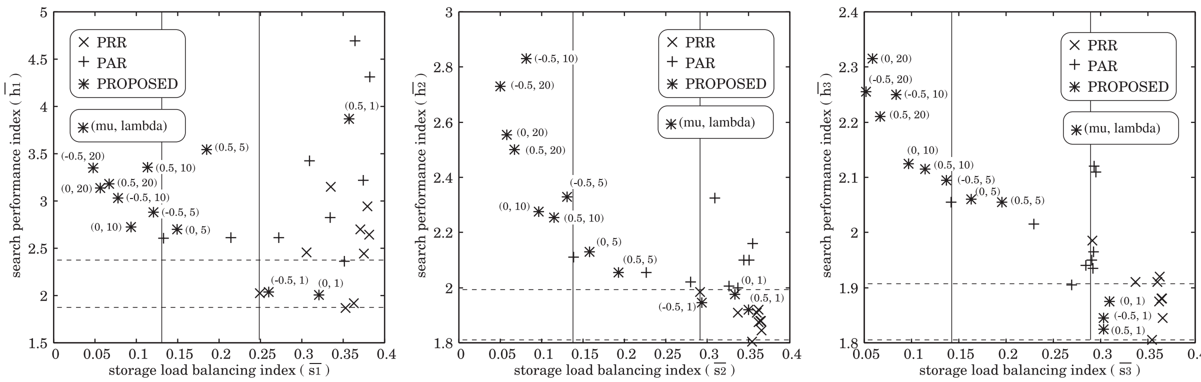
where x_i denotes a peer adjacent to peer x and $d(x)$ denotes the degree of peer x.

We performed the same simulation as that in Sect. 4 using the degree proportional-based k-walker random walk rather than the original k-walker random walk. The simulation results are shown in Fig. 5.

The simulation results in Fig. 5 are similar to those in Fig. 3, so the argument given in the previous sections is mostly valid in this section as well. That is, compared to other replication schemes, even when the degree proportional-based k-walker random walk, which would promote a storage load unbalance, is used, the proposed scheme has a better ability to balance storage load among peers and finds a wider range of trade-off points between storage load balancing and search performance.

However, the simulation results given in this section are different from those of the discussion in the previous section in the case of $\bar{H} = 0.025$. That is, as shown in Fig. 5(a), the search performance is poor when the value of μ is the largest (i.e., $\mu = 0.5$). In the following, we will discuss this difference and show that it can be seen as a kind of transient state before sufficient storage load balancing is achieved.

High-degree peers are more likely to be on the search path in the degree proportional-based k-walker random walk compared to the original k-walker random walk. Therefore, requests for the creation of replicas arrive more frequently at high-degree peers in the degree proportional-based k-walker random walk than the original k-walker random walk. In addition, the replication ratio $P(x, t)$ is large when the value of μ is large. Therefore, when the value of μ is large, replica creation is concentrated on high-degree peers until the value of $\mathbf{D} \cdot L(x, t)$ (i.e., $P(x, t)$) becomes sufficiently close to 0. The concentration is increased particularly when the value of $\bar{H}$ is small, because $\mathbf{D} \cdot L(x, t)$ is not sufficiently close to 0 in the early stage of the simulation. That is, low-degree peers hardly have replicas in the degree proportional-based k-walker random walk when the value of $\bar{H}$ is small. Therefore, in the degree proportional-based k-walker random walk, if the search path happens to be composed mostly of low-degree peers which have few replicas, the number of hops for file search increases greatly, although, in many



(a) $\bar{H} = 0.025$. (b) $\bar{H} = 0.05$. (c) $\bar{H} = 0.1$.

