P2P ネットワークにおける属性条件合致ピアの概数を推定可能なピア検索手法の検討

白木　徹[†]　寺西　裕一[†,††]　春本　要[†††]　竹内　亨[††]　西尾章治郎[†]

† 大阪大学大学院情報科学研究科　〒 565–0871 大阪府吹田市山田丘 1-5
†† 独立行政法人 情報通信研究機構　〒 184–8795 東京都小金井市貫井北町 4-2-1
††† 大阪大学大学院工学研究科　〒 565–0871 大阪府吹田市山田丘 2-1
E-mail: {shiraki.toru,teranisi,nishio}@ist.osaka-u.ac.jp, harumoto@eng.osaka-u.ac.jp, stakeuti@nict.go.jp

あらまし　マーケティングなどを目的とするアプリケーションにおいて，あるアイテムの購買履歴を持つユーザが何人居るかといったことを知りたい場合，指定された検索条件を満たす対象の概数を知ることで目的が達成できる．本研究では，P2P ネットワークにおけるピア検索を対象とし，指定条件に合致するピア数の概数を推定する手法を提案する．提案手法では，Chord 上で Spectral Bloom Filter を用いてピアの持つ要素とその数を集約し，検索合致ピア数の確率分布を尤度関数を用いて推定する．提案手法をシミュレーションによって評価し，有効性を示す．

キーワード　P2P ネットワーク, 検索合致ピア数推定，Spectral Bloom Filter

A Study on a Method for Estimating Approximate Number of Peers Matching Attribute Conditions on P2P Network

Toru SHIRAKI[†], Yuuichi TERANISHI[†,††], Kaname HARUMOTO[†††], Susumu TAKEUCHI[††], and Shojiro NISHIO[†]

† Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka 565–0871 Japan
†† National Institute of Information and Communications Technology
4-2-1 Nukuikitamati, Koganei, Tokyo 184–8795 Japan
††† Graduate School of Engineering, Osaka University
2-1 Yamadaoka, Suita, Osaka 565–0871 Japan
E-mail: {shiraki.toru,teranisi,nishio}@ist.osaka-u.ac.jp, harumoto@eng.osaka-u.ac.jp, stakeuti@nict.go.jp

1. はじめに

膨大な数のピア間でサーバを用いずに情報共有が可能な P2P ネットワークに注目が集まっており，ファイル共有や文書共有だけでなく，条件に合致するユーザの探索などにも応用されている．こうしたアプリケーションでは，ピアが保持するコンテンツ等に付与された属性集合（属性と値の組の集合やキーワード集合など）に対して，ある属性条件に合致するコンテンツを保持するピアを発見するピア検索の機能が要求される．例えば，あるキーワードを含む文書を保持するピアの検索や，ある趣味をもつユーザの検索といったピア検索の機能が実現できる．

こうしたピア検索を実現するための要素技術として，構造化 P2P ネットワークの研究が盛んである．構造化 P2P ネットワークには，Distributed Hash Table (DHT) や Skip Graph [1] などがある． DHT は，Consistent Hashing などによりキー値に対応する値を効率的に分散管理する手法である．Skip Graph は，キー値に基づいて階層化されたオーバレイネットワークを構成することにより，範囲検索にも対応できるオーバレイネットワーク技術である．いずれの手法も，属性条件をキー値に対応付けることによって，指定された属性条件を満たすコンテンツを保持するピアを効率的に検索することができる．

一方，必ずしもどのピアが属性条件を満たすコンテンツを保持しているかを知る必要がなく，属性条件に合致するコンテンツを保持しているピアの数を知ることができれば十分であるようなアプリケーションもある．例えば，各ユーザがそれぞれ 1 つのピアとして P2P ネットワークに参加する図 1 のような環

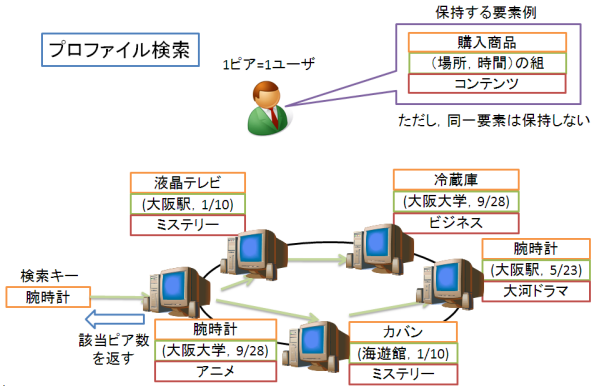

図 1　想 定 環 境

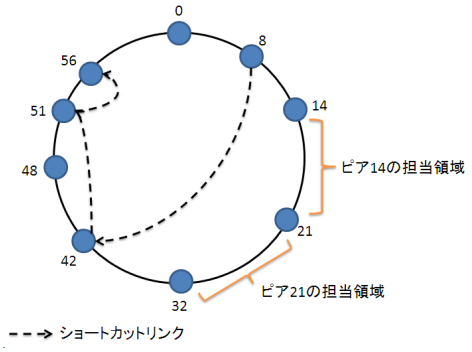

図 2　DHT の例 (Chord)

