

## Peer-to-Peer システムにおける共有ファイルの最適配置手法の提案

Dao Quang Trung<sup>†</sup> 片桐 孝洋<sup>†</sup> 本多 弘樹<sup>†</sup> 弓場 敏嗣<sup>†</sup>

<sup>†</sup> 電気通信大学大学院情報システム学研究科

### 概要

本研究はPeer-to-Peer ファイル共有システムにおける共有ファイルの最適配置手法を提案する。本手法ではシステム内の各マシンを論理的に完全網に形成する。各マシンは互いにピングし、ピングで得られたマシン間の通信時間を基にして共有ファイルの最適配置場所を決定する。提案手法を評価するために実際のインターネット上のマシンを使って、2つ実験を行った。予備実験では提案手法を使って、共有ファイル最適配置場所を決定する。共有ファイルのダウンロードを行う検証実験は、予備実験の結果を検証するため行った。

## An Optimal Placement Method for Sharing Files in Peer-to-Peer Systems

Dao Quang Trung<sup>†</sup>, Takahiro Katagiri<sup>†</sup>, Hiroki Honda<sup>†</sup>, and Toshitsugu Yuba<sup>†</sup>

<sup>†</sup> Graduate School of Information Systems, The University of Electro-Communications

### Abstract

This paper proposes an optimal placement method for sharing files in peer-to-peer file-sharing system. The machines in a system are logically formed in a complete graph, and each other carries out *ping* command. The optimal placement of a shared file is decided based on the communication latency. In order to evaluate the proposal technique, two experiments were actually conducted by using real machines on the Internet. In a preliminary experiment, it was tested whether the shared file optimal arrangement place can be decided using the proposed technique. In the verification experiment, the result of a preliminary experiment was performed in order to verify its correctness.

### 1. はじめに

近年、Peer-to-Peer 技術が著しく発展している。2000 年の Wisconsin 大学の調査[1]によると、Peer-to-Peer ファイル共有システムで最もよく知られた Napster において、TCP トラフィック全体に対して TCP と HTTP はほぼ同じトラフィックであった。しかし 2002 年の Washington 大学の調査では、TCP トラフィック全体に対して Peer-to-Peer 共有ファイルアクセスは 43% を占めていたのに対し、ワールドワイドウェブでは 14% しか占めていない[2]。この結果、Peer-to-Peer 共有ファイルアクセスの増加により、インターネットのバンド幅枯渇現象が生じているといえる。Peer-to-Peer 技術を活用し、Peer-to-Peer システム環境に多くの共有ファイルを配置できる。インターネットの通信性能を維持するために、ネットワーク負荷を軽減できる技術が必要とされる。

本論文では、Peer-to-Peer ファイル共有システムにおける共有ファイルの最適配置場所を決める方法を提案する。この方法では、システム内において共有ファイ

ルを配置可能なマシンを論理的に完全網によって表現する。その完全網において、ピングコマンドを実行した結果から、共有ファイルの最適配置場所を決める。またある容量のファイルを、ピングでの実験環境と同じ環境でダウンロードする実験を行う。実験の結果、共有ファイルをダウンロードする時間を最短とする最適配置場所は、提案した方法から得られる最適配置場所と一致した。

### 2. 背景

#### 2.1 Peer-to-Peer ファイル共有システム

Peer-to-Peer ファイル共有システムは、コンテンツデリバリシステムの1つである。Peer-to-Peer の構造としては、以下図1のように Napster のようなインデックス集中型 Peer-to-Peer と Gnutella のようなインデックス分散型 Peer-to-Peer である。

