

ピアによるファイル提供の局所制御に基づくストレージ負荷分散

大西 圭[†] 市川 憲人[†] 山本 寛^{††} 内田 真人[†] 尾家 祐二^{††}

[†]九州工業大学 ネットワークデザイン研究センター
〒 802-0001 福岡県北九州市小倉北区浅野 AIM ビル

^{††}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: †{ohnishi,ichikawa,m.uchida}@ndrc.kyutech.ac.jp, ††yamamoto@infonet.cse.kyutech.ac.jp,
oie@cse.kyutech.ac.jp

あらまし ファイル共有を目的とした非構造型 Peer-to-peer(P2P) ネットワークのための従来のストレージ負荷分散手法は、主にファイルの複製をピアに生成する際に発生する負荷(ファイル書込み負荷)に注目してきた。本論文は、検索要求ファイルを持つピアがそのファイルをストレージから読み出し提供する際に発生する負荷(ファイル読み込み負荷)に注目し、そのような負荷をピア間で分散する手法を提案する。提案手法は、要求ファイルを持つ注目ピアとその隣接ピアの間で過去のファイル提供回数を比較し、その比較に基づいて注目ピアがファイル提供を行うか否かを決定する。しかしながら、検索要求ファイルを持つピアがそのファイルの提供を拒むことは、ファイル検索ホップ数の増加を引き起こす可能性がある。そこで、ファイル検索ホップ数に関する制約の下で、出来る限り読み込み負荷を分散させる手法パラメータ値の調整を遺伝的アルゴリズムを用いて行う。実験の結果は、調整されたパラメータ値を持つ提案手法とファイル書込み負荷分散に優れたファイル複製配置手法を組み合わせた手法が、優れた大局的ストレージ負荷分散を達成することを示す。

キーワード peer-to-peer(P2P) ネットワーク, ストレージ負荷分散, ファイル読み込み, 複製配置手法

Storage Load Balancing via Local Control of Providing Files by Peers

Kei OHNISHI[†], Kento ICHIKAWA[†], Hiroshi YAMAMOTO^{††}, Masato UCHIDA[†], and Yuji OIE^{††}

[†] Network Design Research Center, Kyushu Institute of Technology

3-8-1 Asano, Kokura-Kita-ku, Kitakyusyu-shi, Fukuoka 802-0001, Japan

^{††} Department of Computer Science and Electronics, Kyushu Institute of Technology

680-4 Kawazu, Iizuka-shi, Fukuoka 820-8502, Japan

E-mail: †{ohnishi,ichikawa,m.uchida}@ndrc.kyutech.ac.jp, ††yamamoto@infonet.cse.kyutech.ac.jp,
oie@cse.kyutech.ac.jp

Abstract The conventional storage load balancing methods for unstructured peer-to-peer (P2P) networks for file sharing have focused mainly on load that is caused when replicas of requested files are created in peers, which is referred to as *load of writing files*. This paper focuses on load that is caused when requested files are read and provided from peers with those files, which is referred to as *load of reading files*, and presents a method that balances such a load among peers. The presented method determines whether or not a peer of interest with a requested file provides the requested file based on the comparison between the numbers of times that files were read from the peer of interest and its neighboring peers. However, there is possibility that refusal of providing a requested file by a peer causes the increases in the number of hops required for successful search. Therefore, we adjust the parameter values of the presented method by a genetic algorithm under the constraint on the number of hops. The experimental results show that the combination of a conventional replication method with high ability in balancing a load of writing files and the presented method yields good global storage load balancing.

Key words peer-to-peer(P2P) network, storage load balancing, reading files, replication method

1. はじめに

P2P ネットワークの応用として注目を集めているものの一つに、ファイル共有システムがある。これは、高価で大記憶容量を持つファイルサーバを導入することなく、各ピアの未使用の記憶スペースを集めることで、ネットワーク全体で大容量の記憶スペースを提供する。ファイル共有を目的とした P2P ネットワークには、実際いくつかの形態がある [21] [13]。ファイル共有を目的とした P2P ネットワークの形態を最も大まかに分けると、ネットワーク上のピアにファイルをどのように分散配置するかを決定する方法を持つネットワーク（構造型 P2P ネットワーク）と、そのような方法を持たないネットワーク（非構造型 P2P ネットワーク）になる。