境において，各ピアがユーザの購買履歴を保持している場合，ある物品の需要を見積もるためには誰がその物品を購入したかを知る必要はなく，その物品を購入したユーザ数すなわちピア数を知ることができればよい．また，ユーザが訪れたスポットの履歴を各ピアが保持している場合，インフルエンザ感染者が発見された場所付近のスポットを訪れたユーザ数を知りたい場合もある．文書検索においてよく用いられる TF-IDF 法では，ある単語が出現する文書数 (DF) を求めることが必要であるが，どのピアがその単語を含む文書を保持しているかという情報は不要である．

しかし，従来の P2P ネットワーク技術を用いて属性条件に合致するピア数を得るためには，与えられた属性条件を満たすピアをすべて特定した上で，その数を数える必要があり，合致ピア数が多いときそのコストは非常に大きくなってしまう．

本研究では，DHT の Chord [8] を拡張した手法である Bloom Finger Table (BFT) [7] の考え方を拡張し，ピアを全て特定することなく属性条件を満たすピア数の概数を得ることができる新たな方法を提案する．BFT では，各ピアは保持する要素の集合を Bloom Filter [2] で管理する．Bloom Filter は，要素集合を固定長のビット列で表現する手法であり，ある要素が含まれているかどうかを効果的に判定することができる手法である．BFT でのルーティングテーブルは Chord と同様の Finger Table で管理され，リンク先のピアの担当範囲に含まれるピアの Bloom Filter を論理和で集約したものを Finger Table 内に管理する．これにより，検索条件に合致するピアが存在する可能性がないリンク先への検索メッセージの転送を抑制することができるが，合致ピア数を得るには検索条件に合致するすべてのピアにメッセージを転送する必要がある．提案手法では Bloom Filter の集約の際に論理和ではなく算術和を用いることにより，リンク先にどの程度の合致ピアが存在するかを判断できるようにするというのが新たなアイデアである．

以下，2 章では関連研究である構造化 P2P ネットワークについて述べる．3 章では提案手法について説明する．4 章では，3 章で提案した手法の有効性をシミュレーションによって評価し，考察する． 5 章で本論文をまとめ，今後の課題について述べる．

2. 構造化 P2P ネットワークによるピア検索

2.1 想 定 環 境

本研究では，P2P ネットワークにおけるピア検索を対象とした合致ピア数検索の実現を目指す．ピア検索とは，一般には指定された条件を満たすピアを特定することを指すが，本研究では各ピアが複数の重複しない要素を保持する状況において，指定された要素を保持するピアを特定することを指す．この要素を「属性＝値」という形式で表現すれば，ピア検索とは特定の属性条件に合致するピアを特定することになる．ピアは P2P ネットワークを構成する論理的な単位であり，単一の計算機上に複数のピアを稼働させてもよいものとする．ユーザとピアが 1 対 1 に対応付けられる場合には，例えばユーザが購入したそれぞれの商品，嗜好情報などのプロファイル，訪問したことがあるそれぞれの場所などのユーザ情報をピアが保持する要素と捉えることができ，これによってユーザ検索が実現できる．また，コンテンツとピアが 1 対 1 に対応付けられる場合には，コンテンツがもつキーワードをピアが保持する要素と捉えることができ，コンテンツのキーワードによる検索が実現できる．

このようなピア検索において，検索で指定された要素を保持するピア数を求めることを，**合致ピア数検索**と呼ぶ．これは，ある要素をもつ検索対象がどれくらい存在するかを純粋に知るのみで目的を達成できるアプリケーションに有効である．

2.2 単一キーによる合致ピア数検索

各ピアが複数の要素を保持している状況において，指定した単一の要素（キー） を保持するピアを検索できる構造化 P2P ネットワークとして Distributed Hash Table (DHT) が挙げられる．