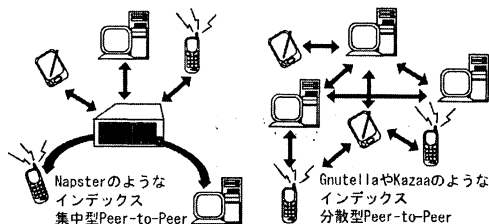


図 1. Peer-to-Peer の構造

## 2.2 最適配置問題

本稿で扱う最適配置とは、ユーザが効率的に資源をダウンロードできるように、配置することである。ここで、「効率的」とは、以下の2つ意味である。

- 1) 共有ファイルのダウンロードに関する最大時間を最小にする。(グラフ理論のK-センタ最小化問題[3]からモデル化される)[4]
- 2) 共有ファイルの平均ダウンロード時間を最小にする。(グラフ理論の施設配置問題[3]とK-メディア最小化問題[4]からモデル化される)[4]

本研究では2つ目の意味と解釈し、共有ファイルの最適配置問題を  $j$  検討する。

## 2.3 制限と従来法との違い

本研究のPeer-to-Peerシステムは次のような制限がある。システム内のマシン数は数10台と限定される。全てマシンのユーザが合意のもとに、共有ファイルを他マシンに設置できる(アップロードできる)とする。

また最適配置問題の観点では、従来のアルゴリズムでは物理トポロジ依存の手法が主流であったが、本案手法は、物理トポロジに依存しない。提案手法では、システム内のマシンが論理的に完全網を構成し、互いにピングコマンドを行って得たデータを基にして、共有ファイル最適配置場所を確定する。

## 3. 最適配置手法

### 3.1 前提モデル

Peer-to-Peerファイル共有システムの最適配置問題を解くにあたって、以下のモデルを考える。

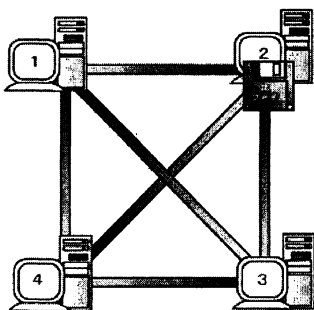


図 2. 完全網の例

- 1) マシンは図2のように完全網を構成する。各マシンは、それらのマシンの中で、どのマシンに共有ファイルが配置されているかを知っているとす。
- 2) 共有ファイルの容量は既知である。
- 3) 共有ファイル転送用プロトコルはTCP/IPプロトコルとする。
- 4) マシンはPingコマンドを提供する。他のマシンへピングパケットを転送できるし、他のマシンからのピングに対し応答をする。

### 3.2 最適配置のための解析

最適配置確定するため以下のステップを行う。

- 1) 各マシンから他の全てのマシンにピングを送信する。
- 2) 送信したピングの結果を別のデータ処理用マシンに送信する。
- 3) データ処理用マシンでデータ解析を行い、最適配置アルゴリズムを適用して、最適配置場所を確定する。

$m$  台のマシンがあるとする(マシン1~マシン  $m$ )。その  $m$  台のマシンの中で共有ファイルの最適配置場所を決める。測定データ  $\alpha_{jk}$  を利用して、データを分析する。ただし、 $\alpha_{jk}$  はマシン  $j$  からマシン  $k$  へピングコマンドを実行したときの平均通信遅延時間である。データを解析するため、以下の複数の行列を使用する。それを以下のように定義する。

#### ある時刻における全体ピング行列 $P_i$

時刻  $i$  におけるピング時間を行列  $P_i$  で表示する。

- 1 時間に1回ピングコマンドを実行するとき1日は24時間なので、 $i$  の値は0から23までとなる。

$$P_i = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \dots & \alpha_{1m} \\ \alpha_{21} & \alpha_{22} & \dots & \alpha_{2m} \\ \dots & \dots & \dots & \dots \\ \alpha_{m1} & \alpha_{m2} & \dots & \alpha_{mm} \end{bmatrix} \quad (1)$$

