

過去の検索経路情報を利用する P2P ネットワークのための複製配置手法

大西 圭[†] 山本 寛[†] 尾家 祐二[†]

†九州工業大学 電子情報工学科
〒820-8502 福岡県飯塚市川津 680-4

E-mail: †k_ohnisi@pluto.ai.kyutech.ac.jp, ††yamamoto@infonet.cse.kyutech.ac.jp, ††oie@cse.kyutech.ac.jp

あらまし 構造化されない Peer-to-peer (P2P) ネットワーク上でのファイル共有を想定し、P2P ネットワークを構成する機能的に対等なピアはそれらに生じるストレージ負荷に関しても対等であるべきであるとの立場から、新たなストレージ負荷分散のための複製配置手法を提案する。提案手法は、各ピアが保持する過去のファイル検索経路情報を用いて、現ファイル検索経路上のピアおよびそれらの隣接ピアの中からピアを選択し、一定の確率で選択ピアにファイルの複製を生成する。実験の結果は、提案手法が、ファイル検索経路上のピアのみに一定の確率でファイルの複製を配置する比較対象手法に比べ、優れたストレージ負荷分散性能を持つことを示す。

キーワード Peer-to-Peer (P2P), 複製配置手法, power-law ネットワーク, ストレージ負荷分散

A Replication Method for P2P Networks Using Information on the Past Search Paths

Kei OHNISHI[†], Hiroshi YAMAMOTO[†], and Yuji OIE[†]

† Department of Computer Science and Electronics, Kyushu Institute of Technology
Kawazu 680-4, Iizuka-city, Fukuoka, 820-8502 Japan

E-mail: †k_ohnisi@pluto.ai.kyutech.ac.jp, ††yamamoto@infonet.cse.kyutech.ac.jp, ††oie@cse.kyutech.ac.jp

Abstract This paper presents a replication method that is meant to balance storage load of peers in unstructured peer-to-peer (P2P) networks for file sharing, taking a position that peers with equal functions should be equal in load. The presented method first selects peers in which a replica of a requested file is made from among peers on the present search path and their neighboring peers using information on the past search paths that each peer holds, and then makes the replica of the requested file in the selected peers with fixed probability. The experimental results show that the presented method is superior in load balancing to a compared method that makes a replica of a requested file only in peers on the present search path with fixed probability.

Key words Peer-to-Peer (P2P), replication method, power-law network, storage load-balancing

1. はじめに

ファイル共有を目的とした構造化されない Peer-to-Peer (P2P) ネットワークをより実用的にするために、2つの観点から研究が行われている。1つは、ユーザが要求するデータを素早くかつ確実に探し出す検索手法に関する研究である。上記の目標を達成するために行われてきた研究の多くは、この検索手法に関するものである [9][6][3]。もう1つは、ピアのストレージ負荷を分散させる方法論に関する研究である。ストレージ負荷分散研究のほとんどは、Distributed Hash Table (DHT) によりファイル管理される構造化された P2P ネットワークに対して行われている [11][8][7][14]。一方、我々はこれまで、ファイル

共有を目的とした構造化されない P2P ネットワークにおいて、サービス授受能力に関して対等なピアを、サービス授受において生じる負荷に関しても対等にすべきであるという立場に立ち、構造化されない P2P ネットワーク上のストレージ負荷分散に関する研究を行ってきた [13]。具体的には、検索対象となり得るファイルの複製をネットワーク上の複数のピアに生成する部分を制御することで、ファイル検索の速さと確実性を増すと同時に、ファイルのピアからの読み込みおよびファイルのピアへの書き込みの際に生じる負荷を分散させる手法の提案を行ってきた。本論文も、我々の先行研究と同様の立場をとる。

本論文は、ファイル共有を目的とした構造化されない P2P ネットワークにおいて、ストレージ負荷分散を達成するための

新たな複製配置手法を提案する。提案手法は、過去のファイル検索経路情報を用いてファイル検索経路上のピアおよびそれらの隣接ピアの中からファイル複製配置ピアを選択し、一定の確率で選択ピアにファイルの複製を生成する。我々が以前に提案した Path Random Replication [13] は、ファイル検索経路上のピアのみで一定の確率でファイルの複製を生成するものであったが、提案複製配置手法は、その方法と、ファイルの生成に関して、ファイルの複製の生成場所だけが異なる。

