

## 非構造 Peer-to-Peer システム上でのピアの有用性に 基づいた問い合わせ処理

山田 太造<sup>†</sup> 相原 健郎<sup>††,†</sup>  
高須 淳宏<sup>††,†</sup> 安達 淳<sup>††</sup>

本研究では, peer-to-peer (P2P) システムでの問い合わせ処理向上のために, ピアに関する情報を格納するための新たな分散インデックスである *Direct Index (DI)* を導入する. DI を用いて問い合わせを行う場合, 問い合わせメッセージをピアの有用性に基づいてルーティングすることができるため, スケーラビリティの向上を実現可能にする. さらにより有用な文書を発見可能にするために, 文書の複製を行い他のピアへ配置する方法を導入する. 文書の複製によって, 問い合わせ処理の成功率の向上が期待される. また, 各種実験を行い, 提案した各方式が問い合わせ処理の向上することを示す.

### Query Processing Based on Usefulness of Peers in Unstructured Peer-to-Peer Systems

TAIZO YAMADA,<sup>†</sup> KENRO AIHARA,<sup>††,†</sup> ATSUHIRO TAKASU<sup>††,†</sup>  
and JUN ADACHI <sup>††</sup>

In this paper, to improve in query processing on a peer-to-peer (P2P) system, we introduce a new distributed index, named *Direct Index (DI)*, where a peer stores other peer information such as a usefulness of a peer in own DI. When a peer starts a query using a DI, because effective query message routing based on the usefulness of peers can be achieved, a DI enables to improve in scalability. To enable the discovery of a useful document, we also introduce replication methods based on dynamic peer grouping, which are expected of an improvement in query success rate. To ensure effects by proposal methods, we show experimental results on query processing.

#### 1. はじめに

現在, Peer-to-Peer (P2P) システムは各ユーザ間で直接データを交換することに適しているだけではなく, データの追加や更新の反映の機敏さや, システム自体の自律性に優れた分散コンピューティングシステムであり, Napster や Gnutella に代表されるデータ交換システムだけではなく, グループウェア<sup>5)</sup> やオンラインコミュニティ<sup>6)</sup> 等にてその効果を発揮している. Gnutella や Freenet に代表される Pure P2P (PP2P) システムを利用した協調作業のためのシステム, たとえば, 文献に関するアノテーションを配布するシステムや掲示板システム等を考える. このシステムでは各ユーザは関心のあるグループへ自由に参加でき,

文書の作成, 更新を自由に行うことができ, また自由にシステムに参加, 離脱できる環境が望まれるためである. しかしながら, 効率的なデータ共有を実現するためには問い合わせ処理の向上が求められる.

PP2P システムで用いられる “flooding” による問い合わせは, 各ピアからある一定の距離にある全ピアに対して問い合わせメッセージを伝播させてしまうため, そのメッセージ数はホップ数に対して指数関数的に増加してしまう. そのためこの仕組みはシステム全体の帯域幅消費量を増加させてしまう. また協調作業システムでは, 素早く検索結果を得られる仕組みが求められるため, 良好な応答時間も求められる.

本研究では, P2P システムでの問い合わせ処理向上のためにピアに関する情報を格納するための新たな分散インデックスである *Direct Index (DI)* を導入する. 各ピアはシステム上の有用なドキュメントを多く保有しているピアの情報をそのピアの DI 内に格納し, 各ピアが DI に基づいて問い合わせを行う場合, 有用なピアのみに直接問い合わせメッセージを送信するた

<sup>†</sup> 総合研究大学院大学情報学専攻

Department of Informatics, The Graduate University  
for Advanced Studies

<sup>††</sup> 国立情報学研究所

National Institute of Informatics

め、flooding による問い合わせ処理に対して、帯域幅消費量の低減及び応答時間の向上が可能である。問い合わせメッセージのルーティングはシステムのスケラビリティを向上させるが有用ではないピアには問い合わせメッセージは転送されないことが多いため、発見されない文書が存在してしまう。またピアがシステムから離脱することによって有用な文書が“喪失”してしまう。そのためより有用な文書の発見と有用な文書の喪失防止のために、文書複製の方法を導入する。14), 18) の文書複製の方法は、有用な文書を優先して複製することで flooding による問い合わせ処理を向上させる。そこで DI を用いた場合の問題を解消し、問い合わせ処理の成功率を向上させるためにこの文書複製の方法の導入する。

本研究では以降、3 節で、各ピアの有用度の決定方法及び DI の構造、管理及び DI を用いた問い合わせ方法を示す。4 節では、文書の複製方法及び更新方法を示す。また、本研究での分散インデックスの効果を示すため、5 節で提案手法と既存の分散インデックスを比較し、さらにデータ配置による効果も示す。

