peer-to-peer システム上での評価の高い peer の発見

Efficient Peer Discovery on peer-to-peer Systems

山田 太造[§] 相原 健郎[¨]
高須 淳宏[a] 安達 淳[©]

Taizo YAMADA Kenro AIHARA
Atsuhiro TAKASU Jun ADACHI

P2P システムはここ数年の間で爆発的な成長を遂げている．そこでのユーザ数および共有データ量はともに膨れ上がる一方で，ネットワーク全体の帯域幅消費量も急増している．また，通常の P2P システムではノード間のメッセージ転送で問い合わせを処理するため効率はよくない．そこで本研究では各ノード上に評価の高い他のノード情報が格納された分散インデックス(*Direct Indices*；以下 *DIs*) を提案し，DIs に示された他のノードに対して直接問い合わせを行う．これにより問い合わせに対する帯域幅消費が減少し、スケーラビリティのあるネットワークの実現が可能となる．また各種実験を行うことでそのパフォーマンスを示す．

Recently, P2P systems have been growing explosively and both the number of users and the amount of shared data increase rapidly. It causes the tremendous consumption of bandwidth of the whole network. Query processing of P2P system is inefficient because it propagates query messages in the P2P network. This paper proposes the concept of *Direct Indices* (*DIs*), which keep highly evaluated nodes. Since DIs enable peers to access the data directly, bandwidth consumption is reduced and scalability of P2P network is achieved. This paper shows the performance of DIs experimentally.

1. はじめに

近年，peer-to-peer（P2P）システムは爆発的な成長を遂げている．P2Pシステムは，役割や能力の等しいノードで構成される分散コンピューティングシステムであり，各ノード間で直接情報のやりとりを行う．そこに参加しているユーザ数，共有データ量は年々増加している．通常，中央にサーバを持たないP2P（Pure P2P;PP2P）システムでは，問い合わせがノード（peer;ピア）で解決できない場合，隣接するノードへその問い合わせを転送することによって処理される．その例としてGnutella[3]やFreenet[4]等が有名である．これらのPP2Pシステムでは，問い合わせは各ピアから隣接する全ピアへ転送されるために，問い合わせのメッセージ数は指数関数的に増加してしまう．このため，ネットワーク全体における帯域幅消費量も莫大な量になっている．P2Pシステムでは，帯域幅消費を減少させることは重要な課題となっている．

本研究ではP2Pシステムでの帯域幅消費の軽減と効率的な問い合わせ方法について考察し，各ピア上で効率的な問い合わせ処理を行うための新たな分散インデックス（Direct Indices;DIs）を提案する．DIsでは，各ピアは他のピアの有用度に関する情報を持ち，この情報に基づいて問い合わせを発行/転送するピアを決定する．ピアの有用度はアクセス頻度などに基づいて時間とともに変化し，その情報はピア間で伝播する．有用な情報に対して直接問い合わせを行うことによって，問い合わせの伝播量を減らし，スケーラビリティの改善をもたらす．

しかし，分散インデックスを導入するためには以下の問題点を考慮しなければならない:

- 各ピアでのインデックスの更新時（join / leave / update）の帯域幅消費を抑えながら，効率的でスケーラビリティのあるインデックス構築可能なメカニズムを導入する必要がある．
- 問い合わせ発行時の帯域幅消費を減少させるメカニズムを導入する必要がある．インデックスを用いない方法に比べネットワーク全体の帯域幅消費を減少させるため，更新と問い合わせのトレードオフの関係を考慮する必要がある．

本論文では，まず2節では関連研究をサーベイする．次に3節ではDIsの概念を導入し，そのアルゴリズムを示す．4節ではDIs を用いた場合とそうでない場合の比較実験を行ない，DIs を用いることでネットワークパフォーマンスが改善されることを示す．

2. 関連研究

P2Pシステムは，インデックスの観点から以下のように分類できる．まず，インデックスを全く用いないシステムでGnutella[3]はこの種類に属する．このタイプのシステムは問い合わせの伝播が全方位に行われるため，問い合わせのメッセージ数が膨大になり，システム全体のパフォーマンス低下を招く欠点がある．