我々はこれまで、ファイル共有を目的とした非構造型 P2P ネットワークを想定し、ネットワークに含まれるピアの間のストレージ負分散と検索性能の間のトレードオフをより良く開拓する複製配置手法に関する研究を行ってきた [18] [16]。我々が以前に提案した複製配置手法は、基本的に、各ピアの現時点での負荷状態に応じてファイルの複製を生成する確率あるいはその生成場所を決定するものである。このような制御によって達成されることは、主にファイルの複製をピアのストレージに生成する際に生じる負荷（以下、ファイル書き込み負荷）の分散である。

本論文では、検索要求ファイルを持つピアがそのファイルをストレージから読み出して提供する際に生じる負荷（以下、ファイル読み込み負荷）に注目し、そのようなファイル読み込み負荷分散を行う手法を提案する。ここでは、前述したファイル書き込み負荷とここで述べたファイル読み込み負荷の和をストレージ負荷と考える。検索要求ファイルを持つピアがその提供を拒否することは、そのピアのファイル読み込み負荷の軽減につながる。一方で、ファイル検索に要するホップ数の増加を引き起こす可能性がある。そこで、本論文では、検索ホップ数に関する制約の下で、より良いストレージ負分散性能を達成する手法パラメータ値の調整を行う。ただし、本論文で提案する手法はファイル複製配置手法ではないため、提案手法を従来のファイル書き込み負荷分散に優れた複製配置手法と組み合わせる。

本論文の構成は次のようである。第 2 節では、関連研究を短く述べる。第 3 節では、提案するファイル読み込み負荷分散手法について説明する。この提案手法はパラメータを持つため、第 4 節において最適化手法である遺伝的アルゴリズム (GA) によりパラメータの最適化を行う。このとき、ファイル検索ホップ数を制約条件とする。第 5 節では、最適化されたパラメータの値を持つ提案手法と従来の複製配置手法との組合せを、ストレージ負分散性能の観点から他の複製配置手法と比較し評価する。第 6 節は、我々の結論である。

2. 関連研究

これまで複製配置手法は、主にファイルの場所を管理する機構が存在しない非構造型 P2P ネットワークにおいて、高速で確実なファイル検索を達成するために開発されてきた [6] [14]。しかし、それらにおいては、検索効率とトレードオフ関係にあるピア間の負荷分散が考慮されていない。一方、負荷分散手法は、主に分散ハッシュ表 (DHT) を用いる構造型 P2P ネットワークのために開発されてきた [17] [12]。DHT を用いる構造型 P2P ネットワークは固定のファイル配置場所と目的ファイルまでの確実な経路を提供し、それにより複製配置手法を必要としない。しかし、予め固定するファイルの場所やトポロジ、およびファイル間の人気の偏りにより、負荷が一部のピアに偏る可能性がある。

我々の先行研究 [18] [19] [16] は、非構造型 P2P ファイル共有ネットワークのための複製配置手法に焦点を当てたが、その目的は、従来の非構造型 P2P ネットワーク研究が目的とした検索性能の向上だけでなく、従来の構造型 P2P ネットワーク研究が目的としたストレージ負分散でもあった。文献 [18] において提案された複製配置手法は、個々のピアの負荷状態に応じてファイルの複製を生成する確率（以下、複製率）をピアに与

える。一方、文献 [19] において提案された複製配置手法は、自ピアのストレージ利用率と隣接ピア群のストレージ使用率を比較し、局所的にストレージ使用率がバランスするように自ピアの複製率を調整する。この複製率の調整機構は、熱拡散現象とのアナロジーを持つ。[16] において提案された複製配置手法は、現検索経路上のピアのストレージ負荷とその隣接ピア群のストレージ負荷を比較し、隣接ピア群の中でストレージ負荷が少ないピアにファイルの複製を生成させる。この複製配置手法を、本論文で提案する読み込み負荷分散手法と組み合わせる。