## 2. 関連研究

P2P システムの問い合わせ処理性能を向上させる手法を以下のように分類することができる。

第 1 の方法はメッセージのルーティングである。Yang 等<sup>15)</sup> は super peer を用いる Hybrid P2P (HP2P) システムにおいて super peer が保有するインデックスを用いた問い合わせ処理の影響を分析している。Pure P2P システムでは様々なアプローチがある。Yang 等<sup>16)</sup> はインデックスを用いた方法としてローカルデータに関するインデックスを伝播させる方法について提案し、検証を行っている。Freene では、各ピアはピア内にキャッシュされたデータを基にインデックスを構築する。しかしながらこの検索方法ではキャッシュに存在しないデータに関して効率的に問い合わせを行えない。Adaptive Probabilistic Search (APS)<sup>12)</sup> は帯域幅消費量を低減することに特化した分散インデキシングの方法である。各ピア上で処理されたオブジェクト毎にそのピアのインデックスにレコードを追加するため、インデックスのサイズは肥大化してしまう傾向がある。コンテンツベースの分散インデックスでは、文書を分類化し、各トピックに属する文書数をそのインデックスに格納する。RI は、問い合わせメッセージの効率的なルーティングを可能にする。しかし、RI はインデックスの更新頻度が高いため、ピアが頻繁にシステムへの出入りを繰り返すと帯域幅消費量が多くなってしまいう欠点がある。他方、分散ハッシュ表 (Distributed Hash Tables; DHTs) を用いた問い合わせ処理も多く研究されている。例えば、CAN<sup>8)</sup>、Chord<sup>10)</sup>、Oceanstore<sup>7)</sup>、Pastry<sup>9)</sup>、Tapestry<sup>17)</sup> 及

び P-Grid<sup>1)</sup> 等が挙げられる。これらのシステムでは特別なネットワーク構造を用いることで効率の良い問い合わせ処理が実現されるが、その効果を持続するためにはピアの出入りによってネットワーク構造を再構築する必要があるため、ピアの join/leave は非常に高コストになってしまう。

問い合わせ処理性能を向上させる第 2 の方法としてデータ複製がある。Sun 等<sup>11)</sup> は HP2P システム上にて Full-Replication (複製をピアグループに属する全ピアへ配置) 及び Partial-Replication (ピアグループの部分集合に複製を配置) に関して考察している。DHTs を基にしたシステムでは、Chen 等<sup>2)</sup> は Tapestry ネットワーク上での、Datta 等は<sup>4)</sup> P-Grid システム上での複製配置方法及びデータ更新の方法を研究している。

## 3. 問い合わせメッセージのルーティング

一定数以上の文書を限られた時間内にユーザに返す問い合わせ処理を行う場合、効率的な問い合わせメッセージのルーティングは不可欠である。有用なピアは多くの有用な文書を保有していると仮定した場合、有用なピアを検知し、そのピアに対して問い合わせメッセージを優先的に送信する方法が考えられる。そこで各ピアに有用度を設け、各ピアの有用度を格納するための新たな分散インデックスである *Direct Index (DI)* を導入した<sup>13)</sup>。各ピアは DI を用いることで有用度の高いピアの情報を得ることが可能になり、それらのピアに対して直接問い合わせメッセージを送信することが可能になる。

問い合わせ処理では、トピックに応じてピアの有用度が異なることも多々ある。そこで我々は *Normal DI (NDI)* 及び *Topic-based DI (TDI)* の 2 つの DI を導入する。NDI を用いた場合、各ピアはトピックに関係なくピアの有用度を格納する。一方、TDI を用いた場合、各ピアはピアの有用度に加えて、それらのピアが保有している各文書のトピックの有用度も格納する。本節ではピアの有用度の決定方法について示し、NDI 及び TDI を用いた問い合わせ処理、管理方法について述べる。

### 3.1 ピアの有用度

有用なピアを検知するためには、**各ピアの有用度**を設定する必要がある。ピアの有用度はそのピアが保有している文書によって決定される。そこで**文書の有用度**を決定する必要がある。各文書のアクセス頻度によってその文書の需要は判断することができる。しかしながら、その文書へのアクセスは最近行われていないのであれば、あまり有用ではない。また、継続的に参照される文書は重要であると判断できる。そこで、各文書の有用度を、ある時間間隔でのアクセス頻度によってではなく、そのアクセス頻度とその文書の更新からの経過時間によって決定し、これを以下の (1) 式

| Peer "A" |            |          |
|----------|------------|----------|
| UID      | usefulness | location |
| A        | 6          | local    |
| I        | 60         | D        |
| J        | 45         | D        |
| F        | 21         | B        |
| B        | 18         | neighbor |
| C        | 12         | neighbor |
| Q        | 6          | C        |

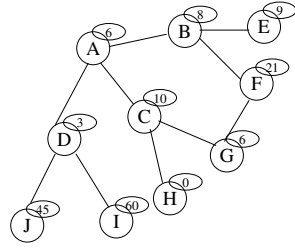


図 1 NDI の例

で表す.

$$E_{doc}(d) \equiv \frac{d_{ref} + 1}{\log(d_{et}) + 1} \quad (1)$$

ここで  $d_{ref}$  を  $d$  の参照回数,  $d_{et}$  を  $d$  を公開時刻もしくは最終更新時刻からの経過時間とする. 次にピア  $p$  の有用度は  $p$  が保有する文書の有用度の総和によって決定することにした. これを (2) 式に示す.