2. 検索履歴を用いる複製配置手法

本節では、構造化されない P2P ネットワーク上でのファイル共有を想定し、ファイル検索に要するホップ数を抑え、同時に特定のピアにストレージ負荷が偏らないようにすることを目標に、新たな複製配置手法を提案する。

手法提案に先立つ我々の動機は次のようである。P2P ネットワーク上のピアは機能の観点からは対等であるが、それらの中で degree(他のピアとの接続数) に関して非一様性があるために、ランダムウォークに基づく検索手法の使用の下で、それらの中にファイル検索経路上に位置する回数に関して非一様性が生じる。具体的には、任意のネットワーク上でランダムウォークを行ったとき、各ノードへの到達確率は、定常状態でその degree に比例することが示されている [10]。このとき、ファイル検索経路上のピアのみをファイルの複製が生成される対象として選択し、それらに複製を生成し続ければ、ファイル検索経路上に位置する回数の多い高 degree ピアにファイルが蓄積される速度が他の低 degree ピアのそれよりも大きくなる。また、ファイル検索に関しても、低 degree のピアから発生するファイル検索要求が、より多くのピアとの接続を持つ高 degree ピアを経ることによって検索要求ファイルを持つピアに高い確率で到達可能になることを考えると、低 degree のピアは高 degree ピアに強く依存していると言える。したがって、高 degree のピアに接続する低 degree のピアが、その高 degree のピアに生じる負荷を代わり負うことで、ファイル検索に関する負荷についての低 degree ピアと高 degree ピアの間不公平性解消に貢献することは、妥当な考え方だと思われる。

ここから提案複製配置手法を説明していく。提案手法の基本的な特徴は、ネットワークが過去に経験したファイル検索の情報(検索履歴)を利用して、ファイルの複製を生成するピアを決定することである。ここで、検索履歴は、特定のピアにより集中して管理されるのではなく、各ピアによって管理される。検索履歴を特定のピアが管理してしまうと、サーバクライアントモデルの場合と同様に、情報を管理するピアが故障などでネットワークから消失することで、ネットワーク全体機能が損なわれる可能性がある。一方、各ピアが自身の過去に経験したことからだけを管理することで、そのピアのネットワークからの離脱がネットワーク全体に大きな影響を与えないと考えられる。提案手法は、以下の 3 つの手続きの繰り返しからなる。

(1) ファイル検索経路を得る。

まず、あるピアから発生したファイル検索要求に対して、与えられたファイル検索手法により、要求ファイルを持つピアを

発見する。そして、ファイル検索要求を発生したピアからそのファイルを持つピアまでの経路を、ファイル検索経路として得る。

(2) 検索要求ファイルの複製を生成するピアを選択し、一定の確率でファイルの複製を生成する。

検索要求ファイルの複製を生成するピアの選択と確率的なファイルの複製生成は、ファイル検索経路上の各ピアとその隣接ピアを対象に行われる。いま、ファイル検索経路上のピア q に、通し番号が付けられた Q 個のピアが接続しているとす。まず最初に、ピア q の Q 個の隣接ピアが、過去にファイル検索経路上に位置した回数 O_i ($i = 1, 2, \dots, Q$) の平均

$$\bar{O}_q = \frac{\sum_{i=1}^Q O_i}{Q}$$

を計算する。次に、ピア q が過去にファイル検索経路上に位置した回数 O_q と \bar{O}_q に関して以下の場合分けを行い、検索要求ファイルの複製を生成するピアを選択し、一定の確率でファイルの複製を生成する。

(a) $O_q \leq \bar{O}_q$ のとき

検索要求ファイルの複製が生成されるピアは、ファイル検索経路上に位置するピア q となる。ピア q には、一定の確率でファイルの複製が生成される。この確率は、提案複製配置手法の持つパラメータであり、以降では複製確率と呼ばれる。