隣接ノード間の相互作用による負荷分散の方法は diffusive method と呼ばれ、主に grid 計算の分野で研究されてきた [7] [4] [8]。複製配置手法によるストレージ負分散と grid における計算負荷分散の間には、次のような本質的な違いがある。grid においては、高速な計算タスクの実行が唯一の目的であり、その唯一の目的を計算負荷分散を通じて達成する。一方、非構造型 P2P ファイル共有ネットワークにおいては、高速かつ確実なファイル検索の提供が機能上の主目的であり、複製配置手法を通じて目的を達成するが、その副次的な作用として負荷の偏りが生じるため、複製配置手法はそれらトレードオフ関係にあることがらを考慮したものでなくてはならない。したがって、非構造型 P2P ファイル共有ネットワークにおけるストレージ負荷の分散に grid における計算負荷分散手法を直接適用することは難しい。

以上のこれまでの P2P ネットワークにおけるストレージ負分散の研究は、ファイルをストレージに書き込むことによって生じる負荷に焦点を当ててきた。ここでは、検索要求ファイルを持つピアがストレージからそのファイルを読み込む際に生じる負荷に注目する。この点に注目した研究はこれまでにない。

3. ファイル読み込み負荷分散手法

本節では、ファイル共有・交換を目的とした P2P ネットワークのためのファイル読み込み負荷分散手法を提案する。

3.1 基本的な考え方

ランダムウォークに基づくファイル検索手法を用いるとき、ウォーカーのピアへの到達確率は、ノードの次数 (degree) に比例する [15]。従って、たとえファイル書き込み負荷が分散され、全てのピアが持つファイルの数が均一になったとしても、ファイルのピアからの読み込み回数は、ピアの次数に比例することになる。このファイルのピアからの読み込み回数を分散させることができれば、さらなるストレージ負分散が達成できる。

しかしながら、ファイルの読み込み回数が多い次数の高いピアがファイルの提供を拒否すれば、ファイル検索ホップ数が大きくなる可能性がある。そこで、検索ホップ数の増加を小さく抑え、同時にファイルのピアからの読み込み回数を分散させる方法を考える必要がある。

3.2 ピアによるファイル提供の局所的制御

まず、式 (1) に示すように、注目ピアの過去のファイル読み込み回数とその隣接ピア群の平均ファイル読み込み回数から、値 R_c を得る。この R_c を非線形システムに入力し、ピアがファイルを提供するか否かを決定する。

$$R_c = \begin{cases} 1 - \frac{r_f}{r_a} & \text{if } r_f \geq r_a, \\ \frac{r_f}{r_a} - 1 & \text{otherwise.} \end{cases} \quad (1)$$

ここで、 r_f は注目ピアの過去のファイル読み込み回数であり、 r_a はそのピアに隣接するピア群の過去の平均読み込み回数である。 R_c は、 $[-1, +1]$ の範囲にある実数となる。

本論文で用いる非線形システムは、式 (1) で表される R_c を入力としたファジィシステムである。ファジィ推論の方法としては、ファジィルールの後件部が定数で与えられる簡略型推論を用いる。簡略型推論を行うファジィシステムのファジィルールは、以下のように表される。

R_c : If x_1 is A_{i1} , x_2 is A_{i2} , ..., and x_n is A_{in} , then y is C_i .

ここで、 i はルール番号、 n は入力変数の数、 x_k ($1 \leq k \leq n$) は入力変数、 A_{ik} はメンバーシップ関数で表現されるファジィ集合、 y は出力変数、 C_i は定数、である。このとき、簡略型ファジィ推論は、式 (2) によって行われる。

$$y = \frac{\sum_{i=1}^P w_i C_i}{\sum_{i=1}^P w_i} \quad (2)$$

ここで、 P はファジィルール数、 w_i は i 番目のルール強度である。ルール強度は、式 (3) で表される。

$$w_i = \prod_{k=1}^n m_k \quad (3)$$

ここで、 m_k は、 k 番目のメンバーシップ関数値である。

注目ピアが検索要求ファイルを持っているとき、以下の式 (4) により式 (2) で表される y を 1 か -1 に変換する。このとき、変換された値 R が 1 のとき、その注目ピアは検索要求ファイルを提供し、-1 のとき検索要求ファイルの提供を拒否する。この式 (4) によりピアのファイル提供を制御する方法を、提案手法 1 と呼ぶ。

$$R = \begin{cases} 1 & \text{if } y \geq 0, \\ -1 & \text{otherwise.} \end{cases} \quad (4)$$