DHT に は Chord，Pastory [5]，Tapestry [9]，CAN [4]，Kademlia [3] などの方式がある．方式は異なるが，いずれもピア間で特定の規則に基づく何らかのリンクを構成することにより， $O(\log n)$ で指定したキーに対応する値を検索でき，また，同様の理由で $O(\log n)$ でキーの登録ができる（n はピア数）．DHT では各ノードはハッシュ空間の一部を担当し，それに基づきキーの担当ピアが決定される．例えば Chord では図 2 のように各ノードはノード ID のハッシュ値によってリング上のハッシュ空間にマッピングされ，担当範囲が決定される．各ピアはハッシュ空間上で 2 のべき乗の距離だけ離れた位置に時計

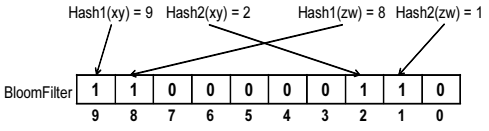

図 3　Bloom Filter

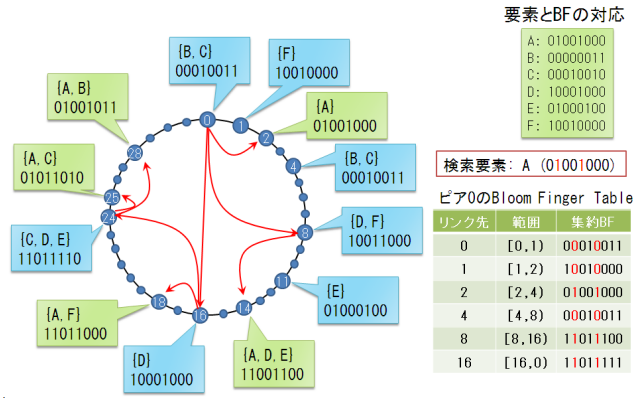

図 4　Bloom Finger Table

回りで最も近い位置にマッピングされているピアへのリンクを Finger Table に保持することにより，指定されたキーを担当するピアに効率的にメッセージを転送することができる．

DHT を用いて合致ピア数検索を実現する場合，要素をキーとしその要素を保持するピア ID を値とする方法が考えられる．しかしその場合，あるピアが m 個の要素を保持する場合，m 個のピアに登録しなければならない．従って，そのようなピアが新たに P2P ネットワークに参加する際には要素の登録コストとして $O(m \log n)$ の通信が必要となる．また，特定の要素はすべて特定のピアに登録されることになり，ある要素を保持するピア数は容易に得られるものの，負荷の均一化の観点で問題がある．

2.3　複数キーによる合致ピア数検索

Chord を拡張し，1 回の検索で複数要素を検索可能とする手法に Bloom Finger Table (BFT) [7] がある．BFT は Chord におけるリンク構造を管理する Finger Table を，Bloom Filter を用いて拡張したものである．

Bloom Filter とは，要素集合を固定長のビット列で表現するデータ構造である（図 3 参照）．具体的には，ビット長を m とした場合，ビット集合に含まれる各要素に対して $[0, m-1]$ の整数値を返す k 個のハッシュ関数によって k 個のハッシュ値を求め，対応するビットを 1 にセットしたものが Bloom Filter となる．ある要素（集合）が含まれているかどうかを検査するには，その要素（集合）から Bloom Filter 生成時と同様に k 個のハッシュ値をそれぞれ求め，Bloom Filter の対応するビットがすべて 1 でなければその要素（集合）は含まれていないと判断することができる．逆に，対応するビットがすべて 1 であれば，その要素（集合）は含まれている可能性が高いと判断できるが，ハッシュの衝突による偽陽性が発生する可能性もある．偽陽性発生確率は，ビット長 m，ハッシュ関数の数 k，追加される要素数 n に依存する．

BFT では，各ピアは自身が保持する要素集合から Node Bloom Filter (NBF) を生成する．さらに Finger Table 上のリンク先が担当する範囲 $[2^{t-1}, 2^t)$ に存在するピア群の NBF の論理和を取り集約したものを Finger Table と対応付けて Finger Bloom Filter (FBF) として保持する．検索時は，検索要求の要素集合に対応する Request Bloom Filter (RBF) を生成し，Finger Table 内の FBF のうち RBF を包含する FBF をもつリンク先ピアへ検索要求を転送する．これを各ピアが繰り返すことにより，検索する要素集合をもつピアまで検索要求が転送されていく．例えば，図 4 では，ピア 0 の Finger Table は自身の FBF（ピア 0 の NBF）とピア 1 の FBF（ピア 1 の NBF），ピア 2，ピア 3 の論理和から生成した FBF，ピア 4 から 7 までの論理和から生成した FBF，ピア 8 から 15 までの論理和から生成した FBF，ピア 16 から 31 までの論理和から生成した FBF を含んでいる．図 4 ではピア 2，8，16 の FBF が RBF を包含しているため，それらのピアへ検索要求が転送され，受信したピアでも同様に検索要求の転送を繰り返す．この方法により，全ての条件合致ピアを見つけてその数をカウントすれば条件合致ピア数を求められる．