第2のグループは，インデックスをsuper peerで集中管理するタイプで，Napster[1]やOpenNap[2]などがあげられる．しかし，このタイプのシステムは，super peerが停止してしまうとシステム全体も停止してしまう．そのため，super peerの管理コストの問題がある．

第3のグループは，各ピアがインデックスを保持するものでFreenet[4]などが例としてあげられる．Freenetでは各ピアは最近参照されたデータをキャッシュに蓄える．しかし，このインデックス内に存在しないデータを検索する場合の検索方法は考慮されていない．Yang等は，routing indexを提案している[5]．このインデックスでは，P2Pシステムが扱うデータのカテゴリを定義し，このカテゴリごとにデータを多く有するピアの情報をインデックスとして用いる．しかし，あらかじめ各データを分類する方法を考慮する必要があり，スケーラビリティに優れているとは言い難い．

この他にもP2Pシステム上での効率の良い情報検索法の研究が行われており，[6]ではインデックスを用いた方法と用いない方法を示している．[7, 8, 9, 10, 11, 12]ではP2Pシ

♣ 学生会員 総合研究大学院大学数物科学研究科博士後期課程 t_yamada@grad.nii.ac.jp
♦ 非会員 国立情報学研究所ソフトウェア研究系 / 総合研究大学院大学 kenro.aihara@nii.ac.jp
♠ 正会員 国立情報学研究所ソフトウェア研究系 / 総合研究大学院大学 takasu@nii.ac.jp
♥ 正会員 国立情報学研究所教授 adachi@nii.ac.jp

ステムでの効率的なデータ検索が可能である．しかし，これらの研究では，特別なネットワーク構造内でのみ有効な方法である．

3. Direct Indices（DIs）

本節では，Direct Indices（DIs）の概念とそのアルゴリズムを提案する．

3.1 Direct Indices の概念と 3 つのセッション

P2P システムの分散インデックスでは，通常各ピアは半径 r 以内にあるピアの情報をインデックスとして保持する．ここでのピアの情報としては，そのピアが持つデータのメタデータ（データ名，データサイズ，作成時間等）や，そのピア自身の情報（ping の結果，IP アドレス等）がある．例えば，[6]ではインデックスを用いて以下の問い合わせ処理が行なわれる．ピア p が問い合わせを行う場合，まず p 自身のインデックスを使って問い合わせ処理を行う．ここで該当するデータがあればそのデータを結果として返す．また，返された結果が不十分な場合は，一定の規則に従って他のピア p_0 に問い合わせを転送する．p_0 はそのインデックスに対し問い合わせ処理を行う．これを繰り返す．転送規則の例として，問い合わせを行うピアと転送を打ち切るピアをホップ数で指定するものがある．例えば，問い合わせを行うピアは，問い合わせを発したピアから 1 および 5 ホップで到達できるピアであり，5 ホップを越えて問い合わせを転送しないといった規則が用いられる．P2P システムでは隣接ピアのネットワークへの参加・離脱などのタイミングでインデックスが更新されその更新が伝播する．この更新メッセージはピアの半径 r が大きければそれだけ増加することになる．

本論文では，各ピアは P2P ネットワーク上で下記の処理を通してインデックスやデータをやりとりするものとする．

- P2P システムに参加する join session
- 問い合わせを行う query session
- データを取得する transport session

各ピア p は隣接するピア p_0 から有用なピアの情報をピアの集合 rec(p_0,p)として受け取ることによってインデックスを取得・更新する．インデックス情報の受取りは join session および query session で行われる．有用なピアの情報はノードごとに有用度に基づいてランキングされて格納されており，ピア p が問い合わせを行う場合は，そのうちの上位 n 個のピアに対して問い合わせを行う．問い合わせは，結果に満足するまで繰り返す．このとき，問い合わせを行ったピアから結果とともにピアの有用度の情報を検索結果と合わせて取得しインデックスを更新する．インデックスが更新されるとその情報が隣接ノードに伝播する．