(b) $O_q > \bar{O}_q$ のとき

ピア q は、自分を除く自分への接続ピアの中から 1 つだけ検索要求ファイルの複製が生成されるピアを選択する。このとき、次の手続き (3) のところで説明されるポイントというものが用いられる。ポイント数は、0 以上の整数値である。いま、過去にピア q が隣接する Q 個のピアのそれぞれに与えたポイント数を P_i ($i = 1, 2, \dots, Q$) とする。このとき、 i ($i = 1, 2, \dots, Q$) 番目の隣接ピアが、検索要求ファイルの複製が生成されるピアとして選択される確率 S_i は、以下の式で表される。

$$S_i = \frac{\frac{1}{1+P_i}}{\sum_{i=1}^Q \frac{1}{1+P_i}}$$

この式が意味することは、過去にピア q からより多くのポイントを与えられたピアは、検索要求ファイルの複製が生成されるピアとして選ばれることからより高い確率で選ばれることができることである。そして、選択されたピアには、一定の確率でファイルの複製が生成される。この確率は、提案複製配置手法の持つパラメータであり、以降では複製確率と呼ばれる。

以上の手続きにおいて、もし、検索要求ファイルの複製を生成する対象に選ばれたピアが、その検索要求ファイルを既に持っていたとき、その複製は選択ピアに生成されない。

(3) ピアのポイントおよびファイル検索経路上に位置した回数を更新する。

ファイル検索経路上で検索要求ファイルを持つピアがある方向を検索経路上流と呼ぶとき、検索経路上のピアは、1 つ検索経路上流のピアにファイル検索要求を伝播した回数を、1 だけ増加させる。この各ピアが隣接ピアへ検索要求を伝播した回数を、以下ではポイントと呼び、以下では、1 ポイントをピアに与える、という言い方をすることがある。ポイントの記録に関

しては、ポイントを与えたピアがどのピアにポイントを与えたかを記録する。よって、手続き (2) において、あるピアの隣接ピアからファイル複製配置ピアを選択するとき、そのピアが保持するポイント情報のみからピアを選択できる。ポイント渡しが終わった後、ファイル検索要求を発したピアおよび検索要求ファイルを持っていたピアを含むファイル検索経路上の全てのピアは、過去にファイル検索経路上に位置した回数を 1 だけ増やす。最初のファイル検索が始まるまでは、全てのピアが記録・管理するポイント数および過去にファイル検索経路上に位置した回数は、0 である。

以上が我々の第一の提案である。ここで、上記提案手法の手続き (2) において、隣接ピアに過去に与えたポイントを用いてファイルの複製を配置するピアを決定するところを、代わりに隣接ピアの degree を用いて決定する手法も考える。手続き (2) において、各ピアが過去に検索経路上に位置した回数を用いるところにも、ピアの degree を用いることができるが、ピアが検索経路上に位置する回数から degree とほぼ等価な情報を得られると考えられるため、そのような手法は考えない。さらに、上記手続き (2) の部分で、過去に検索経路上に位置した回数に関係なく、全てのピアに隣接ピアからファイル複製配置ピアを選択することを許す手法も考える。よって、提案手法としては、以下の 4 つがある。

提案手法 1 (a): 手続き (2) でポイント使用, 手続き (2) で過去に検索経路上に位置した回数使用 (上記の詳述手法)。

提案手法 1 (b): 手続き (2) でポイント使用, 手続き (2) で全てのピアに隣接ピアからのファイル複製配置ピア選択を許可。

提案手法 2 (a): 手続き (2) でポイントの代わりに degree 使用, 手続き (2) で過去に検索経路上に位置した回数使用。

提案手法 2 (b): 手続き (2) でポイントの代わりに degree 使用, 手続き (2) で全てのピアに隣接ピアからのファイル複製配置ピア選択を許可。

3. 実験による評価

3.1 シミュレーションモデル

P2P ネットワーク・シミュレーションモデルの設定は、次のようである。ネットワーク上に存在するピアの総数を 10000 とし、ピア間のリンクの総数を 20000 とする。使用するシミュレーションモデルにおけるネットワーク・トポロジは、文献 [5] に示された手順に従い決定されている。このネットワーク・トポロジの degree の頻度分布は、べき法則 (power-law) [4] [2] [1] に従う (図 1)。例えば、代表的な構造化されない P2P ネットワークの一つである Gnutella ネットワークの degree の頻度分布は、おおよそべき法則に従うことが報告されており [12], degree の頻度分布がべき法則に従うネットワークを構造化されない P2P ネットワーク研究に用いることは妥当だと考えられる。しかしながら、本論文では、動的なネットワーク・トポロジの変化は考慮していない。