また、注目ピアが検索要求ファイルを持っているとき、以下の式 (5) により式 (2) で表される y を 0 以上 1 以下の実数に変換する方法も考える。ここで変換された値 R は、ファイルを提供する確率を表す。この式 (5) によりピアのファイル提供を制御する方法を、提案手法 2 と呼ぶ。

$$R = \begin{cases} 0 & \text{if } y < 0, \\ 1 & \text{if } y > 1, \\ y & \text{otherwise.} \end{cases} \quad (5)$$

4. GA による手法パラメータ最適化

本節では、前節に示した 2 つの提案手法 (提案手法 1 と提案手法 2) と複製配置手法の組合せがストレージ負荷分散と探索性能の間のトレードオフ関係をより良く開拓するように、最適化手法を用いて提案手法の持つパラメータ (ファジィシステムのパラメータ) を最適化する。その最適化手法として遺伝的アルゴリズム (以下、GA と呼ぶ) [10] [11] を用いる。以下では、シミュレーションの詳細を説明する。

4.1 P2P シミュレーションモデル

本論文で用いられる P2P シミュレーションモデルは、文献 [16] で用いられたものと同じである。以下に簡単に説明する。ネットワーク上に存在するピアの総数を 10,000 とし、ピア間のリンクの総数を 20,000 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、その次数の頻度分布はべき法則 (power-law) [3] [2] [1] に従う。本論文で用いるトポロジは、文献 [16] で用いられたものと同じである。

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 [14] を用いる。これは、ファイル検索要求を発生したピアから、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生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。

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。本論文で用いられるファイルの初期配置は、文献 [16] で用いられたものと同じである。また、ファイル間に人気の偏りはないものとする。

4.2 ファジィシステムと GA の設定

まず、ファジィシステムの設定であるが、ファジィメンバーシップ関数として三角形メンバーシップ関数を用いる。ファジィシステムへの入力は、式 (1) で表される R_c の 1 個であり、その出力 y は、提案手法 1 の場合、式 (4) によって +1 (ファイル提供) か -1 (ファイル提供拒否) に変換され、提案手法 2 の場合、式 (5) によってファイルを提供する確率に変換される。1 個のファジィシステムにおける三角形メンバーシップ関数の数は 10 とする。最適化手法である GA は、図 1 に示される $center_i, left_i, right_i$ ($i = 1, 2, \dots, 8$) とファジィルールの後部部の定数 C_i を後述する基準の下で最適化する。

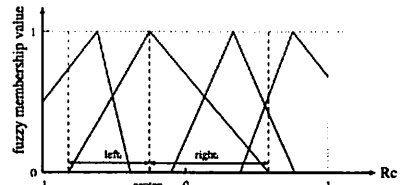


図 1 ファジィメンバーシップ関数。

次に、上述したファジィシステムのパラメータの値を最適化する GA は、実数コーディングを行う実数値 GA とする。その実数値 GA の世代交代モデルとして Minimal Generation Gap [20] を用い、交叉演算として BLX- α [9] (α の値として 0.36) を用いる。突然変異演算は用いない。実数値 GA が最適化するパラメータの範囲は、 $center_i \in [-1, 1], left_i \in [0, 1], right_i \in [0, 1], C_i \in [0, 1]$ 、である。また、ファジィルールの後部部の値 C_i の探索範囲は、 $[-1, 1]$ とする。BLX- α 交叉により、パラメータの値がパラメータ範囲を超えたときは、範囲の最外値に補正される。10 個のファジィルールを持つファジィシステムのパラメータの数は 40 であるので、実数値 GA は計 40 個のパラメータを最適化する。用いる実数値 GA の個体数を 100、その実行終了条件を 5,000 回の評価値計算とする。

また、ファジィシステムのパラメータから成る GA 個体の評価は、前節に述べた P2P シミュレーションモデルを実行し、文献 [16] に示した評価基準により評価する。つまり、ストレージ負荷分散度合いの指標は、横軸にピアの次数、縦軸をピアからのファイル読み込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、各ピアの読み込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 (以降ではストレージ負荷係数と呼ぶ) とする。ここで実数値 GA は、このストレージ負荷係数をより小さくする個体を探索する。読み込み回数と書き込み回数度合いは (以降ではそれらを読み込み負荷係数、書き込み負荷係数と呼ぶ)、それぞれ単独で GA 個体の評価に用いられないが、ストレージ負荷係数を得るのと同様な方法で得る。GA の実行の際に、GA 個体の中には、全てのピアにファイルの提供を拒否させるものがある。つまり、一切ピアへのファイルの複製生成が起こらず、結果としてファイル書き込み負荷が発生しないことがある。このとき、そのような GA 個体には評価値として強制的