#### 平均全体ピング行列 $P$

平均全体行列  $P$  は、 $i=0,1,2,\dots,23$  のとき、以下のように計算される。すなわち、行列要素は、測定期間について平均したピング時間を表す。

$$P = \frac{1}{24} \times \sum_{i=0}^{23} P_i \quad (2)$$

#### ある時刻における受信ピング行列 $R(i,j)$

時刻  $i$  におけるマシン  $j$  の受信ピング値  $R(i,j)$  は、他の全てのマシンからマシン  $j$  へピングした結果の平均である。式(3)のように定義する。

$$R(i, j) = \frac{\sum_{l=1}^m \alpha_{jl}}{m} \quad (3)$$

また、時刻*i*における受信ピング行列は、次の行列(4)で表される。

$$R(i) = [R(i,1) \ R(i,2) \ \dots \ R(i,m)] \quad (4)$$

- ある時刻における送信ピング行列  $S(i, j)$   
時刻*i*におけるマシン *j*の送信ピング値  $S(i, j)$ は、他の全てのマシンへマシン *j*からピングした結果の平均である。次式(5)で与えられる。

$$S(i, j) = \frac{\sum_{l=1}^m \alpha_{lj}}{m} \quad (5)$$

時刻による送信ピング行列は、次の行列(6)で表される。

$$S(i) = [S(i,1) \ S(i,2) \ \dots \ S(i,m)] \quad (6)$$

- 評価行列  $E$   
最後に、マシン *j*における評価値  $E(j)$ を、式(7)のように計算する。

$$E(j) = \frac{\sum_{i=0}^{23} (R(i, j) + S(i, j))}{24} \quad (7)$$

ここで評価行列  $E$ は、評価値  $E(j)$ から構成される行列である。

$$E = [E(1) \ E(2) \ \dots \ E(m)] \quad (8)$$

- 平均化行列  $A$   
平均化行列  $A$ は、式(9)のような行列である。

$$A = \frac{1}{m} \times \begin{bmatrix} 1 & 1 & \dots & 1 \end{bmatrix} \quad (9)$$

### 3.3 共有ファイル最適配置アルゴリズム

共有ファイルの最適配置場所を決定するアルゴリズムを以下に示す。

- 1) ある時刻における全体ピング行列  $P$ を式(1)のように生成する。
- 2) 受信ピング行列  $R(i)$ は式(3)と式(4)を用いて、次式(10)で計算される。

$$R(i) = (P_i A^T)^T = AP_i^T \quad (10)$$

- 3) 送信ピング行列は式(5)と式(6)を用いて、次式(11)で計算される。

$$S(i) = AP_i \quad (11)$$

- 4) 評価行列  $E$ を計算する。

$$E = \frac{1}{24} \times \sum_{i=0}^{23} (R^i + S^i) = \frac{1}{24} \times A \times \sum_{i=0}^{23} (P_i + P_i^T) \quad (12)$$

- 5) 評価行列  $E$ の中で最小要素を探す。その値に対応するマシンが共有ファイルの最適配置場所である。

## 4. 予備実験

### 4.1 目的と内容

予備実験の目的は、ピングの測定結果を用いて、3章で提案した方法を用いて最適配置を行うことである。この予備実験で得られた結果の正当性の検証は、次の検証実験で行う。

ここでは予備実験のため、Peer-to-Peer ファイル共有システムを設定する。4台のマシンを用いて完全網を作る(マシン1~マシン4)。そのほか、マシン5は、データ解析用マシンとして使用する。実験に用いた4台のマシンを表2に示す。

実験対象とするピングで送信したデータサイズとして5種類 100bytes、500bytes、1000bytes、5000bytes、10000bytesを用いた。

### 4.2 実験手順

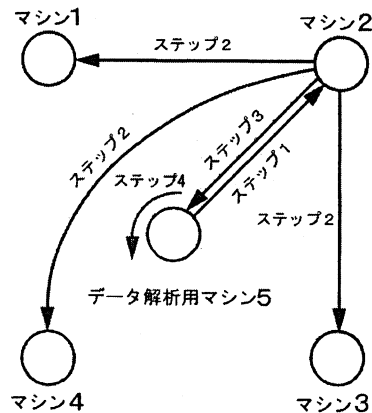


図 3. 予備実験の流れ

実験の流れは、以下の4つのステップから構成される。これらのステップと3章最適配置方法の3ステップの対応は表2の通りである。

表 1. 予備実験の流れ

予備実験の流れ	3章の最適配置提案手法
ステップ1:データ解析用マシン5から他のマシンにピング要求を送る。	マシンから他のマシンにピングする。
ステップ2:マシン1-4はマシン5からピング要求がきたときピングを他のマシンに送る。	
ステップ3:データ解析用マシン5にピングの結果を送る。	ピングした結果をデータ解析用別のマシンに送信する。
ステップ4:マシン5でデータ分析を行う。	データ解析用マシンでデータ分析を行い、最適配置場所を決定する。