各ピアの最大ストレージ容量を 20 とする。各ファイルを保持するために必要なストレージ容量を 1 とする。よって、各

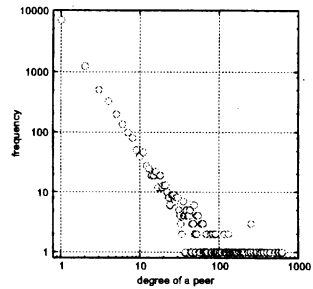


図 1 使用するネットワーク上のピアの degree についての頻度分布。この頻度分布は、べき法則に従っている。

ピアが保持できる最大ファイル数は 20 である。ピアのストレージ容量が限界のときにファイルの複製生成要求が発生した場合、最も配置時刻の古いファイルをそのファイルで置き換える。

ファイル検索手法としては、16-walker random walk [9] を用いる。これは、ファイル検索要求を発したピアから、検索要求を知る 16 のウォーカーが、同期して次に移動するピアをランダムに選択していく検索手法である。1 回の P2P シミュレーションモデルの実行において、計 5 万回のファイル検索が行われる。ファイル検索要求を発するピアおよび要求ファイルの種類はランダムに選ばれる。

3.2 評価方法

本論文で用いるストレージ負荷分散度合いの指標は、横軸にピアの degree、縦軸をピアからのファイル読み回数とピアへのファイル書き込み回数の和、とするグラフ上に、各ピアの読み回数と書き込み回数の和をプロットしたとき、それらの点に対して最小二乗法によって当てはめられた直線の傾きの値とする。このストレージ負荷分散度合いを表す直線の傾きの値を、以降では単純にストレージ負荷と呼ぶ。同様な方法で読み・書き込み負荷分散度合いは与えられ、以降ではそれらを単純に読み負荷、書き込み負荷と呼ぶ。以上のストレージ負荷、読み負荷、書き込み負荷は、それらの値が小さいほど負荷分散が達成されていることを表す。つまり、高 degree のピアと低 degree のピアの負荷の差が小さいことを表す。

平均ホップ数算出の方法は、ネットワークに存在するファイルの種類の時間的変化および平均ホップ数の算出に実際用いる検索の選択の仕方に関係する。ネットワークに存在するファイルの種類は、計 110 種類ある。しかし、一度もファイル検索が行われていない時点では、ネットワークに存在するファイルは 100 種類である。各種類のファイルは、10 個ずつネットワーク上に配置される。この初期配置自体はランダムに決定されるものであるが、実際に用いる初期配置はどのシミュレーション実行においても同じとする。次に、1 万回目のファイル検索が終了した直後に、新たな 10 種類のファイルがネットワーク上に追加配置される。各種類のファイルは、10 個ずつネットワーク上に配置される。このときも初期に配置されるファイルと同様に、配置自体はランダムに決定されるものであるが、実際に用いる配置はどのシミュレーション実行においても同じとする。

平均ホップ数, つまり決められた期間の全てのファイル検索に要したホップ数の平均値は, ネットワーク上に最初から存在する種類のファイルと, 途中から追加配置される種類のファイルのそれぞれに対して算出される。まず, ネットワーク上に最初から存在する 100 種類のファイルの検索に要する平均ホップ数に関しては, 10001 回目から 30000 回目のファイル検索に要したホップ数を対象に計算される。次に, 途中から追加配置される 10 種類のファイルに関しては, 20001 回目から 40000 回目のファイル検索に要したホップ数を対象に計算される。

3.3 実験結果

前節までに説明した P2P シミュレーションモデルおよび評価指標を用いて, 提案複製配置手法のストレージ負荷分散性能, およびそれらを用いた場合のファイル検索に要する平均ホップ数を実験的に調べる。ここで, 比較対象として, 提案複製配置手法とはファイルの複製配置場所だけが異なる Path Random Replication [13] (以下では, PRR と呼ぶ) を用いる。PRR は, ファイル検索手法が得たファイル検索経路上に, 一定の確率 (複製確率) で検索要求ファイルの複製を生成するものである。また, もう一つの比較対象手法として, ファイル検索経路上のピアに, ピアの degree に反比例した確率, $p = C/D$ (C :定数, D :ピアの degree, $C/D > 1$ のとき $p = 1$), でファイルの複製を生成する手法 (以下では, RPID: Replication with Probability Inversely proportional to Degree と呼ぶ) を用いる。RPID は, ピアへ検索要求が到達する確率がピアの degree に比例するとき, その確率と複製確率の積がどのピアにおいても一定となることを意図している。