$$E_{peer}(p) \equiv \sum_{d \in D(p)} E_{doc}(d) \quad (2)$$

ここで,  $D(p)$  はピア  $p$  の保有する文書の集合とする.  $E_{peer}(p)$  をそのまま用いた場合, ネットワーク上の有用度が高い特定のピアに対して問い合わせメッセージの集中が起こる危険性がある. そこで, ネットワークでの位置に依存した有用度を導入し, ピア  $p_1$  からピア  $p_2$  へ渡される  $p_1$  の有用度を以下のように定めた.

$$ER_{p_1}(p_1, p_2) = E_{peer}(p_1) + \sum_{p \in \mathcal{P}(p_1) - \{p_2\}} \frac{ER_{p_1}(p, p_1)}{F} \quad (3)$$

ここで  $\mathcal{P}(p_1)$  は  $p_1$  の隣接ピア集合,  $F$  は各ピアの隣接ピア数の最大値 (ファンアウト数) である. この式によって  $p_1$  と  $p_2$  間の距離が遠くなるほど,  $A$  に渡された  $B$  の有用度は低く設定される.  $ER_{p_1}(p, p_1)$  は  $p_i$  の DI 内に格納された  $p$  の有用度であり, DI の更新手続きに従って適宜更新されていくが<sup>13)</sup>, 更新手続きの過程でタイムラグが生じることもある.

### 3.2 Normal DI

NDI を用いる場合, 各ピアはそのピアの DI 内に UID (ピアの識別子), value (ピアの有用度) 及び location (そのピアが位置する方向) をピアの情報として格納する. ここでピア  $A$  及び  $C$  がピア  $B$  の隣接ピアであり,  $A$  と  $C$  は隣接していないが  $A$  は  $C$  の情報を  $B$  によって知っている場合,  $A$  の DI 内の  $B$  の location は “neighbor”,  $C$  の location としてそのピアの情報を送信したピアの UID である  $B$  として格納する. 図 1 はピア  $A$  の DI の例である.

次に, ピア間の join について述べる. たとえば,  $p_1$  と  $p_2$  が join する場合, 互いに更新メッセージを作成し, 交換することで join が行われる. 各ピアは定期的に隣接ピアに ping メッセージを送信し, 隣接ピア

からの pong メッセージを受け取ることによって隣接ピアの状況を確認する. このとき, pong メッセージを受信できなければ, その隣接ピアがシステムから離脱したと判断し, その隣接ピアから得たピア情報を削除することで leave 処理を行う. join/leave によってインデックスの更新が行われるため, インデックス更新が引き続き行われる.

図 1 において,  $A$  はインデックス更新が必要になり, 更新メッセージを  $B$  に送信するとき,  $ER_A(A, B)$  を計算する. この値と DI に格納されている各ピアの有用度の内で上位  $m$  個を  $A$  が推奨するピア集合  $rec(A, B)$  とする.  $rec(A, B)$  と  $A$  の P2P システム上での識別子を一組としたものを更新メッセージ  $update(A_{uid}, rec(A, B))$  とする. このメッセージを受け取った  $B$  はその DI を更新する. さらに  $B$  は更新が必要になれば同様に更新の手続きを行う.  $B \rightarrow F \rightarrow G \rightarrow C \rightarrow A$  と更新メッセージが伝播された場合, この更新の起源である  $A$  に更新メッセージが送信されてしまい, インデックス更新が無限に続く危険性がある. そこで各インデックス更新メッセージに識別子を設ける. 各ピアは更新メッセージを受け取ったとき, その更新メッセージの識別子によってその更新メッセージを以前受け取ったかどうかを判断することができる. もし更新メッセージを以前受け取っているならばその更新を破棄し, 更新処理を終了する.

ピア  $A$  が問い合わせを行う場合,  $A$  内にある文書を対象とした問い合わせ処理だけでは十分な検索結果が得られないのであれば, 問い合わせメッセージの転送を行う. このとき  $A$  の DI 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メッセージを送信する. 問い合わせメッセージ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送信する.  $A$  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 DI 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メッセージを送信する. この操作を十分な結果が得られるまで続ける. この問い合わせ処理において, 有用ではないピアにはメッセージは送信しない. また, 各ピアは問い合わせを行う前には既知ではなかったピアに対しても問い合わせを行うことが可能になり, 幅広くシステムに問い合わせメッセージを送信するが帯域幅消費量を抑えることも可能にする.

### 3.3 Topic-Based DI

NDI を用いた場合, 各ピアは有用な文書を得ることができるが, NDI には各文書の内容に関する情報が格納されていないため, 問い合わせ内容に応じたピアの選択ができない. そこで文書のトピックに関する情報も含んだ *Topic-based DI (TDI)* を導入する. TDI は, 各ピアの UID, value 及び location 以外に **トピッ**

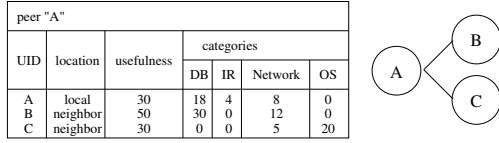


図 2 TDI の例

クに対する有用度も格納する。TDI では、ピア  $p$  のトピック  $t$  に対する有用度を (4) 式のように  $p$  が保有するトピック  $t$  の文書の有用度の総和で表す。

$$E_{topic}(p, t) \equiv \sum_{d \in D(p, t)} E_{doc}(d) \quad (4)$$

ここで  $D(p, t)$  は  $p$  が保有している、トピック  $t$  の文書の集合である。図 2 は TDI の例である。各ピアは、一定の範囲のトピックの文書を所有するため、TDI においてトピックに対する有用度の情報はあまり大きくならない。そのため、NDI と比較すると TDI のサイズは大きくなるが、際だって大きくなることはない。