ただし，合致ピア数を知る目的では，DHT と同様に合致ピアすべてに検索要求を転送する必要がある．

3.　提 案 手 法

BFT は，検索要求に合致するピアがリンク先の集約範囲に存在する可能性があるかどうかの判定は可能であるが，リンク先に合致ピアがいくつ存在するかはわからない．そこで提案手法では，FBF の構成時に集約範囲の論理和ではなく算術和をとることによって，リンク先の集約範囲にいくつの合致ピア数が存在するかを推定できるようにする手法を提案する．

FBF として論理和ではなく算術和をとることにより，生成される FBF は 0 と 1 の二値の Bloom Filter ではなく，0 以上の整数値をとる Bloom Filter となる．このような Bloom Filter は，同一要素を複数含むことを想定した Bloom Filter である Spectral Bloom Filter (SBF) [6] と呼ばれる Bloom Filter となる．次節では，SBF について簡単に説明する．

3.1　Spectral Bloom Filter

Spectral Bloom Filter (SBF) は，通常の Bloom Filter に対し多重集合 (multi-set) を扱えるようにしたデータ構造である．k 個のハッシュ関数を $h_1, \ldots, h_k$，SBF を構成する m 個のカウンタ値を $C_0, \ldots, C_{m-1}$ とすると，ある要素 x を f_x 個追加する場合，$C_{h_1(x)}, \ldots, C_{h_k(x)}$ を f_x だけ増加させる（図 5 参照）．

このようにして得られる SBF では，検索要素を x とするとき，k 個のカウンタ値 $C_{h_i(x)} (1 \leqq i \leqq k)$ の最小値が，要素 x が含まれる上限数となる．このとき，ハッシュの衝突によって k 個のカウンタ値の最小値が実際の要素数とならない場合がある．例えば図 5 の中段の状態では A が 3 個，B が 2 個，C が 4 個追加されているような SBF が得られているが，要素 B の数は，対応するカウンタ値（5 と 6）の最小値である 5 ではなく，実際には 2 である．このように k 個のカウンタ値の最小値が実際の要素数とならない確率は，Bloom Filter の偽陽性の確率と

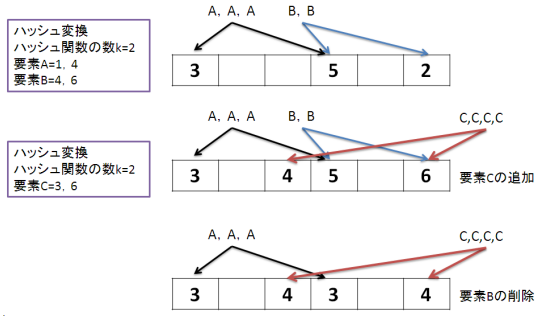

図 5　SBF における要素の追加・削除

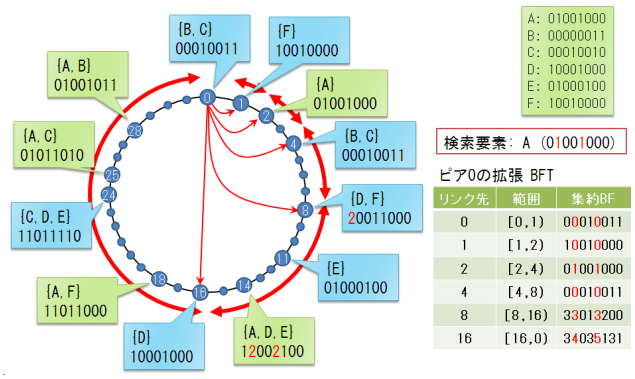

図 6　適 用 方 法

同様に k 個のカウンタ値がすべて他の要素によって増加している確率であり，個別要素数を n とした場合に以下の式で表される．ここで，個別要素数とは互いに異なる要素の数である．

$$\left(1-\left(1-\frac{1}{m}\right)^{kn}\right)^{k} \approx (1-e^{-kn/m})^{k} \tag{1}$$