複製確率は, 提案複製配置手法と PRR とともに, 100%, 60%, 20% の計 3 種類を用いる。また, RPID に関しては, C の値として, 1, 10, 20 の 3 種類を用いる。RPID の C の値を, 今回用いるネットワーク上のピアの最大 degree である 625 以上にした場合, RPID は, 複製確率 100% の PRR と同じになる。1 種類の複製確率あるいは C の値に対して, ネットワーク・トポロジが一定のシミュレーションモデルを, 異なる乱数列を用いて計 20 回実行し, 結果はそれらの平均値として示される。観測されるデータは, 前節に示したストレージ負荷と平均ホップ数に加え, ピアへの総書き込み回数, ネットワーク上の総ファイル数, ファイルの書き込み冗長度, ネットワークのファイル容量の使用度, の 4 つがある。ピアへの総書き込み数は, 検索終了までにネットワーク全体で発生したファイルの書き込み回数の合計である。ネットワーク上の総ファイル数とは, 検索終了時にネットワーク全体が保持するファイルの総数である。この数は, 総書き込み数と等しくなることはない。その理由は, 総ファイル数に初期・追加配置ファイル (計 1100 個) が含まれているからである。また, ファイルの書き込み冗長度とは, 総書き込み数と総ファイル数をそれぞれ T_w , T_f としたとき, $1 - (T_f - 1100)/T_w$ で表される数値であり, 古いファイルへの上書きが起こった度合いを知ることができる。最後に, ファイル容量使用度とは, ネットワーク上の 1 万個のピア全体で保持できるファイル数 2×10^5 に対して, ファイル検索終了時にネットワーク全体でその何割を使用しているかを示す値である。以上の観測データ

を, 表 1 に示す。また, ストレージ負荷分散度合いを表す直線の一例として, 複製確率が 100% の 4 つの提案手法と PRR, および $C = 20$ とした RPID を用いた場合の結果を図 2 に示す。

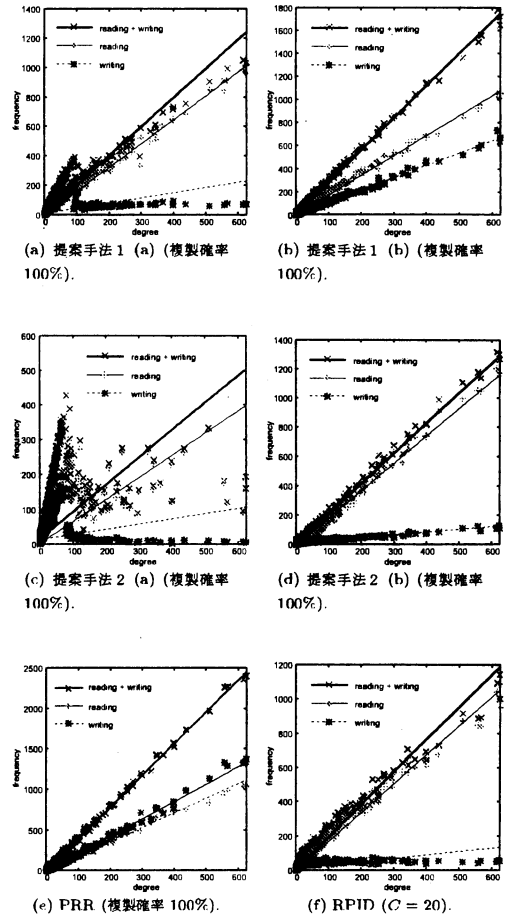


図 2 ストレージ負荷を与える回帰直線の例。

3.4 結果の考察

まず, 第一の提案であった提案手法 1 (a) と PRR の比較を行う。表 1 によると, 提案手法 1 (a) が示すストレージ負荷は, どの複製確率を用いた場合でも, PRR よりも優れていることが分かる。このとき, 初期配置ファイルと追加配置ファイルに対する平均ホップ数は, 両手法間で差がないことが分かる。提案手法 1 (a) と PRR の間のストレージ負荷の違いは, 主に書き込み負荷に関するものであることが分かる。図 2(a)(e) によると, PRR を用いた場合, 読み込み・書き込み回数ともにピアの degree の増加とともに線形増加しているが, 提案手法 1 (a) を用いた場合, degree が約 100 以下のピアの書き込み回数は degree に関して線形増加し, degree が約 100 より大きいピアの書き込み回数はおおよそ同じであることが分かる。この書き込み回数の不連続

表 1 実験の結果. 個々の結果は, 20 回のシミュレーション実行結果の平均である.