トピックに対する有用度の変化があれば、各ピアはインデックス更新メッセージをその隣接ピアへ送信するため、TDI を用いた場合、NDI を用いた場合よりも各ピアのインデックスの更新頻度が高くなる。

TDI を用いた問い合わせ処理ではピア  $p$  はその DI 内の各ピア  $p_i$  をランク付けするとき、 $E_{topic}(p_i, t)$  を用いる。 $p_i$  が問い合わせ  $q$  を行う場合、DI 内の  $q$  と関連するトピックの有用度を用いてランク付けする。 $p_i$  の隣接ピア  $p$  へ問い合わせメッセージを送信した場合、 $p$  はその検索結果とともにその問い合わせに関連するトピックの有用度に応じて推奨ピアを決定する。このとき、 $p$  の DI 内に格納されている各ピア  $p_j$  に対して  $E_{topic}(p_j, t)$  を計算し、上位  $m$  個のピアを推奨ピアとする。

図 3 は TDI を用いた問い合わせ処理の例である。ピア A は問い合わせ “DB” を行う場合、A のローカルで問い合わせが終了しないならば A はその DI 内の各ピア  $p_i$  に対して  $E_{topic}(p_i, “DB”)$  ( $p_i \in \{B, C\}$ ) を計算し、問い合わせメッセージの送信先を決定する。このとき  $E_{topic}(B, “DB”) = 18$ ,  $E_{topic}(C, “DB”) = 0$  であるため A は B に問い合わせメッセージを送信する。次に B はローカルに問い合わせを行い、 $E_{topic}(p_j, “DB”)$  ( $p_j \in \{D, E\}$ ) を計算する。このとき  $E_B(D, B) = 40$ ,  $E_B(E, B) = 20$  であり、 $E_{topic}(D, “DB”) = 8$ ,  $E_{topic}(E, “DB”) = 16$  である。B は推奨ピア集合  $\{< D, 8 >, < E, 16 >\}$  と検索結果を A に送信する。このとき D の有用度は E の有用度よりも高いが、A は D よりも E をより有用であると判断する。各ピアは TDI を用いることによって、問い合わせメッセージを送信先を修正することが可能になり、NDIs と比較し問い合わせ処理性能の向上が期待できる。

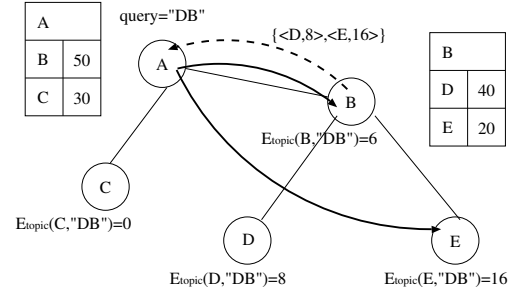


図 3 TDIs を利用した問い合わせ処理

#### 4. 文書の複製と配置

DI を用いることによって問い合わせ処理は向上するが、各ピアの leave によってシステムから文書が“喪失”してしまう。また、DI を用いた場合、各ピアは問い合わせメッセージを有用なピアのみに送信するため、有用ではないピアには問い合わせメッセージが伝播されない可能性が高い。DI の仕組みにより、有用ではないピアが保有している文書は各ピアの問い合わせ結果に含まれにくいため、文書が“埋没”してしまう。そこで文書の複製を行い、複製を他のピアに配置することでこの問題を解決する。文書の複製の方法は 14), 18) の方法を用いる。この複製方法ではピアの有用度に応じて各ピアの複製を生成する個数を決定するとともに、文書の有用度によって複製対象文書の選択を行う。そして、動的にピアグループを形成し、そのピアグループからピアを選択し、複製を配置する。

##### 4.1 複製対象文書の選択

文書を複製する場合、その文書が各ユーザにとってあまり有用ではない場合、その文書の複製はあまり効率的であるとはいえない。そこで、各文書の有用性を基に複製対象となる文書を選択し、その文書の複製数を決定する。PP2P システムでは、サーバが存在しないため、システム全体の文書の情報を集約することは困難である。また、各文書の有用度は参照回数や時間の経過により非常に変動しやすい。そのため、文書の有用度で文書の複製数を決定するためには文書の有用度の情報を頻繁に交換する必要があるため、システムのパフォーマンス低下させる要因になる可能性がある。そこで各文書の有用度を用いてシステム上のピアに有用度を設け、ピア  $p$  とその各隣接ピアの有用度によって各ピアの複製数を相対的に決定する。 $p$  とその隣接ピアの集合を  $\mathcal{P}$ 、1 ピアあたり  $c$  個の複製を生成する場合、 $p$  とその隣接ピア全体での複製の合計数は  $c \cdot |\mathcal{P}|$  となる。このとき  $p$  の複製数  $P_{peer}(p)$  は次の式にて決定する。

$$P_{peer}(p) = c \cdot |\mathcal{P}| \cdot \frac{E_{peer}(p)}{\sum_{p_i \in \mathcal{P}} E_{peer}(p_i)} \quad (5)$$