3.2　適 用 方 法

BFT では，各ピアの NBF は通常の Bloom Filter であったが，そのまま算術和で集約すると SBF とならない．そこで，各ピアの NBF は Counting Bloom Filter (CBF) [10] を適用する．CBF は Bloom Filter に対して要素の削除に対応するために拡張されたものであり，SBF と同様に 0 以上の整数値をカウンタ値をとるものである．各ピアは 1 つの要素を複数保持しないことを想定しているため，この CBF を算術和で集約したものを FBF とすることにより，各 FBF は対応する集約範囲に存在するピアが保持する要素集合を表現した SBF となる（図 6 参照）．また，検索に合致するピア数を求めることと検索に合致する要素数を求めることは同義となる．

3.1 節で述べたように，検索に合致する要素数は検索要素に合致するカウンタ値の最小値が上限となる．しかし，集約範囲が大きくなると追加される要素数が増加するため，カウンタ値の最小値が実際の要素数と異なる確率が大きくなる．例えば，$k=2$，カウンタ長 $m=5000$ の場合，各ピアが 100 個の要素を保持している状況で 512 ピアの NBF を集約して得られた SBF では，SBF に追加されている 51,200 要素のうちの個別要素数が 10,000 要素であったとすると，式 (1) より

$$\left(1-\left(1-\frac{1}{5000}\right)^{2\times 10000}\right)^{2} \approx 0.9637$$

となり，カウンタ値の最小値を用いるだけでは約 0.9637 の確率で正しい要素数が得られない．そこで，SBF のカウンタ値から実際の合致要素数，すなわち合致ピア数を推定することは必要となる．

3.3　検索合致ピア数の推定

3.3.1　集約範囲ごとの合致ピア数の推定

まず，Finger Table における各リンク先に対応する集約範囲ごとに，合致ピア数を推定する．検索要素 x に対し，合致ピア数を確率変数 X_i（i は集約レベル）とし，SBF で表現されたレベル i の FBF のカウンタ値を $\mathbf{C}^i=(C^i_0,\ldots,C^i_{m-1})$，該当カウンタ値の最小値を $m^i_x=\min^k_{j=1}(C_{h_j(x)})$ とすると，求めたい確率分布は以下のような条件付き確率である．

$$P(X_i=s_i \mid \mathbf{C}^i=(C^i_0,\ldots,C^i_{m-1}))$$

ただし，

$$0 \leqq s_i \leqq m^i_x$$

である．

ここで，ベイズの定理より，

$$\begin{aligned}
&P(X_i=s_i \mid \mathbf{C}^i=(C^i_0,\ldots,C^i_{m-1}))\\
&= \frac{P(\mathbf{C}^i=(C^i_0,\ldots,C^i_{m-1}) \mid X_i=s_i)P(X_i=s_i)}{P(\mathbf{C}^i=(C^i_0,\ldots,C^i_{m-1}))}\\
&= \frac{P(\mathbf{C}^i=(C^i_0,\ldots,C^i_{m-1}) \mid X_i=s_i)P(X_i=s_i)}{\sum^{m^i_x}_{l=0} P(\mathbf{C}^i=(C^i_0,\ldots,C^i_{m-1}) \mid X_i=s_i)P(X_i=s_i)}\\
&\propto P(\mathbf{C}^i=(C^i_0,\ldots,C^i_{m-1}) \mid X_i=s_i)P(X_i=s_i)
\end{aligned}$$

であり，事前分布 $P(X_i=s_i)$ を一様分布とすると，

$$\begin{aligned}
&P(X_i=s_i \mid \mathbf{C}^i=(C^i_0,\ldots,C^i_{m-1}))\\
&\propto P(\mathbf{C}^i=(C^i_0,\ldots,C^i_{m-1}) \mid X_i=s_i)
\end{aligned}$$

となる．つまり，検索要素 x の合致ピア数についての事前知識がないとすると，合致ピア数 X_i の確率分布は尤度関数 $P(\mathbf{C}^i=(C^i_0,\ldots,C^i_{m-1}) \mid X_i=s_i)$ に比例する．

ここで，$X_i=s_i$ という条件の下で $\mathbf{C}^i=(C^i_0,\ldots,C^i_{m-1})$ となる確率は，x を s_i 個追加する前の状態の SBF に対して x を s_i 個追加した結果，k 個のカウンタ値が $C^i_{h_j(x)}(1 \leqq j \leqq k)$ というカウンタ値になる確率である．したがって，$\mathbf{C}^i$ の中の k 個のカウンタ $C^i_{h_j(x)}$ から s_i を減じた状態の SBF においてカウンタ値 u の出現頻度を $hist_{s_i}(u)$ とすると，

$$\begin{aligned}
&P(\mathbf{C}^i=(C^i_0,\ldots,C^i_{m-1}) \mid X_i=s_i)\\
&= \prod^{k}_{j=1} \frac{hist_{s_i}(C^i_{h_j(x)}-s_i)}{m}
\end{aligned}$$