	複製 確率	全負荷	読負荷	書負荷	初期データ ホップ	追加データ ホップ	総書込 回数	総フア イル数	書込 冗長度	容量 使用度
提案手法 1 (a)	100%	2.008805	1.654479	0.354325	2.186222	2.145899	116665	91812	0.222	0.459
	60%	1.922022	1.695768	0.226254	2.232426	2.191225	92994	81494	0.135	0.407
	20%	1.789390	1.692647	0.096743	2.510412	2.556894	68578	67526	0.031	0.337
提案手法 1 (b)	100%	2.818721	1.736181	1.082540	2.256043	2.232286	116728	96418	0.183	0.482
	60%	2.543999	1.869296	0.674703	2.323406	2.283876	92216	81863	0.124	0.409
	20%	2.431218	2.175614	0.255604	2.515585	2.459460	66267	64496	0.043	0.322
提案手法 2 (a)	100%	0.846188	0.693888	0.152300	2.533979	2.536525	136983	114721	0.170	0.573
	60%	0.777402	0.678673	0.098729	2.659988	2.692902	107973	98061	0.101	0.490
	20%	0.745461	0.702713	0.042749	3.065490	3.111979	75278	74473	0.025	0.372
提案手法 2 (b)	100%	2.071111	1.874910	0.196201	2.371058	2.269914	124466	121935	0.029	0.609
	60%	2.188826	2.044016	0.144810	2.492575	2.379650	98974	98498	0.015	0.492
	20%	2.396375	2.318806	0.077569	2.793871	2.696142	70520	71223	0.005	0.356
PRR	100%	3.946370	1.812510	2.133859	2.251680	2.221457	117613	76336	0.360	0.381
	60%	3.218925	1.921045	1.297880	2.294458	2.269918	92048	70059	0.250	0.350
	20%	2.658706	2.190813	0.467893	2.452511	2.424756	65923	60830	0.093	0.304
RPID	C = 20	1.929090	1.726421	0.202669	2.221253	2.170926	89596	79038	0.130	0.395
	C = 10	1.754830	1.637294	0.117536	2.361785	2.369007	87306	82136	0.071	0.410
	C = 1	1.124849	1.115591	0.009257	3.664166	3.764472	81478	82552	0.0003	0.412

が生じるピアの degree が, 隣接ピアからの複製配置ピア選択が許されるピアと許されないピアを分ける間接的な原因であると考えられ, その degree 以上のピアの書込み回数をほぼ一定にする能力が, ストレージ負荷分散にとって最も大きな要素であると言える. ただし, 隣接ピアから複製配置ピアを選択する権利の有り無しを分ける条件を今回とは違ったものにする, 今回得た結果とは違った結果を得ることになると考えられる. 以上より, 同数のファイルネットワーク上に配置するとき, その配置の仕方が負荷分散にとって重要であることが示唆される.

次に, 提案手法 1 (a) と 2 (a) の比較から, 隣接ピアからの複製配置ピア選択にポイントを用いることと degree を用いることの違いを考察する. 図 3 に, 提案手法 1 (a) を用いた場合の, degree が d_x のピア X が, degree が d_y のピア Y に与えたポイントの総数を描いたグラフを示す. 図 3 によると, 提案手法 1 (a) を用いた場合, 中・高 degree のピアは, 中・高 degree のピアよりもむしろ低 degree のピアにポイントを多く与えていたことが分かる. 従って, 隣接ピアからの複製配置ピア選択権のある中・高 degree のピアが複製配置ピアを選択するとき, 隣接ピアからの複製配置ピア選択権のない低 degree のピアが選択される確率は低かったと考えられる. 中・高 degree ピアは, それらの間の書込み回数が概ねバランスするように相互にファイル複製要求を出し合っていたと考えられる. 一方, 提案手法 2 (a) の場合, 低 degree のピアは複製配置ピア選択権を持たないことに加え, 中・高 degree のピアから複製配置ピアとして選択される確率も高くなる. これにより, 図 2(c) に示すように, 中・高 degree ピアの保持するファイル数が時に最大ストレージ容量に満たないことが起こるほど, 中・高 degree のピアが保持するファイル数が小さくなることで, 高いストレージ負荷分散が達成され, 一方で, 表 1 に示すように, 提案手法 2 (a) のファイル検索ホップ数が提案手法 1 (a) のそれよりも

増加したと考えられる.