に 10^5 が与えられる。

探索性能の指標は、平均ホップ数とし、実数値 GA が最小のストレージ負荷係数を与える GA 個体を探索する際の制約条件として用いられる。つまり、GA 個体の平均ホップ数 h が予め与えられた値 H を超えたとき、その個体にペナルティを与える。具体的には、GA 個体のストレージ負荷係数に、 $10 \times (h - H)$ の値が加えられる。この H として、本論文では、2.2, 2.4, 2.6, 2.8 の 4 種類を用いる。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、10,001 回目から 30,000 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、20,001 回目から 40,000 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。

4.3 提案手法と複製配置手法の組合せ

2つの提案手法(提案手法1と提案手法2)と組み合わせられる複製配置手法として、文献[16]の中で提案した、ポイントとファイル検索経路上に位置した回数を用いる複製配置手法(以下では RUP と呼ぶ)を用いる。詳細は文献[16]に示されているが、RUP は、各ピアの過去の検索への貢献を考慮して現検索経路上のピアだけでなく、経路上のピアの隣接ピアも複製配置の対象にする。RUP は、これまでで最も良くストレージ負荷分散と検索性能の間のトレードオフ関係を開拓できる複製配置手法の1つである。用いる複製配置手法の複製確率は100%とする。

以降では、RUP と提案手法1および RUP と提案手法2との組合せを、それぞれ、RUP+提案手法1、RUP+提案手法2のように呼ぶ。

4.4 シミュレーション結果と考察

前節に示した RUP+提案手法1 と RUP+提案手法2 について、4 種類の H の値を制約条件とした GA を実行した。各 GA の実行で最終的に得たストレージ負荷分散と平均ホップ数を、RUP+提案手法1については表1に、RUP+提案手法2については表2示す。また、そのときの得られた最良個体から生成される式(4)の R の値と式(5)の R を、それぞれ図2と図3に示す。また、最良個体が得られたときのストレージ負荷分散、読み込み負荷分散、書き込み負荷分散を表すグラフを図4と図5に示す。

表1 RUP+提案手法1を用いた場合の実数値 GA 実行により得た最良結果。

制約条件 H	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
2.2	1.800868	1.420486	0.374382	2.199561	2.161687
2.4	1.256012	0.802839	0.453173	2.396300	2.354437
2.6	1.078834	0.523464	0.555371	2.587944	2.593407
2.8	1.032834	0.370360	0.662474	2.781090	2.764381

表2 RUP+提案手法2を用いた場合の実数値 GA 実行により得た最良結果。

制約条件 H	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
2.2	2.004938	1.631464	0.372812	2.200066	2.082913
2.4	1.419386	0.949540	0.469846	2.399219	2.389757
2.6	1.124624	0.586397	0.538227	2.587573	2.568872
2.8	1.030814	0.401272	0.629543	2.798133	2.687467

まず、表1と2によると、RUP+提案手法1とRUP+提案手法2ともに、ホップ数の制約が緩くなる、つまり H の値の増加に従いより高いストレージ負荷分散を達成していることが分かる。また、その理由が、 H の値の増加に伴い書き込み負荷係数が増加するが、その増加の大きさを上回る大きさで読み込み負荷係数が減少しているからだと分かる。