表 1 実験で用いたパラメータ

| 定数                 |          | 値      |
|--------------------|----------|--------|
| 文書数                |          | 3,000  |
| ファンアウト             |          | 4      |
| 文書の最大アップロード数       |          | 4      |
| 文書の最大ダウンロード数       |          | 4      |
| join の確率           |          | 0.18   |
| leave の確率          |          | 0.014  |
| ネットワークからの離脱の確率     |          | 0.006  |
| 問い合わせの確率           |          | 0.09   |
| 文書修正の確率            |          | 0.008  |
| 文書作成の確率            |          | 0.002  |
| 問い合わせの終了条件         |          | 20     |
| Time to live (TTL) |          | 8      |
| 推奨ピア数              |          | 4      |
| 最大複製保有数            |          | 20     |
| 1 ピアあたりの平均複製数      |          | 8      |
| 変数                 | 説明       | 値      |
| <i>num_topic</i>   | 文書のトピック数 | 20     |
| <i>num_peer</i>    | ピア数      | 10,000 |

ピア  $p$  は以下の手続きで複製対象となる文書を選択し、その複製を他のピアへ配置する。

- (1)  $p$  が保有している各文書  $d$ 、 $d$  の既に存在する複製の個数を  $d_{rep}$  とする。 $d_{rep}$  の合計が  $P_{peer}(p)$  以上であればこの手続きを終了する。
- (2)  $d$  の有用度  $E_{doc}(d)$  を計算する。
- (3)  $E_{doc}(d)/(d_{rep} + 1)$  を計算し、この値に応じて所有する文書を降順に並べる。
- (4) 最もランクの高い文書を選択し、その文書の複製を配置する。配置の方法は 4.2 節にて述べる。
- (5) 1 に戻る。

#### 4.2 文書の配置方法

文書を配置する場合、動的にピアグループを形成し、そのグループに属するピアへ複製を配置する。ここでは、ピアグループの形成方法及び複製配置先の決定方法を以下に示す。

- PN (Placement on Neighbors) 方式: ピアグループとして各ピアの隣接ピアを選択し、最も有用度の低いピアへ複製を配置する。この方法では、複製文書を配置するピアに 1 ホップで到達できるため、文書を配置しているピアの状態をすばやく把握することが可能である。
- PIR (Placement on Inverted References) 方式: P2P システムでは、ある文書を取得するときに直接ピア間で文書を転送する。ここで、ある文書  $d$  へ参照を行う場合、参照を行ったピアは  $d$  に対して関心があり、 $d$  を保有するピアと  $d$  を参照したピアの間には関連性があると考えられる。あるピア

ピア  $p_1$  が保有する任意の文書が他のピア  $p_2$  より参照されたことがあれば、この方式では、 $p_1$  は  $p_2$  を  $p_1$  と同じピアグループと見なす。複製配置先は参照を行った最後のピアから選択し、そのピアへ複製を配置しようとする。

- PRP (Placement on Related Peers) 方式: 各ピアはそのピアと類似の問い合わせを発行するピアを同じピアグループと見なす。また配置先はグループに属する各ピアの最近の問い合わせキーワード  $qt$  件を基に問い合わせキーワード  $k$  の頻度によって決定する。たとえば、ピア  $p_1$  と  $p_2$  はともに “DB” というキーワードでの問い合わせを行うことが多い場合、 $p_1$  と  $p_2$  の関連性は高く、 $p_1$  が複製を配置する場合、 $p_2$  は複製配置の対象となる。文書配置を行うピアはそのピアが既知であるピアから配置対象となるピアを選択するが、この文書配置方式を用いる場合、各ピアは他のピア情報を得るときに、他のピアの存在以外にも問い合わせの状況情報も得る必要がある。
- ランダム方式: 各ピアは、そのピアが既知である任意のピアを同じピアグループに属すると見なし、そのピアグループに属するピアへランダムに複製を配置する。

ランダム方式以外の各文書配置方式において文書配置対象となるピアを見つけ出すことができない場合がある。その場合、ランダム方式を用いて文書を配置する。

## 5. 実 験

### 5.1 実験の概要

ここでは NDI 及び TDI の効果を明らかにするために、各ピアの文書の情報を格納する Routing Index(RI)<sup>3)</sup> を用いた検索方法及び flooding による検索方法 (Gnutella) と比較した。また、文書複製の配置の効果を確かめるため、4 節で与えた各複製配置方法及びデータ配置を行わない No-Replication の 5 つの方式を TDI を用いて比較した。計測項目は以下の通りである。応答時間を評価するために、一定以上の検索結果が得られるのに必要とされたホップ数を用いた。帯域幅消費量を評価するために、問い合わせメッセージ数及びインデックス更新メッセージ数を用いた。また、問い合わせの成功率、つまり、各問い合わせが一定以上の検索結果を取得する確率も計測した。