となる．

例えば，図 6 のピア 16 から 31 までを集約したレベルについての合致ピア数の確率分布の導出方法を図 7 を用いて説明す

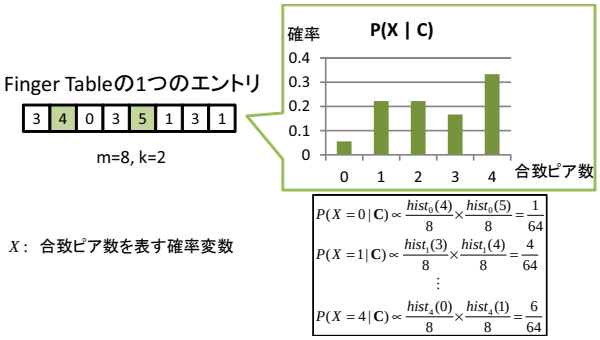

図 7　尤度関数による分布推定

る．合致ピア数を表す確率変数を X とすると，

$$
\begin{aligned}
&P(X=0 \mid \mathbf{C}=(3,4,0,3,5,1,3,1)) \\
&\propto \prod_{l=1}^{k} \frac{hist_0(C_{h_j(x)}-0)}{m} \\
&= \frac{hist_0(4)}{8} \times \frac{hist_0(5)}{8} = \frac{1}{8} \times \frac{1}{8} = \frac{1}{64} \\
&P(X=1 \mid \mathbf{C}=(3,4,0,3,5,1,3,1)) \\
&\propto \prod_{l=1}^{k} \frac{hist_1(C_{h_j(x)}-1)}{m} \\
&= \frac{hist_1(3)}{8} \times \frac{hist_1(4)}{8} = \frac{4}{8} \times \frac{1}{8} = \frac{4}{64}
\end{aligned}
$$

というように計算でき，図に示すような確率分布を得ることができる．

3.3.2　全集約範囲における合致ピア数の推定

前節までで，各集約範囲ごとの合致ピア数の確率分布が求められる．全ピア中の合致ピア数を求めるためには，求められた集約範囲ごとの確率分布を足し合わせる必要がある．ここで，各集約範囲における合致ピア数の確率変数を $X_i(0 \leqq i \leqq maxlv)$（$maxlv$ は最大レベル）とすると，確率変数 X_i の和の確率分布を求めることとなる．ここで，ピアはハッシュ関数のよってハッシュ空間にランダムに配置されるため，集約単位ごとの合致ピア数は集約範囲に含まれるピア数に応じた多項分布に従うと考えられる．

具体的には，2 つの確率変数 X_i と X_j について，FBF から求められる最大値をそれぞれ m_x^i, m_x^j とし，集約範囲に含まれるピア数をそれぞれ w_i, w_j とすると，

$$
\begin{aligned}
&P(X_i=x_i \wedge X_j=x_j) \\
&= P(X_i=x_i|X_j=x_j)P(X_j=x_j) \\
&\propto {}_{x_i+x_j}C_{x_i} \left(\frac{w_i}{w_i+w_j}\right)^{x_i} \left(\frac{w_j}{w_i+w_j}\right)^{x_j} \\
&\quad \times P(X_i=x_i)P(X_j=x_j)
\end{aligned}
$$

となるので，和の確率分布は以下のようにを求められる．

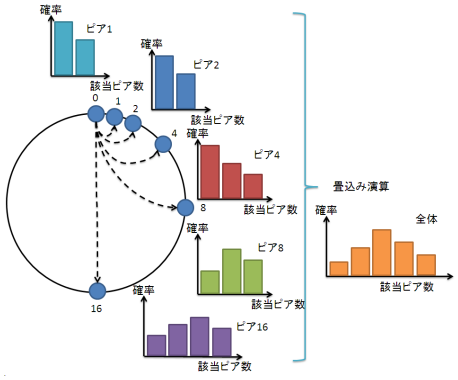

図 8　畳込み演算

表 1　シミュレーション環境

SBF のカウンタ長	5,000
ハッシュ関数の数	2
全要素数	50,000
1 ピアが保持する要素数	100
全ピア数	1,024
Zipf 分布の歪度 (s)	0.5

$$
\begin{aligned}
&P(X_i+X_j=s) = \sum_{t=\max(0,s-m_x^j)}^{\min(m_x^i,s)} P(X_i=t \wedge X_j=s-t) \\
&= \sum_{t=\max(0,s-m_x^j)}^{\min(m_x^i,s)} P(X_i=t|X_j=s-t)P(X_j=s-t) \\
&\propto \sum_{t=\max(0,s-m_x^j)}^{\min(m_x^i,s)} \left\{ {}_sC_t \left(\frac{w_i}{w_i+w_j}\right)^t \left(\frac{w_j}{w_i+w_j}\right)^{s-t} \right. \\
&\quad \left. \times P(X_i=t)P(X_j=s-t) \right\}
\end{aligned}
$$