### 4.3 実験設定

実験を行うため、以下のような設定を準備した。

- 1) データ解析用マシンから 1 時間ごとにプログラムを自動的に実行し、結果を収集する。プログラムを自動的に実行するため、データ解析用マシンに cron を設定した。
- 2) 実行するプログラムは、他のマシンにアクセスするため、各マシンに自動的にアクセスできるよう ssh を設定した。
- 3) 収集データを置く場所を確定した。

### 4.4 実行と結果

実験は 2003 年 9 月 17 日 5:00~2003 年 9 月 30 日 12:00 の期間で行い、毎時間、ピングを行った(1つピングコマンド実行5回パケットを投げる)。実験期間中、データ解析用マシンの故障で実験用プログラムが起動できなかった箇所が何時間もある。

実験結果を表 2 に示す。表中の値は、評価行列 E の要素の値である。本手法では、表 2 の結果から、評価値が最小となる要素に対応するマシンが共有ファイルの最適配置場所であると判断する。

表 2. 予備実験の結果(ms)

データサイズ	評価値 E(1)	評価値 E(2)	評価値 E(3)	評価値 E(4)	最適配置場所
100B	9.8	13.8	13.1	12.6	マシン1
500B	13.4	15.5	18.8	18.2	マシン1
1000B	18.2	19.9	26.2	26.4	マシン1
5000B	45.5	48.5	72.6	74.9	マシン1
10000B	77.4	78.7	127.3	135.7	マシン1

データサイズを増加することにより、評価値の差が増大する。従って、最適配置場所がより明確に決定できるようになる。

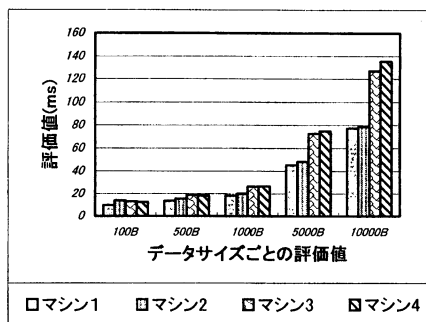


図 4. 評価値の比較

図 5 に示すように本ピングデータは、他論文[参考文献]のピングデータのように周期的な変化はしていない[5]。原因はネットワーク経路が比較的短い(3 マシンは東京都内、1 マシンは大阪にあるが光ファイバで東京と接続している)からであると考えられる。通信時間 0 のときはデータ解析用マシン 5 が故障していることを示す。

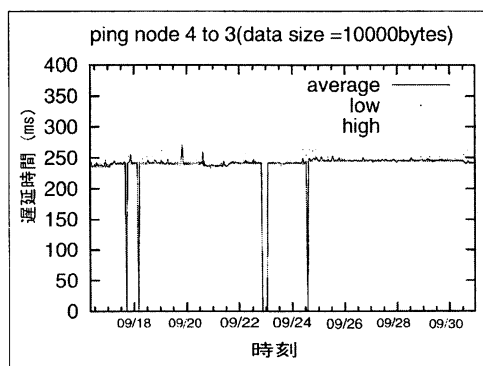


図 5. マシン4からマシン3へピングした結果のグラフ

## 5. 検証実験

### 5.1 目的と内容

検証実験は、本論文で提案する共有ファイルの最適配置アルゴリズムが正しく機能するかを検証することを目的とする。予備実験で用いた同じ実験環境において、各マシンにおいて、決められたファイルをダウンロードする時間を測定する。この実験を行うためのファイルサイズは、以下 4 種類がある: 0bytes、10bytes、30bytes、50bytes。