本実験では、ピアの操作をシステムへの参加状況の変化 (join / leave)、問い合わせ処理、データ取得・データ配置、データの更新・作成とし、各ピアは、1 単位時間に上記のいずれかの操作を一定の確率で行うものとした。表 1 は実験で用いた各定数及び各変数を示している。また、この表に示されている各変数の値は、その変数を固定した場合の値である。各ピアはシステムへ再び join を行うとき、最後にシステムへ参加

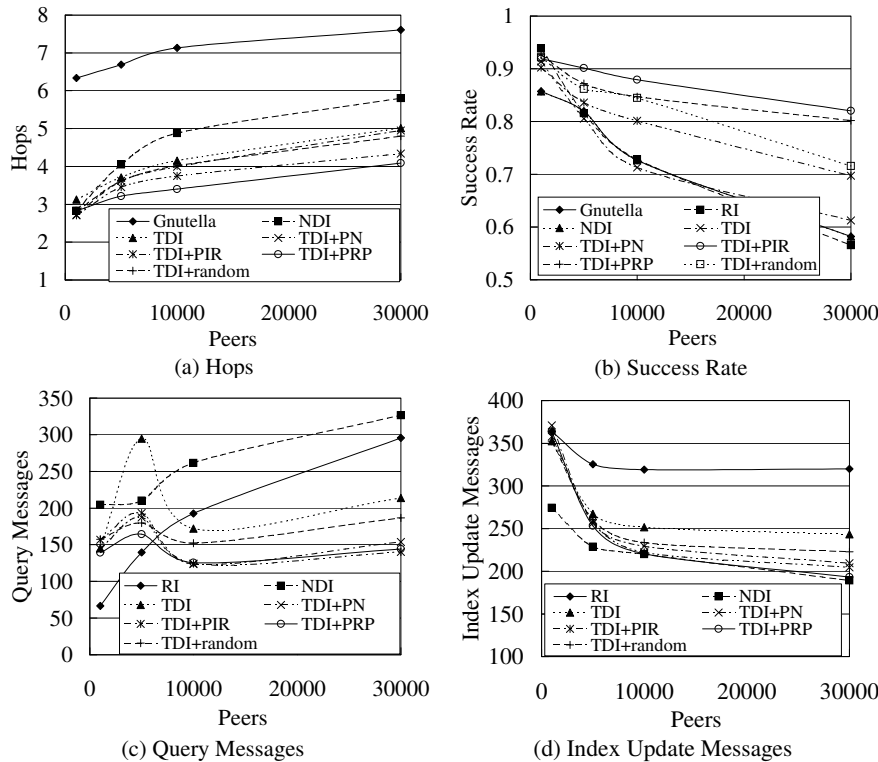


図 4 ピア数に対する問い合わせ処理性能

していたときの隣接ピア、これらのピアに対して join できない場合は類似の問い合わせメッセージを出していたピア、その他のピアの順序で join を行おうとする。また leave を行う場合、各ピアは過去の問い合わせにおいて平均検索結果数が少なかったピアに対して leave する。また各ピアには嗜好するトピックを持っており、問い合わせを行う場合、そのトピックに関する問い合わせを行う確率は 0.6 とし、データを生成する場合は常にそのトピックに関するデータを生成するものとした。

初期データ分布として偏在のあるデータ分布を用いた。実際にはシステム全体の 80% のデータをシステム上の 20% のピアが保有するデータ分布を用いた。この実験では全データのサイズは同じであると仮定している。また、用いたネットワークポロジはサイクルの存在する木構造とし、検索モデルとしてブーリアンモデルを用いた。P2P システムの問い合わせ処理では、同一のデータを異なるピアから受け取ることがある。本研究では同一のデータを複数個受け取った場合、1 つの結果を得たものとした。

## 5.2 実験結果

図 4 はピア数を変動させたときの問い合わせ処理の結果を示す。ピア数を増加させた場合、初期の文書数は増加させないためピアの増加に伴い、各手法のパフォーマンスは低下する。NDI 及び TDI はピア数が増

加しても、一定以上の検索結果を得るのに必要なホップ数を flooding による方法の約半分程度に低減することが可能であった (図 4 (a))。NDI 及び TDI を用いた場合、各ピアは多くの有用な文書を保有しているピアに対して直接問い合わせメッセージを送信することが可能であるため、ピアは flooding による問い合わせ処理に比べ少ないホップ数で一定以上の検索結果を得ることが可能である。またこれらの DI を用いた場合、遠くに位置する有用なピアに対しても問い合わせメッセージを送信することが可能であるため、スケラブルに問い合わせを行うことができる。ピア数が 1,000 であるとき NDI と TDI のパフォーマンスにあまり差はないが、ピア数の増加に伴い、TDI は NDI よりも優れたパフォーマンスを示す。これは TDI は問い合わせメッセージの送信先をトピックに対する有用度を用いて修正することが可能であるためである。

文書複製の方法を TDI と組み合わせた効果もこの図に示している。各文書複製方法も単に TDI を用いた場合に対してそのパフォーマンスが優れていることが分かった。特に TDI+PRP の手法が他の手法を組み合わせた場合よりも優れていた。この状況では PRP 方式は各ピアのトピックに対する有用度を他の配置方式よりも高くするため、PRP 方式と組み合わせた場合、TDI による問い合わせメッセージの送信先の修正がより効果的であると予想できる。同様の理由で図 4