Fig. 5 Simulation results for the degree proportional-based k-walker random walk. The solid lines indicate the smallest standard deviations of storage utilization ratios that PRR and PAR could achieve. The broken lines indicate the smallest numbers of hops that PRR and PAR could achieve.

cases, file search would succeed with a small number of hops because file search performance is highly dependent on high-degree peers. We can justify the above hypothesis by observing the number of hops needed to find the requested file in the early stage of a simulation. That is, we confirmed that the number of hops often becomes 100 hops for the entire period only when $\mu = 0.5$, where 100 hops is the maximum number of hops allowed in one file search. An extremely large number of hops for file search makes the mean number of hops large. Note that the above-mentioned tendency is decreased when the value of λ is large because the bias of the number of created replicas among peers are eliminated in such a case.

The simulation results presented in this section and in the previous section indicate that the relationships between the parameter values and the performance depends on the states: the state in which only a few replicas of files are distributed over the network, that in the state in which replicas are sufficiently distributed, and that in the medium state. The reason why the relationship in the state in which replicas of files are sufficiently distributed over the network differs from those of the other states is that the storage utilization ratios of some peers have reached one, and their states of storage load cannot be distinguished. This situation is not desirable because a storage load balancing mechanism does not work fully. Therefore, for example, using the storage access ratio per unit time rather than the storage utilization ratio would make it possible not only to guess the relationship but to maintain a stable state in which a storage load balancing mechanism works as well.

6. Conclusion

The present paper has introduced a file replication scheme that is analogous to thermal diffusion and is intended to balance the storage load among peers in an autonomous and distributed manner. The results of the theoretical analysis have shown that the statistical behavior of the storage load balancing controlled by the proposed scheme has an analogy with a thermal diffusion phenomenon. In addition, the relationship between the parameter values of the proposed scheme and its ability with respect to load balancing and searching has been discussed. The results of the simulation experiments have shown that, compared to other replication mechanisms, the proposed mechanism is not only better at balancing the storage load among peers, which was the primary objective, but also a wider range of trade-off points between storage load balancing and search performance can be found by adjusting two parameters in the proposed scheme.

Acknowledgments

This work was partly supported by the National Institute of Information and Communications Technology, and the Japan Society for the Promotion of Science, Grant-in-Aid for Scientific Research (S) (No. 18100001).

References

[1] M. Uchida, K. Ohnishi, K. Ichikawa, M. Tsuru, and Y. Oie, “Dynamic storage load balancing with analogy to thermal diffusion for p2p file sharing,” Proc. Workshop on Interdisciplinary Systems Approach in Performance Evaluation and Design of Computer & Communications Systems (Inter-Perf 2006), 6 pages, Pisa, Italy, Oct. 2006.

[2] E.K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, “A survey and comparison of peer-to-peer overlay network schemes,” IEEE Communications Surveys & Tutorials, vol.7, no.2, pp.72–93, Second Quarter 2005.

[3] E. Cohen and S. Shenker, “Replication strategies in unstructured peer-to-peer networks,” Proc. ACM SIGCOMM 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, pp.177–190, Pittsburgh, PA, USA, Aug. 2002.

[4] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, “Search and replication in unstructured peer-to-peer networks,” Proc. 16th international conference on Supercomputing, pp.84–95, New York, USA, June 2002.

[5] A. Rao, K. Lakshminarayanan, S. Surana, R. Karp, and I. Stoica, “Load balancing in structured P2P systems,” Proc. 2nd International

Workshop on Peer-to-Peer Systems (IPTPS'03), LNCS2735, pp.68–79, Berkeley, CA, USA, Feb. 2003.

[6] D.R. Karger and M. Ruhl, "Simple efficient load balancing algorithms for peer-to-peer systems," Proc. sixteenth annual ACM symposium on Parallelism in algorithms and architectures, pp.36–43, Barcelona, Spain, June 2004.

[7] H. Yamamoto, D. Maruta, and Y. Oie, "Replication method for load balancing on distributed storages in P2P networks," IEICE Trans. Inf. & Syst., vol.E89-D, no.1, pp.171–180, Jan. 2006.

[8] G. Cybenko, "Dynamic load balancing for distributed memory multiprocessors," Parallel and Distributed Computing, vol.7, no.2, pp.279–301, Oct. 1989.