## 5.2 実験の手順

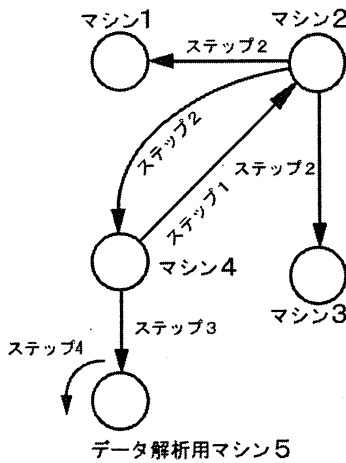


図 6. 検証実験の流れ

実験の流れは、以下の4つのステップから構成される。これらのステップと3章最適配置方法の3ステップの対応は、表3の通りである。またそれらのステップを図6で示す。

表 3. 検証実験の流れ

検証実験の流れ	3章の最適配置提案手法
ステップ1: マシン1から他のマシンにアップロード要求を送る。	マシンから他のマシンにピングする。
ステップ2: 要求されたマシンは残りマシンへファイルをアップロードし、アップロード時間を測定する。	
ステップ3: 測定したデータをマシン1とデータ解析用マシン(マシン5)に送る。	ピングした結果をデータ解析用別のマシンに送信する。
ステップ4: マシン5でデータ解析を行う。	データ解析用マシンでデータ分析を行い、最適配置場所を決定する。

## 5.3 実験の設定

実験を行うため、以下の設定を準備した。

- 4種類の0bytes、10Mbytes、30Mbytes、50Mbytesファイルを作成する。
- 実行プログラムが自動的に実行できるように cron を設定する。

- 実行プログラムが他のマシンにアクセスできるように、ssh と特別なスクリプトプログラムをそれぞれマシンに設定する。

## 5.4 実行と結果

実験は2003年11月10日15:00~2003年11月11日23:00まで、計32時間行われた。2時間ごとに、scp コマンドでアップロードを行った。また「time scp ...」コマンドでアップロード時間を測定した。

実験の結果を表4に示す。表4のように、平均アクセス時間が最小となるマシンは、予備実験で行った実験の結果と一致し、それはマシン1であった。

表 4. 検証実験の結果(ms)

ファイルサイズ	マシン1	マシン2	マシン3	マシン4	最適配置場所
0MB	4.28	5.84	5.17	5.01	マシン1
10MB	18.12	20.46	131.08	129.1	マシン1
30MB	45.88	49.43	360.2	384.4	マシン1
50MB	73.51	79.12	581.63	623.4	マシン1

本検証実験での考察を以下にまとめる。

ファイルサイズが大きくなると、平均ダウンロード時間が大きくなる。またファイルサイズが大きくなると、そのファイルサイズを共有ファイルサイズとするような最適配置場所はより明確になる。図7を見ると、50MB ファイルをダウンロードするときの最適配置場所は、明らかにマシン1かマシン2である。マシン3やマシン4からの平均ダウンロード時間は、マシン1やマシン2と比べて明らかに大きい。しかしその差は、ファイルサイズ0MB や10MB のファイルでは、明確にならない。

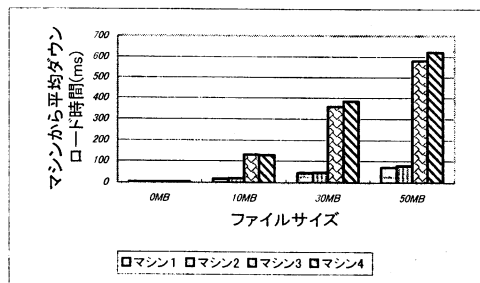


図 7. ファイルの平均ダウンロード時間

## 6. 問題点

### 6.1 ADSL 問題点

検証実験結果から、得られたダウンロード時間はア

ップロード時間より小さい。原因はADSL用インターネットでの通信特性と考えられる。

ADSL用ネットワークでは、ダウンロード時間とアップロード時間が違う[6]。しかし、現在の最適配置問題を解決する手法では、2マシン間の両方向通信時間が同一としている。今後の課題として、ダウンロードとアップロードの時間も考慮して、最適配置を行う手法を開発する必要がある。