次に, 提案手法 1 (a) と 1 (b) と 2 (a) と 2 (b) の比較から, 隣接ピアからの複製配置ピアの選択が許可されないピアを導入することの効果を検討する. まず, 提案手法 1 (a) と 1 (b) の比較であるが, 図 2(a)(b) によると, 提案手法 1 (b) を用いた場合, 全ピアに隣接ピアからの複製配置ピア選択を行う権利があるため, 提案手法 1 (a) を用いた場合よりも, 低 degree のピアへの書込み回数が減少し, 高 degree のピアへの書込み回数が増加している. このことは, 提案手法 2 (a) と 2(b) の間でも起こっている. しかしながら, 提案手法 1 (b) の読込み負荷は, 提案手法 1 (a) のそれとほぼ同等であることが分かり, さらに表 1 によると, 平均ホップ数も提案手法 1 (a) のそれとほぼ同等であることが分かる. この事実が示唆することは, 中・高 degree のピアが一定数以上のファイルを持すれば, 読込み負荷と検索ホップ数が大まかに一定以下になることである. 図 2(b)(d) によると, degree を用いて隣接ピアからの複製配置ピア選択を行った方が, ポイントを用いた場合よりも書込み負荷のバランスに優れていることが分かる. しかし, これにより, ファイル検索要求の到達確率が低い, ネットワーク上で多数を占める非常に低い degree のピアへファイルが分散して多く配置されるため, degree を用いた方が, ポイントを用いた場合よりも, 検索ホップ数, 読込み負荷が若干高いと考えられる.

最後に, RPID は, 表 1 と図 2(f) によると, 優れた書込み負荷分散を達成している. RPID は, ファイル検索要求のピアへの到達確率がピアの degree に比例すると考えられる状況で, degree に反比例する確率をピアへの複製確率として用いているため, $C = 1$ としたときに, ファイル書込み回数が全てのピアにおいておおよそ同じになることが予想される手法である. しかし, 検索ホップ数の減少を同時に達成するためには, 中・高 degree のピアが一定数以上のファイルを持する必要がある

と考えられ、その条件を満たすような C の値を設定する必要がある。今回用いたネットワーク設定に対する適切な C の値を見るために、1 から 625 までの C の値に対するストレージ負荷、書き込み負荷、読み込み負荷、初期・追加配置ファイルに対する平均ホップ数、を描いたグラフを図 4 に示す。図 4 によると、 C の値が 20 前後のとき、最もストレージ負荷分散の度合いと要した平均ホップ数のバランスが良いことが分かる。