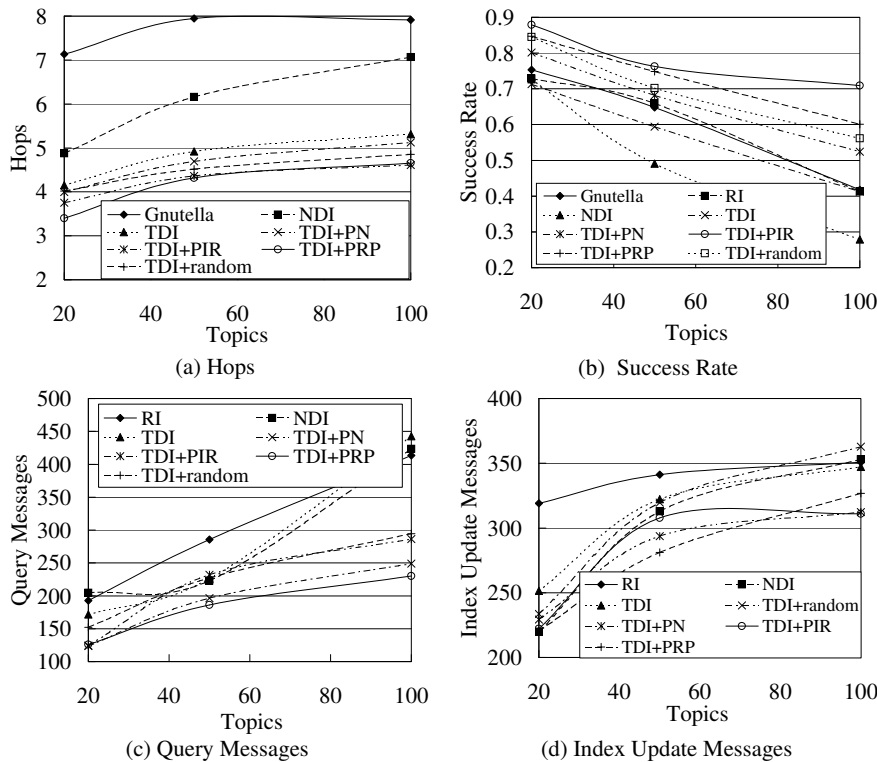


図5 トピック数に対する問い合わせ処理性能

で示される結果にて TDI を用いた場合及び TDI と各複製配置方式を組み合わせた場合のパフォーマンスが優れていると考えられる。

図 4 (b) はピア数の増加に対する問い合わせの成功率を示す。NDI 及び TDI はピア数の増加に伴いそのパフォーマンスは低下し、RI や flooding を用いた場合とあまり変わらないパフォーマンスを示す。しかしながら TDI と複製配置を組み合わせた場合、大幅なパフォーマンス改善を示す結果を得た。特に、PRP 及び PIR の各方式との組み合わせは他の手法よりも優れていた。この理由として、PRP 及び PIR 方式を用いた場合では、ピアが嗜好するトピックと関連する文書の複製がそのピアに配置されたためである。また複製配置によって、文書の“喪失”や“埋没”に対応可能であることも示している。本研究では同一の検索結果を複数得た場合、1 つの結果を得たに過ぎないと判断するため、この結果より、より多くの検索結果を得ることが可能であったことを示す。

図 4 (d) はインデックス更新のメッセージ数を示す。NDI を用いた場合、最も更新メッセージ数が少なかった。それは RI や TDI と比較した場合、NDI はインデックスの更新に対してあまり敏感ではないためである。対照的に、RI は NDI に対しインデックスの更新に敏感であることが分かる。TDI は各トピックの有用度の変化に対して敏感であるが、有用ではない

ピアのインデックス更新情報の伝播をあまり行わないため、RI と NDI の中間的な更新頻度であった。

次にシステムのトピック数を変化させた場合の結果を図 5 に示す。トピック数を増加させた場合、各トピックあたりの文書数は減少するため、問い合わせ処理のパフォーマンスは低下してしまう。NDI を用いた場合、この影響を多大に受けてしまう。有用なピアの情報を格納する NDI であるが、トピック数が増加すると、各ピアがそのピアの DI 格納している有用なピアが保有している文書に対して嗜好する確率が大幅に低下してしまう。そのため、一定以上の検索結果を得るまでのホップ数及び問い合わせメッセージ数、問い合わせの成功率は低下してしまう。また、インデックス更新の頻度も高くなってしまった結果を得た。各ピアは一定の確率で他のピアと leave する。このとき、最も検索結果が良くないピアに対して leave する。各ピアが他のピアに対して leave を行う場合、各ピアは過去の検索結果を下に leave 対象となるピアを選択する。この状況下で NDI を用いた場合、有用度の高いピアへ問い合わせメッセージを送信したとしても、多くの検索結果が得られるとは限らないため、有用度の高いピアを推奨したピアに対して leave する可能性が高い。よって、インデックス更新の頻度は高くなってしまった。