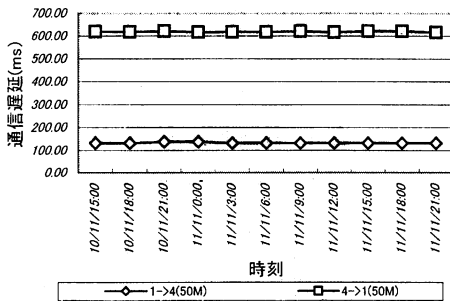


図 8. マシン4のアップロードダウンロード時間

## 6.2 平均計算の問題点

現在ピングで得られた結果を平均にして、最適配置場所を確定している。

ピング用データサイズが増加するにつれて、ピング結果の通信遅延頻度分布の特性が変わる。図9で示すの様に、グラフ形状は1つ山から、2つ山や3つ山グラフに変わっていく。こうした特性を

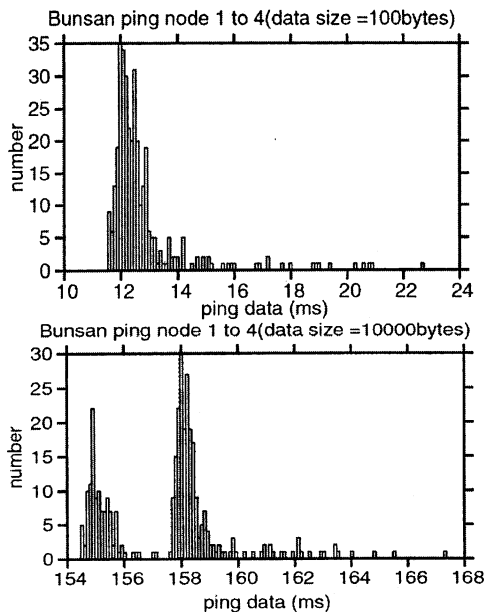


図 9. 通信遅延頻度分布の特性

今後の課題はPingデータを平均にするだけでなく、よく出てくる遅延時間の頻度分布を考慮して、提案手法の数式を改良する必要がある。

## 7. まとめ

本研究ではPeer-to-Peerファイル共有システムにおける最適配置手法を提案した。この手法では、論理的な完全網を作り、ピングデータを基にして、共有ファイル最適配置を決定する。

提案手法では、自動的に共有ファイル最適配置を確定する。実際sshやcron設定により最適配置確定するための手順を自動的に行うことができた。

提案手法を用いて、実際インターネット環境で実装と検証を行った結果、以下の結果が得られた。

1. 予備実験ではピングで使用した5種類のデータサイズで行って、同じ共有ファイルの最適配置場所が得られた。
2. 実際ファイルをダウンロードするときの実際の最適配置場所は、予測した最適配置場所と一致した。

## 参考文献

- [1] D. Plonka, University of Wisconsin-Madison, Napster traffic measurement, March 2000. Available at <http://net.doit.wisc.edu/data/Napster>, March 2000.
- [2] S. Saroiu, K. P. Gummadi, R.J. Dunn, S. D. Gribble, H. M. Levy, An analysis of internet content delivery systems, In Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI 2002), Boston, MA, December 2002.
- [3] <http://www.nada.kth.se/~viggo/problemlist/>
- [4] Lili Qiu, Venkata N. Padmanabhan, Geoffrey M. Voelker, On the placement of web server replicas, Proc. of INFOCOM 2001, Anchorage, USA, April 2001.
- [5] Christopher J. Freitas, Derrick B. Coffin, Richard L. Murphy, The characterization of a wide area network computation, Parallel Computing Vol. 29, pp. 879-894, Jul. 2003.
- [6] S. Saroiu, P. K. Gummadi, and S. D. Gribble, A measurement study of peer-to-peer file sharing system, In Proceeding of Multimedia Computing and Networking, Jan. 2002.