次に、図2によると、どの H の値についても、ある R_c の値

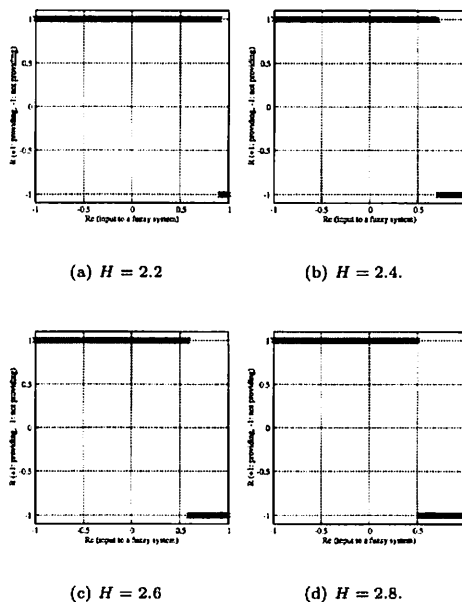


図2 RUP+提案手法1を用いた場合の実数値 GA 実行により得た最良個体が生成する式(4)の R の値。1はピアのファイル提供、-1はピアのファイル提供拒否を意味する。

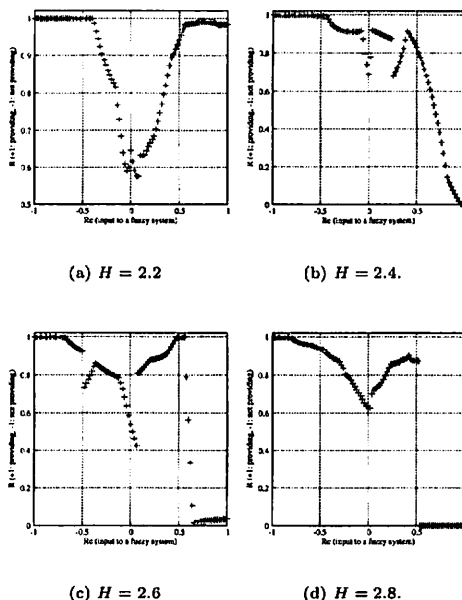


図3 RUP+提案手法2を用いた場合の実数値 GA 実行により得た最良個体が生成する式(5)の R の値。値はファイルを提供する確率を表す。

を境に、ピアがファイルを提供するか否かが分かることが分かる。より具体的には、注目ピアのファイル読み込み回数とその隣接ピア群の平均読み込み回数より多いある点、つまり式(1)の R_c が正の値になる範囲のある点で、そのピアはファイルの提

供を拒否するようになることが分かる。さらに、ファイル検索ホップ数の制約が緩くなる、つまり H の値の増加にともない、ピアがファイルを提供するか否かを決定する R_c の値が、-1 側に移動する。これは、検索により多くの検索ホップ数を費やすことが可能になることで、ファイルの読み込み回数の多いより多くのピアがファイルの提供を拒否できるようになるからと考えられる。

また、図 3 は、 R の値が提供か否かという 2 値ではなく提供確率を表す連続値ということもあり、図 2 と全く同じではないが、 $H = 2.2$ のときを除き、図 2 と傾向が似ていることが分かる。つまり、ある R_c の値を境に非常に高いファイル提供確率と非常に低いファイル提供確率に分かれる。

RUP+提案手法 1 の場合、どの H の値についても、ピアがファイルを提供するか否かを、 R_c についての 1 個の値から決定される。この事実は、制御の観点からは有用だと考えられる。一方、RUP+提案手法 2 の場合、ピアがファイルを提供するか否かを、1 個の値から決定することができない。さらにそれは、RUP+提案手法 1 のストレージ負荷分散性能と同等か若干劣る性能しか示せていない。従って、RUP+提案手法 1 と RUP+提案手法 2 を比較した場合、性能の観点からも用いやすさの観点からも、RUP+提案手法 1 が優れているといえる。今後、異なるネットワークポロジに対しても同様なことが言えるかを調査する必要がある。

5. 他の複製配置手法との比較

前節において得られた最も良い負荷分散性能を与えるファジシステム (GA 個体) を含む提案手法と既存複製配置手法の組合せ手法 (RUP+提案手法 1 と RUP+提案手法 2) を、既存の複製配置手法 (RUP) と比較する。

各手法について、ネットワーク・トポロジが一定のシミュレーション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、ストレージ負荷と平均ホップ数に加え、ピアへの総読み込み回数、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。ピアへの総読み込み数は、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、古いファイルへの書き込みが起った度合いを知ることが出来る。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割合を使用しているかを示す値である。結果を表 3 に示す。