TDI を用いた場合、他のインデックスを用いた場合に比べてトピック数の増加に対応しているといえる結

果を得た。トピック数の増加に対して、問い合わせの成功率は低下し問い合わせメッセージ数は増加してしまいが、ホップ数はあまり変化が無かった。この状況下にて TDI と複製配置を組み合わせた場合、TDI のパフォーマンスは飛躍することも分かった。問い合わせ成功率に関して TDI+PIR の組み合わせは NDI を用いた場合と比較した場合、大幅なパフォーマンス向上を確認することができた。また、TDI を PRP と組み合わせた場合よりも成功率に関するパフォーマンスが優れている。DI を用いた場合、各ピアは問い合わせメッセージをあまり受け取らないピアが存在する。PRP では他のピアの問い合わせを利用した文書複製の方法である。そのため、多くの問い合わせメッセージを受け取るピアが文書複製を行う場合、PRP 方式によって効果的な文書複製の配置が期待される。しかしながら、あまり問い合わせメッセージを受け取らないピアが PRP 方式による文書複製配置を行う場合、PRP 方式の効率性を発揮できないためそのパフォーマンスを低下させてしまうと考えられる。

## 6. おわりに

P2P システムを利用した協調作業システムでは、効率的な問い合わせ処理が求められる。そこで、本研究では P2P システムの問い合わせ処理の向上のために 2 つ Direct Index (DI) を導入した。更に、これらの DI が問い合わせ処理へ与える効果を確かめる実験を行った。その結果、本研究で提案した TDI はトピック増加に対して柔軟に対応し、応答時間の向上および帯域幅消費量の低減に役立つことが示された。また、TDI と動的なピアグループ形成を基にした複製配置方法を組み合わせることで問い合わせ処理の成功率も向上させることが示された。特に TDI と PRP 方式または PIR 方式の組み合わせは、TDI の効果のさらなる向上に有効であった。

P2P システム上の全ピアが完全なインデックス情報を維持することは困難であるが、効率的な問い合わせ処理のためにインデックス更新は重要な要因である。そこで、今後はインデックス更新の頻度が与える問い合わせ処理への効果について分析し、効果的なインデックスの更新法を研究する必要があると考えている。

## 参考文献

- 1) Aberer, K.: P-Grid : A self-organizing access structure for P2P information systems, *CoopIS 2001*, Trento, Italy (2001).
- 2) Chen, Y., Katz, R. and Kubiawicz, J.: Dynamic Replica Placement for Scalable Content Delivery, *IPTPS'02*, Cambridge, USA (2002).
- 3) Crespo, A. and Garcia-Molina, H.: Routing Indices For peer-to-peer Systems, *ICDCS 2002*, Vienna, Austria (2002).
- 4) Datta, A., Hauswirth, M. and Aberer, K.: Updates

- in Highly Unreliable, Replicated Peer-to-Peer Systems, *IEEE ICDCS 2003*, Rhode Island (2003).
- 5) GrooveNetworks: <http://www.groove.net/>.
- 6) ICQ: <http://web.icq.com/>.
- 7) Kubiawicz, J et al.: OceanStore: An Architecture for Global-Scale Persistent Storage, *ASPLOS 2000*, Cambridge, USA (2000).
- 8) Ratnasamy, S et al.: A scalable content-addressable network, *ACM SIGCOMM 2001*, San Diego, USA (2001).
- 9) Rowstron, A. and Druschel, P.: Pastry: Scalable, distributed object location and Routing for large-scale peer-to-peer systems, *Middleware 2001*, Heidelberg, Germany (2001).
- 10) Stoica, I., Rorris, R., Karger, D., Kaashoek, M. and Balakrishnan, H.: Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications, *ACM SIGCOMM 2001*, San Diego, USA (2001).
- 11) Sun, Q. and Garcia-Molina, H.: Partial Lookup Services, *IEEE ICDCS 2003*, Rhode Island, USA (2003).
- 12) Tsoumakos, D. and Roussopoulos, N.: Adaptive Probabilistic Search for Peer-to-Peer Networks, *P2P'03*, Linkoping, Sweden (2003).
- 13) Yamada, T., Aihara, K., Takasu, A. and Adachi, J.: A Distributed Index System for Efficient Query Processing in Peer-to-Peer Networks, *PACRIM'03*, Victoria, Canada (2003).
- 14) Yamada, T., Aihara, K., Takasu, A. and Adachi, J.: Replica Placement for Effective Document Sharing Mechanisms in Peer-to-Peer Networks, *IMSA 2004*, Hawaii, USA (2004).
- 15) Yang, B. and Garcia-Molina, H.: Comparing Hybrid Peer-to-Peer Systems, *VLDB 2001*, Roma, Italy (2001).
- 16) Yang, B. and Garcia-Molina, H.: Improving Search in peer-to-peer Networks, *ICDCS 2002*, Vienna, Austria (2002).
- 17) Zhao, B., Kubiawicz, J. and Joseph, A.: Tapestry: An Infrastructure for Fault-tolerant Wide-area Location and Routing, Technical report, U. C. Berkeley Technical Report UCB//CSD-01-1141 (2000).
- 18) 山田太造, 相原健郎, 高須淳宏, 安達淳: Peer-to-Peer システム上での効率的なデータ配置による問い合わせ処理とロードバランスへの寄与, 情報処理学会論文誌: データベース (TOD22), Vol. 45, No. SIG5, (2004). to appear.