[9] J. Boillat, "Load balancing and poisson equation in a graph," Concurrency Practice and Experience, vol.2, no.4, pp.289–313, Dec. 1990.

[10] A. Corradi, L. Leonardi, and F. Zambonelli, "Diffusive load balancing policies for dynamic applications," IEEE Concurrency, vol.7, no.1, pp.22–31, 1999.

[11] R. Motwani and P. Raghavan, Randomized Algorithms, Cambridge University Press, 1995.

[12] T. Bu and D. Towsley, "On distinguishing between internet power law topology generators," Proc. IEEE Infocom 2002, pp.638–647, New York, NY, USA, June 2003.

[13] A.L. Barabasi and R. Albert, "Emergence of scaling in random networks," SCIENCE, vol.286, pp.509–512, Oct. 1999.

[14] R. Albert and A.L. Barabasi, "Topology of evolving networks: Local events and universality," Phys. Rev. Lett., pp.5234–5237, Dec. 2000.

[15] L.A. Adamic, R.M. Lukose, A.R. Puniyani, and B.A. Huberman, "Search in power-law networks," Phys. Rev., vol.E64, 046135, 2001.

[16] M. Ripeanu, A. Iamnitchi, and I. Foster, "Mapping the gnutella network," IEEE Internet Comput., vol.6, no.1, pp.50–57, Jan./Feb. 2002.

Masato Uchida received the B.E., M.E. and D.E. degrees from Hokkaido University, Hokkaido, Japan in 1999, 2001, and 2005, respectively. In 2001, he joined NTT Service Integration Laboratories, Tokyo, Japan. Since August 2005, he has been an Associate Professor in Network Design Research Center, Kyushu Institute of Technology, Kyushu, Japan. His research fields include computer communication networks and information-based learning theory. He is a member of the ACM.

Kei Ohnishi received Bachelor, Master, and Doctor Degrees of Design from Kyushu Institute of Design, Japan in 1998, 2000, and 2003, respectively. He worked as a postdoctoral researcher for University of Illinois, Kyushu Institute of Technology, and Human Media Creation Center/Kyushu. Since October 2007, he has been an associate professor at Kyushu Institute of Technology. His research interests include soft computing techniques and P2P networks. He is a member of the SOFT.

Kento Ichikawa recieved B.Sc., M.Sc. and D.Sc. from University of Tokyo, Japan in 1999, 2001, and 2005, respectively. He has worked as a postdoctoral researcher for Kyushu Institute of Technology since April 2005. His research interests include network science and theoretical physics. He is a member of the Physical Society of Japan.

Masato Tsuru received B.E. and M.E. degrees from Kyoto University, Japan in 1983 and 1985, respectively, and then received his D.E. degree from Kyushu Institute of Technology, Japan in 2002. He worked at Oki Electric Industry Co., Ltd., Nagasaki University, and Telecommunications Advancement Organization of Japan. In 2003, he joined the Department of Computer Science and Electronics, Kyushu Institute of Technology as an Associate Professor, and then has been a Professor in the same department since April 2006. His research interests include performance measurement, modeling, and management of computer communication networks. He is a member of the IEEE, ACM, IPSJ, and JSSST.

Yuji Oie received B.E., M.E. and D.E. degrees from Kyoto University, Kyoto, Japan in 1978, 1980 and 1987, respectively. From 1980 to 1983, he worked at Nippon Denso Company Ltd., Kariya. From 1983 to 1990, he was with the Department of Electrical Engineering, Sasebo College of Technology, Sasebo. From 1990 to 1995, he was an Associate Professor in the Department of Computer Science and Electronics, Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology, Iizuka. From 1995 to 1997, he was a Professor in the Information Technology Center, Nara Institute of Science and Technology. Since April 1997, he has been a Professor in the Department of Computer Science and Electronics, Faculty of Computer Science and Systems Engineering, Kyushu Institute of Technology. His research interests include performance evaluation of computer communication networks, high speed networks, and queueing systems. He is a fellow of the IPSJ and a member of the IEEE.