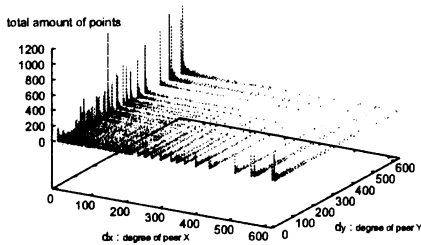


図 3 提案手法 1 (a) を用いた場合の、degree が dx のピア X が、degree が dy のピア Y に与えたポイントの総数。

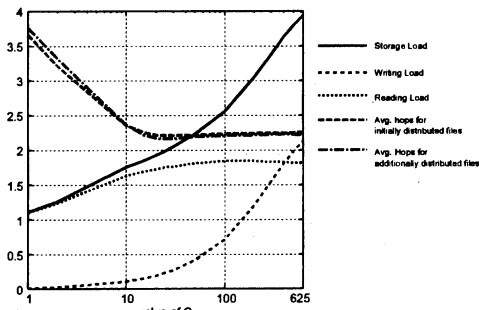


図 4 RPID を用いた場合の、 C の値と評価指標の関係。

4. おわりに

平均検索ホップ数と読み込み負荷を一定以下に抑え、かつストレージ負荷分散を達成するためには、まず、与えられた時間内に、中・高 degree のピアが一定数以上のファイルを保持する必要がある。ただし、この必要なファイル数は、ネットワーク規模やデータ容量などに依存するだろう。従って、ストレージ負荷分散の中心は、ファイル書き込み回数の分散を行うことにある。提案手法 1 (a) と RPID は、パラメータ調整を適切に行うと、平均検索ホップ数の減少とストレージ負荷分散とのバランスに関して、最も優れていると言える。これは、平均検索ホップ数と読み込み負荷を一定以下に抑えるために必要なファイル数が中・高 degree のピアに確保された上で、ファイル書き込み回数の分散が行われているからだと考えられる。

提案手法 1 (a) と RPID の間には、提案手法 1 (a) が検索履歴を用い、RPID が degree を用いる、などの方法論の違いはあるが、基本的には、degree に反比例するような確率で複製配置するピアを選択する、あるいは実際に複製配置することでファ

イルの書き込み回数を分散している。今後、異なる静的なトポロジや動的に変化するトポロジを含め、異なるネットワーク設定に対して、それぞれの手法の長所と短所を明らかにしていく必要がある。

謝 辞

本研究は、総務省「ユビキタスネットワーク技術の研究開発」からの委託研究成果である。また、九州大学 山下雅史教授、小野廣隆助手より本研究に関する有益なご助言を頂いた。

文 献

- [1] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman. Search in power-law networks. *Physical Review E* 64, 046135, 2001.
- [2] R. Albert and A.-L. Barabasi. Topology of evolving networks: Local events and universality. *Physical Review Letters*, pages 5234–5237, December 2000.
- [3] S. Ata, Y. Gotoh, and M. Murata. Replication strategies in peer-to-peer services over power-law overlay networks. In *Proceedings of The 7th Asia-Pacific Network Operations and Management Symposium (APNOMS 2003)*, Fukuoka, Japan, October 2003.
- [4] A.-L. Barabasi and R. Albert. Emergence of scaling in random networks. *SCIENCE*, 286:509–512, October 1999.
- [5] T. Bu and D. Towsley. On distinguishing between internet power law topology generators. In *Proceedings of IEEE Infocom 2002*, volume 5244, pages 638–647, New York, NY, USA, June 2003.
- [6] E. Cohen and S. Shenker. Replication strategies in unstructured peer-to-peer networks. In *Proceedings of the ACM SIGCOMM 2002 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, Pittsburgh, PA, USA, August 2002.
- [7] R. Hsiao and S.-D. Wang. Jelly: A dynamic hierarchical p2p overlay network with load balance and locality. In *Proceedings of 24th International Conference on Distributed Computing Systems Workshops - W4: MDC (ICDCSW'04)*, pages 534–540, Tokyo, Japan, March 2004.
- [8] D. R. Karger and M. Ruhl. Simple efficient load balancing algorithms for peer-to-peer systems. In *Proceedings of the sixteenth annual ACM symposium on Parallelism in algorithms and architectures*, Barcelona, Spain, June 2004.
- [9] Q. Lv, P. Cao, E. Cohen, and S. Li, K. Shenker. Search and replication in unstructured peer-to-peer networks. In *Proceedings of the 16th international conference on Supercomputing*, pages 84–95, New York, USA, June 2002.
- [10] R. Motwani and P. Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.
- [11] A. Rao, K. Lakshminarayanan, S. Surana, R. Karp, and I. Stoica. Load balancing in structured p2p systems. In *Proceedings of 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)*, Berkeley, CA, USA, February 2003.
- [12] M. Ripeanu, A. Iamnitchi, and I. Foster. Mapping the gnutella network. *IEEE Internet Computing*, 6(1):50–57, January/February 2002.
- [13] H. Yamamoto, D. Maruta, and Y. Oie. Replication methods for load balancing on distributed storages in P2P networks. In *Proceedings of The 2005 International Symposium on Applications and the Internet (SAINT2005)*, pages 264–271, Trento, Italy, February 2005.
- [14] Y. Zhu and Y. Hu. Efficient, proximity-aware load balancing for DHT-based p2p systems. *IEEE Transactions on Parallel and Distributed Systems*, 16(4):349–361, April 2005.