表 3 によると、RUP+提案手法 1 と RUP+提案手法 2 は、 $H = 2.2$ のとき、RUP とストレージ負荷分散性能の差があまりないが、 H がそれより大きくなると、高い読み込み負荷分散性能を達成することで、RUP のストレージ負荷分散性能を大きく改善していることが分かる。ホップ数 H の増加を許すことで、ネットワーク全体のファイル書き込み回数とファイル数が増加するが、それにより書き込み負荷係数が大きく増加しない理由は、RUP のファイル書き込み回数の分散性能が高いからだと考えられる。

実際の P2P ファイル共有ネットワークにおいては、ユーザが許容できる検索時間 (検索ホップ数)、およびネットワークが許容するトラフィック量がそれぞれ異なり、ホップ数を犠牲にすることで高いストレージ負荷分散性能を得ることが必ずしも良いとは言えない。しかしながら、ユーザが許容できる検索時間やネットワークが許容するトラフィック量が大きいときには、提案手法は、大きなストレージ負荷分散性能の改善に寄与するといえる。ただしそのとき、書き込み負荷分散性能に優れた複製配置手法の併用が必要だと考えられる。

6. おわりに

本論文は、ピアのストレージからファイルを読み込む際に生じる負荷に注目し、その負荷をピア間で分散させる手法を提案した。提案手法を、ピアのストレージへファイルを書き込む際に生じる負荷の分散に優れた複製配置手法と組み合わせ、シミュレーションにより評価した。

シミュレーションの結果は、適切にパラメータ値が調整された提案手法は、ファイルの読み込み負荷を分散させるだけでなく、探索性能をそれほど犠牲にせずに優れた大局的ストレージ負荷分散に寄与することを示した。さらに、この事実は、ファイルの書き込み負荷分散に優れた複製配置手法とファイル読み込み負荷分散に優れた手法は、互いに干渉して負荷分散性能を劣化させることなく、協調して高い大局的負荷分散性能を生み出すことを示唆する。今後は、異なるネットワークポロジに対しても提案手法の有効性を確認する。

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。

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, 046195, 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] A.-L. Barabasi and R. Albert. Emergence of scaling in random networks. *SCIENCE*, 286:509–512, October 1999.
- [4] J. Boillat. Load balancing and poisson equation in a graph. *Concurrency Practice and Experience*, 2(4):289–313, December 1990.
- [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] G. Cybenko. Dynamic load balancing for distributed memory multiprocessors. *Parallel and Distributed Computing*, 7(2):279–301, October 1989.
- [8] S. Douglas and A. Harwood. Diffusive load balancing of loosely-synchronous parallel programs over peer-to-peer networks. <http://arxiv.org/abs/cs.DC/0410009>, 2004.
- [9] L. J. Eshelman and D. J. Shaffer. Real-coded genetic algorithms and interval-schemata. *Foundations of Genetic Algorithms 2*, pages 187–202, 1993.
- [10] D. E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Addison-Wesley, Reading, MA, 1989.
- [11] D. E. Goldberg. *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Kluwer Academic Publishers, Norwell, MA, 2002.
- [12] 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.
- [13] 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*, 7(2):72–93, Second Quarter 2005.
- [14] Q. Lv, P. Cao, E. Cohen, and S. Li, K. Shenker. Search and replication in unstructured peer-to-peer networks. In

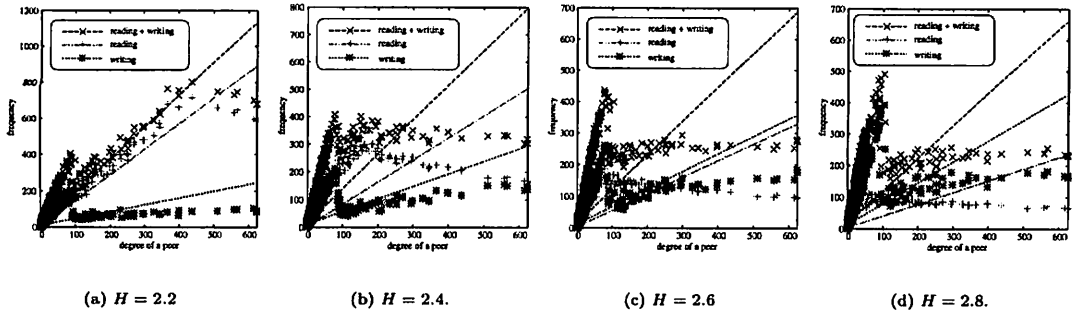


図 4 RUP+提案手法1を用いた場合の各種ストレージ負荷係数.