3.2 ピアの有用度

各ピアを評価するためにピアに有用度を設定する．本研究において，各ピアの有用度はそのピア保有のデータの有用度及び保有データ数によって評価されるものとした．各データの有用度は各データの需要によって決定する．各データの需要はその参照回数によって判断する．しかしながら，各データの需要は時間の経過とともに変化する．そのため参照回数が比較的多いデータであっても，現在はあまり需要がない可能性がある．よって，データ公開からの時間経過とともにそのデータの有用度を下げる．これにより，新規性は高いがあまり参照されていないデータの有用度の有用度を高く設定することが可能となる．また，各ピアの保有データ数も考慮する必要があると考えられる．

以上を踏まえ，ピアの有用度は以下のように定義される．まず，各ピアは，以下の式で定義されるそのピア固有の有用度 $E_{local}(p)$ を持つ．

$$E_{local}(p) \equiv \sum_{d \in D(p)} \frac{d_{ref} + 1}{|D(p)| \cdot \log(t_c - d_m + e)}$$

ここで，D(p)は p が保有するデータの集合であり，d_{ref} はデータ d の参照回数，d_m は d を最後に更新した時刻である．また，t_c は現在の時刻である．

次に，各ピアからその隣接ピアへ渡される有用度について考える．本研究では，ここでの有用度を先ほどの $E_{local}(p)$ をそのまま用いなかった．その理由は以下の通りである．ネットワーク上の有用度が比較的高い特定のピアに対して問い合わせのメッセージの集中が起こる危険性がある．その危険性を回避するためにメッセージを程良く分散することが必要となる．またネットワーク上の各ピアにおいて，そのピアの隣接ピアを考慮する必要性がある．あるピア p の各隣接ピアは比較的有用であれば，p はそのネットワーク上の要所であると判断されるためである．そのため各ピアの隣接ピアを考慮した有用度を設定する．

以上を考慮し，ピア p_1 からピア p_2 へ渡される p_1 の評価値 $E_{global}(p_1, p_2)$ を以下のように定めた．

$$E_{global}(p_1, p_2) = E_{local}(p_1) + \sum_{p \in Nei(p_1)} \frac{E_{global}(p, p_1)}{F}$$

ここで,Nei(p_1)は p_1 の join session で隣接するピアで p_2 以外のピア p の集合を表している．また，F は各ピアの隣接ピア数の最大値である．

これらの定義によって，ピアの評価はそのピアを格納してある DIs によって異なる値となり，問い合わせの分散がはかられる．この技法は PP2P システムにおいてのみ可能なピアの評価方法であり，HP2P システムでは困難である．それは，HP2P システムでは各ピアの情報は super peer のみによって維持されるため，各ピアの評価はそのシステム内において一意に決定されてしまうためである．

3.3 join session

join sessionは主にピア間の更新メッセージの伝播に用いられる．更新は以下のように行われる．まず，更新が必要な状況になると，ピアp はE$_{global}$(p,p_i)を計算する．ここでp_iはpの隣接ピアである．この値とDIs に格納されている各ピアの有用度の内で上位m個をpが推奨するピア集合rec(p,p_i)とする．rec(p,p_i)とpのP2Pシステム上での識別子p_{uid}を一組としたものを更新メッセージupdate(p_{uid},rec(p,p_i)) とする．このメッセージを受け取ったピアp_i はp_iのDIsを更新する．さらにp_iは更新が必要になれば同様に更新の手続きを行う．

更新が必要かどうかの判定はその更新状況による．たとえば，データが追加されたが推奨するピアには影響がない場合は更新メッセージを送信しない．

次に，ピア間の連結（join）について述べる．joinは更新の一種であると考える．たとえば，p_1とp_2のjoinではそれぞれ相手ピアへの更新メッセージを作成し，互いにその更新メッセージの交換に成功すれば，joinに成功したとする．そして，p_1とp_2は各DIsを更新し，必要があれば更新メッセージをそれぞれのjoin sessionの隣接ピアへ送信する．