X_0 から X_{maxlv} までをこのように畳み込むことにより，全集約範囲における合致ピア数の確率分布を得ることができる（図 8）．

4.　評　　価

提案手法の有効性を確認するため，シミュレーションによる評価を行った．

シミュレーション環境の設定値を表 1 に示す．SBF のカウンタ長は 5,000 とし，各ピアは 100 個の要素を保持するものとした．また，各ピアが保持する要素には偏りがあるものとし，以下のような Zipf 分布に従うものとした．

$$
f(k;s,N) = \frac{1/k^s}{\sum_{n=1}^{N} 1/n^s} \qquad (N:\text{全要素数}, k:\text{rank}, s:\text{歪度})
$$

評価においては，全要素数を 50,000 とし，歪度 s を 0.5 とした．

4.1　推　定　例

rank = 0, 5, 10, 20 の要素を指定したときの推定した確率分布とそのときの実際の合致ピア数を図 9, 10, 11, 12 に示す．図 9 の最尤推定値は 206，実測値は 214，推定度は 0.9626 であった．図 10 の最尤推定値は 101，実測値は 94，推定度は 0.9255

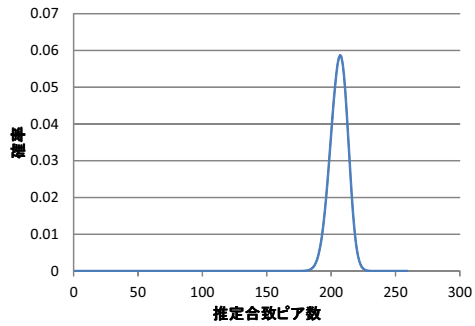

図 9　rank = 0, 合致ピア数 214, 最尤推定値 206

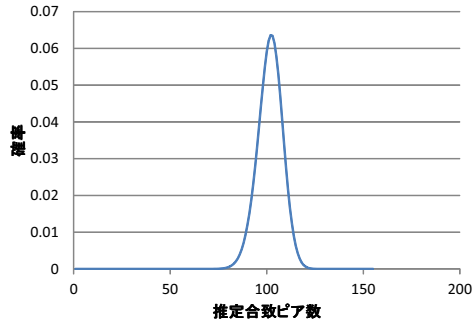

図 10　rank = 5, 合致ピア数 94, 最尤推定値 101

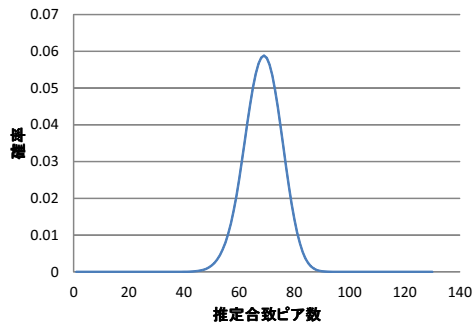

図 11　rank = 10, 合致ピア数 68, 最尤推定値 68

であった．図 11 の最尤推定値は 68，実測値は 68，推定度は 1.0 であった．図 12 の最尤推定値は 24，実測値は 41，推定度は 0.5854 であった．ここで推定度とは，最尤推定値を L，実際の合致ピア数を R とした場合，下記の式で表される評価指標である．

$$S = 1 - \frac{|L - R|}{R}$$

これらの結果から，少ない誤差で正しく推定できている場合もあるが，あまり正しく推定できていない場合もあることがわかる．推定度が低くなる場合は，検索要素のハッシュ値がたまたまカウンタ値の大きいカウンタにマッピングされてしまったためだと考えられる．

4.2　最尤推定値と実測値との差

合致ピア数が 0, 5, 10, 15 であるような要素について，最尤推定値と合致ピア数との誤差を評価した結果を図 13 に示す．この結果から，合致ピア数よりも最尤推定値が大きくなってしまう場合の誤差の分布は，実要素数によらないことがわかる．