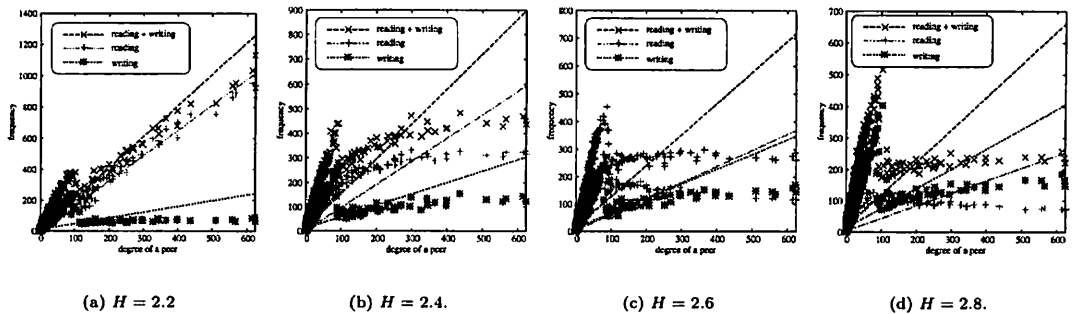


図 5 RUP+提案手法2を用いた場合の各種ストレージ負荷係数.

表 3 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. RUP の結果は, 文献 [16] に示したものである.

	複製 確率	H	全負荷	読負荷	番負荷	初期データ ホップ	追加データ ホップ	総書込 回数	総フ ァイル数	書込 冗長度	容量 使用度
RUP+提案手法1	100%	2.2	1.818189	1.447949	0.370239	2.224180	2.192865	118645	92999	0.225	0.464
	100%	2.4	1.273515	0.800826	0.472689	2.419427	2.391463	129249	99320	0.240	0.496
	100%	2.6	1.101328	0.524077	0.577251	2.593912	2.551356	139854	104041	0.263	0.520
	100%	2.8	1.056212	0.369941	0.686271	2.784034	2.728333	152159	108260	0.295	0.541
RUP+提案手法2	100%	2.2	2.038264	1.654311	0.383954	2.204877	2.159886	117596	91148	0.234	0.455
	100%	2.4	1.423450	0.949606	0.473844	2.414901	2.382742	128557	97565	0.249	0.487
	100%	2.6	1.138822	0.586499	0.552323	2.613503	2.567244	139303	102635	0.271	0.513
	100%	2.8	1.041196	0.401978	0.639218	2.798952	2.743351	151951	107132	0.302	0.535
RUP	100%	-	2.008805	1.654479	0.354325	2.186222	2.145899	116665	91812	0.222	0.459

Proceedings of the 16th international conference on Super-computing, pages 84-95, New York, USA, June 2002.

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

[16] K. Ohnishi, H. Yamamoto, K. Ichikawa, M. Uchida, and Y. Oie. Storage load balancing via local interactions among peers in unstructured P2P networks. In *Proceedings of the International Workshop on Peer-to-Peer Information Management (P2PIM)*, Hong Kong, May 2006 (to appear).

[17] 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.

[18] H. Yamamoto, D. Maruta, and Y. Oie. Replication method for load balancing on distributed storages in P2P networks. *IEICE Transactions on Information and Systems - Special*

Section on New Technologies and their Applications of the Internet III, January 2006.

[19] 市川 憲人, 大西 圭, 内田 真人, 筒 正人, and 尾家 祐二. P2p ファイル共有アプリケーションにおける負荷分散を実現する複製配置手法—熱拡散現象とのアナロジー—. In *電子情報通信学会技術研究報告*, 3 2006 (to appear).

[20] 佐藤 浩, 小野 功, and 小林 重信. 遺伝的アルゴリズムにおける世代交代モデルの提案と評価. *人工知能学会誌*, 12(5):734-744, 1997.

[21] 星合 隆成 須永 宏. P2P 総論 [II] —P2P デクノロジー—. *電子情報通信学会誌*, 87(10):887-896, 10 2004.