ピア間の切断(leave)はjoinと正反対の動作である．leaveもjoinと同様に更新の一種であると考える．それぞれのピア間でleaveを行う場合，それぞれのピアのDIsからleaveする方向にあるピア情報を削除すれば十分である．

表1 実験に用いる各定数

定数	値
ピア数	10000
共有データの種類	10
共有データのオリジナル数	300
結果の満足数	20

表2 実験に用いる各変数

変数	説明
init_contents(p)	pの初期共有データ
F	ファンアウト数
$recommend_{join}$	join sessionでの推奨ピア数
$recommend_{query}$	query sessionでの推奨ピア数
$number_{query}$	発行する問い合わせ数

3.4 query session

query sessionは主に問い合わせに用いられるセッションである．問い合わせの手続きは以下の通りである．問い合わせが発行されると，まずピア内にあるデータに対しての問い合わせ処理を行う．この問い合わせで十分な検索結果が得られれば処理を終了する．不十分な場合は問い合わせの転送を行う．このとき問い合わせを発行するピアのDIsに格納されたピアで有用度が上位n以内にあるピアに対して問い合わせが行われる．問い合わせを受けたピアは，問い合わせ処理を行い，その結果とともにそのピアの推奨ピア集合も添付して送信する．問い合わせ結果を受け取ったピアは十分なデータが得られれば処理を終了する．不十分な場合は，問い合わせ結果とともに受け取った推奨ピア集合とDIsに存在するピア集合との和集合から問い合わせ送信済みのピア集合を除いたピア集合に対して有用度の上位n個のピアに問い合わせを発行する．この操作を十分な結果が得られるまで続ける．

4. 実験

ここでは実験を行うための環境，実験に用いた変数や定数の説明，及び実験の結果について述べる．

4.1 準備

表1と表2はそれぞれ実験に用いた各定数及び各変数を表している．また，表3は実験の計測項目を表している．ここでは，各ピアpのinit_contents(p)の数は0から100までの値とした．また各実験において，ある問い合わせqが満足されるとは，qの検索結果が一定数以上返されたこととする．

実験を行うに当たり，P2Pシステムでの問い合わせ方法として次の3つを用意した．

- Direct Indicesを用いる
- Iterative Deepeningを用いる
- indexを用いない

最初の方法は本研究の分散インデックスを用いた問い合わせ法である．最後の方法はGunutella[3]に用いられているものである．この方法では問い合わせの転送先は全隣接ピアであり，転送された問い合わせはそのピアで処理するとともに転送も行う．2番目の方法も問い合わせ先は全隣接ピアとなるが，それ以上の転送は行わずその問い合わせを凍結させる．問い合わせの発行ピアが結果に満足すればこの処理は終了する．満足しなければ，同様の問い合わせを再度転送する．再度問い合わせのメッセージを受け取ったピアは，問い合わせの処理は行わず，先ほど凍結した問い合わせを解凍し，単に隣接ピアへ問い合わせを転送する．以上を繰り返すことで問い合わせを満足しようとする方法である．[6]

表3 計測項目

変数説明	説明
$time_{forward}$	問い合わせ転送回数
$number_{update}$	更新メッセージ数
$number_{forward}$	問い合わせの転送回数

4.2 実験結果

まず，Fを3から8として各問い合わせ法について問い合わせの転送回数を計測した結果が図1である．この値は1ピアあたりの問い合わせ転送メッセージ数を示している．インデックスを用いない場合，一旦問い合わせを発行すればその問い合わせを凍結することができないため，問い合わせの転送回数は莫大な量になった．Iterative Deepening を用いた場合，問い合わせが満足すれば問い合わせを凍結することが可能であるため問い合わせ転送回数は減少した．DIsを用いた場合は問い合わせ対象のピアを限定するため，Iterative Deepeningを用いた場合よりも問い合わせメッセージ数を減少させることが可能になった．