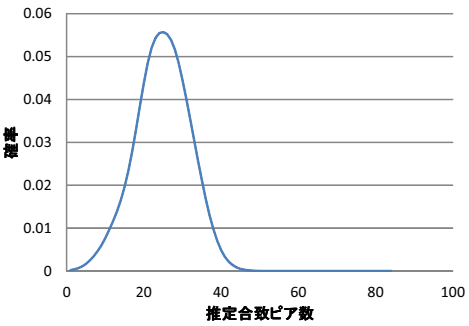

図 12　rank = 20, 合致ピア数 41, 最尤推定値 24

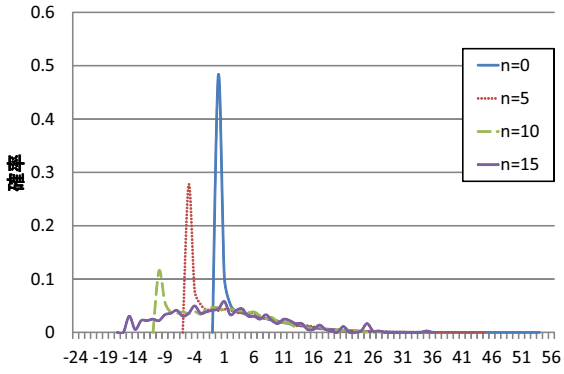

図 13　最尤推定値と実測値との差

また，合致ピア数が 5 や 10 のように比較的小さい場合，最尤推定値が 0 付近となる確率が高いことがわかる．

5.　ま　と　め

本稿では，Chord をベースとした Bloom Finger Table を拡張し，指定条件を満たすピアの概数を推定する方法を提案した．また，推定した確率分布と実際の検索合致ピア数との差を比較し，ある程度まで正しく推定できていることを確認した．

本評価において，最尤推定値と合致ピア数の誤差が大きい場合があったが，そのような場合にはリンク先ピアにメッセージを送信し，受信ピアの保持する Finger Table の集約情報から同様に確率分布を推定し，その結果を返信して確率分布に反映させることで精度が向上できると考えられる．また，提案手法では単一要素の合致ピア数しか求めることができない．AND 検索や OR 検索といった複合検索に対応できるよう，検討を進めたい．

謝　　辞

本研究の一部は，日本学術振興会科学研究費補助金 (22500057) の研究助成および総務省委託研究「ユビキタス・プラットフォーム技術の研究開発」による成果である．

文　　献

[1] Aspnes, J. and Shah, G.: Skip Graphs, ACM Transactions on Algorithms, Vol. 3, No. 4, Nov. 2007.
[2] Bloom, B.H.: Space/Time Trade-offs in Hash Coding With Allowable Errors, Communications of the ACM, Vol. 13, No. 7, pp. 422–426, July 1970.
[3] Maymounkov, P. and Mazieres, D.: Kademlia: A Peer-to-

Peer Information System Based on the XOR Metric, Proc. International Workshop on Peer-to-Peer Systems, pp. 53–65, Mar. 2002.

[4] Ratnasamy, S., Francis, P., Handley, M., Karp, R., and Shenker, S.: CAN: A Scalable Content-Addressable Network, Proc. ACM SIGCOMM'01, pp. 161-172, Aug. 2001.

[5] Rowstron, A. and Druschel, P.: Pastory: Scalable, Decentralized Object Location and Routing for Large-Scale Peer-to-Peer Systems, Lecture Note in Computer Science, Vol. 2218, pp. 329–350, Nov. 2001.

[6] Cohen, S. and Matias, Y.: Spectral Bloom Filters, Proc. ACM SIGMOD'03, pp. 241–252, June 2003.

[7] 佐藤一帆, 松本倫子, 吉田紀彦: 複数キーワード検索に対応した分散ハッシュ型 P2P ネットワーク, 第 6 回情報科学技術フォーラム, CD-ROM, Sept. 2007.

[8] Stoica, I., Morris, R., Kargery, D., Kaashoek, M.F. and Balakrishnan, H.: Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications, Proc. ACM SIGCOMM'01, pp. 149–160, Aug. 2001.

[9] Zhao, B.Y., Kubiatowicz, J., and Joseph, A.: Tapestry: An Infrastruture for Fault-Tolerant Wide-Area Location and Routing, Computer, Vol. 74, pp. 11–20, June 2001.

[10] Fan, L., Cao, P., Almeida, J., and Broder, A.Z.: Summary Cache: A Scalable Wide-Area Web Cache Sharing Protocol, IEEE/ACM Transactions on Networking, Vol. 8, No. 3, pp. 281-293, June 2000.