次に，Fを上記と同様に変化させ，DIsを用いて場合の分散インデックス更新のメッセージ数を計測した．その結果が図2である．ここで示してある結果は1ピアあたりのjoinとleave以外の更新メッセージ数である．図1のDIsの値に図2の値を加えてもネットワーク全体のメッセージ数は減少する結果となった．DIs を更新したピアが更新メッセージを送信する可能性がある場合，まず送信する前に送信先のピアに対する前回の送信情報を比較する．比較の結果，あまり差異がない場合は更新メッセージを送信しない．このことより，不必要と思われる更新メッセージの送信を未然に防ぐことができている．

各問い合わせ法において問い合わせ結果が満足されるまで問い合わせを繰り返した結果を図3に示す．図3の結果は問い合わせを満足するまでの時間と同等である．インデックスを用いない場合は上記の手続きが行えないため，結果を満足したときの結果を返してきたピアのホップ数の最大値を計測した．ここでは，DIsは他の方法よりも問い合わせを繰り返す回数が圧倒的に少ない．これは，DIsに格納されている情報が問い合わせ先の選択に非常に有効であることを示している．

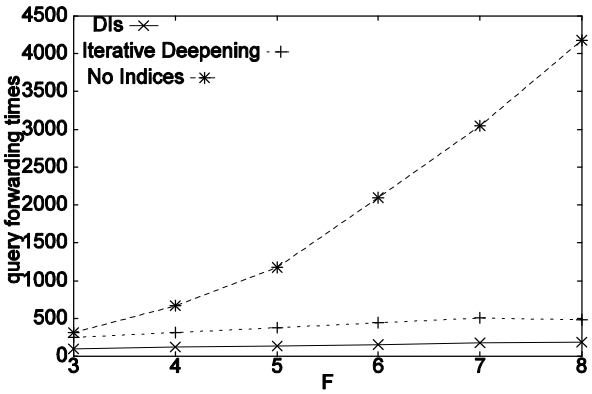

表1 問い合わせの転送回数

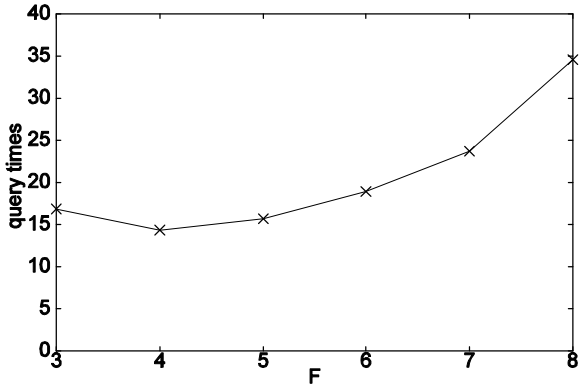

表2 更新メッセージ数

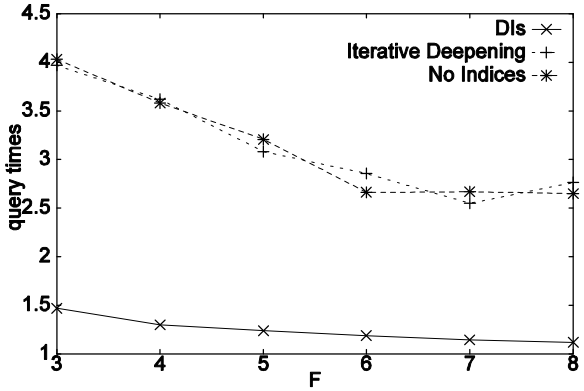

表3 問い合わせの発行回数

5. おわりに

現在，P2Pシステムは一般的に認知されはじめ，データ共有システムをはじめグループウェア[13]等，多様な形式で用いられてはじめている．しかしながら，P2Pシステムでは効率的な問い合わせと帯域幅消費の軽減の2つの問題がある．そこで本研究ではこの問題を解決するために分散インデックスであるDirect Indices (DIs) を導入した．また，DIsを用いた場合とインデックスを用いない場合での各種実験を行った．その結果，DIsを用いた場合ではインデックスを用いない場合に比べ，問い合わせの転送メッセージ数を軽減し，規定以上の問い合わせ結果を返す時間も改善された．これにより，P2Pシステム全体のパフォーマンス改善されることを確認した．

[文献]

[1] Napster, “http://www.napster.com/”.
[2] OpenNap: Open Source Napster Server, “http://opennap.sourceforge.net/”.
[3] Gnutella website, “http://www.gnutella.com/”.
[4] FreenetProject website, “http://freenet.sourceforge.net/”.
[5] Arturo Crespo, Hector Garcia-Molina, “Routing Indices For peer-to-peer Systems”, Proc. The 22nd International Conference on Distributed Computing Systems (ICDCS), Vienna, AUSTRIA, July 2002.
[6] Beverly Yang, Hector Garcia-Molina, “Improving Search in peer-to-peer Networks”, Proc. The 22nd International Conference on Distributed Computing Systems (ICDCS), Vienna, AUSTRIA, July, 2002.
[7] S.Ratnasamy, P.Francis, M.Handley, R.Larp, S.Shenker, “A scalable contentaddressible network”, In ACM SIGCOMM, San Diego, USA, August 2001.
[8] J.Kubiatowicz, D.Bindel, Y.Chen, S.Czerwinski, P.Eaton, D.Geels, R.Gummadi, S.Rhea, H.Weatherspoon, W.Weimer, C.Wells, B.Zhao, ”OceanStore: An Architecture for Global-Scale Persistent Storage”, Proc. the 9th International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS), Cambridge, USA, November 2000.
[9] B.Zhao, J.Kubiatowicz, A.Joseph, “Tapestry: An Infrastructure for Faulttolerant Wide-area Location and Routing”, U. C. Berkeley Technical Report UCB//CSD-01-1141, April 2000.
[10] Y.Chen, R.Katz, John D. “Dynamic Replica Placement for Scalable Content Delivery”, Proc. 1nd Internal Workshop on Peer-to-Peer systems(IPTPS), Cambridge, USA, March 2002.
[11] I.Stoica, R.Rorris, D.Karger, M.F.Kaashoek, H.Balakrishnan, “Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications”, In ACM SIGCOMM, San Diego, USA, August 2001.
[12] A.Rowstron, P.Druschel, “Pastry: Scalable, distributed object location and Routing for large-scale peer-to-peer systems”, In Middleware, Heidelberg, Germany, November 2001.
[13] Groove Networks, Inc., Desktop Collaboration Software, “http://www.groove.net/”.

山田 太造 Taizo YAMADA
総合研究大学院大学数物科学研究科博士後期課程在学中．データベースシステム，分散処理，情報検索等に興味を持つ．情報処理学会，日本データベース学会各学生会員．

相原 健郎 Kenro AIHARA
国立情報学研究所ソフトウェア研究系助手．1997 年東京大学大学院工学系研究科博士課程修了．博士（工学）．人工知能，発想支援システムの研究に従事．特に人間の創造性を伸ばすことを目的とする知的活動支援の理論，方策に興味を持つ．人工知能学会，情報処理学会，日本認知科学会，ACM 各会員．

高須 淳宏 Atsuhiro TAKASU
国立情報学研究所/総合研究大学院大学助教授．1989 年東京大学大学院博士課程終了．博士（工学）．データベースシステム，機械学習の研究に従事．電子情報通信学会，情報処理学会，人工知能学会，ACM，IEEE 各会員．

安達 淳 Jun ADACHI
1981 年東京大学大学院工学系研究科博士課程修了．工学博士．東京大学大型計算機センター助手，文部省学術情報センター研究開発部助教授、教授を経て現在国立情報学研究所教授．東京大学大学院情報理工学研究科教授を併任．データベースシステム，分散処理システム，情報検索，電子図書館システム等の開発研究に従事．電子情報通信学会，情報処理学会、IEEE，ACM 各会員．