Vol.2011-MBL-57 No.13
Vol.2011-UBI-29 No.13
2011/3/7

ノード間の通信遅延時間を考慮した Bloom フィルタによる情報検索の提案

唐笠良太[†]　　佐藤文明[†]

Bloom フィルタは，分散システムにおける情報検索方法の一つとして注目されている．Bloom フィルタは，分散ハッシュテーブル（DHT）とくらべて検索に複数のキーワードが使えるなど自由度が高い. 従来の研究で, B 木構造に基づく Bloom フィルタが提案されているが，木の構成方法はノードの ID に基づいており，通信遅延時間が考慮されていなかった．本研究では，B 木構造に基づく Bloom フィルタにおいて，情報検索の際のノード間の通信遅延時間を短縮するための方法として，木構造へのノードの追加・脱退の構成方法について提案し，従来研究との検索時の遅延時間の比較を行った

Information Lookup Method Based on Bloom Filters Composed in Consideration of Communication Delay

RYOTA TOGASA[†]　FUMIAKI SATO[†]

For looking up data in P2P, the method based on the distributed hash table (DHT) is actively researched. On the other hand, the method based on the Bloom filter is also researched so far. Bloom filter is more flexible because two or more keywords can be used to lookup. In the previous research, B-tree structure is used to compose the Bloom filter. Since the previous research uses the node ID to compose the B-tree, the query may be forwarded with long delay. In this research, we propose the method to compose with long delay. In this research, we propose the method to compose the B-tree considering of communication delay. We decide the position of the new node at the tree based on the communication delay. We evaluate the response time of proposed method by simulation.

1. はじめに

近年，情報検索の研究が行われており，検索エンジンのようなクライアントサーバ型の情報検索と異なり，新しい情報検索の方法として P2P 型の情報検索が行われるようになってきた．

P2P における情報検索では，分散ハッシュテーブル[1,2,3,4]を用いた情報検索の方法が研究されている．分散ハッシュテーブルの他にも，代表的なものにハイブリット型 P2P とピュア型 P2P と呼ばれる方法がある．分散ハッシュテーブルを用いた情報検索は，問い合わせをフラッディングするピュア型 P2P に比べてネットワークの負荷が軽く，ハイブリッド型 P2P に比べてサーバの負荷が広く分散できる特徴がある．しかし，複数のキーワードでの検索がしにくく，メンテナンスのコストがかかるという問題点がある．これに対して，Bloom フィルタ[5]を利用した情報検索の方法が従来から研究されている．

Bloom フィルタとは，キーワードに対して複数のハッシュ関数を掛けて得られた値をビット列の位置とみなして，その位置のビットを 1 にすることで得られたビット列である．Bloom フィルタは，そのサイトに蓄積されたコンテンツのすべてのフィルタを OR 演算することで，サイト全体のフィルタを集約することができる．その集約されたフィルタを，検索用フィルタの AND 演算を行うことで，そのサイトに情報が含まれている可能性があるかないかを決定できる．

Bloom フィルタにおける問い合わせの転送方式に，分散ハッシュテーブルの一方式である Chord[4]を使った方式[6]や，m分木の B 木に基づく方式[7,12]が提案されている．従来の B 木の基づく方式では，ノード ID によって木構造への参加位置を決定していたため，通信遅延時間は考慮されていなかった．そのため，検索要求が長い遅延時間を伴って転送されることがあった．

本研究では，m分木の B 木に基づく Bloom フィルタにおいて，情報検索の際のノード間の通信遅延時間を短縮するための方法として，通信遅延時間を用いて木構造に参加・脱退する構成方法について提案する．

2. 関連研究

2.1 P2P

P2P(Peer to Peer)とは，ネットワーク上で対等な関係にある端末間を相互に直接接続し，データを送受信する通信方式である．また，そのような方式を用いて通信するソ

[†] 東邦大学理学部情報科学科
Department of Information Science, Faculty of Science, Toho University

Vol.2011-MBL-57 No.13
Vol.2011-UBI-29 No.13
2011/3/7

フトウェアやシステムの総称である．データの送り手と受け手が分かれているクライアントサーバ方式などと対比される用語で，利用者間を直接つないで音声やファイルを交換するシステムなどが実用化されている．

P2P モデルには情報検索において，大きく分けてハイブリッド型 P2P とピュア型 P2P と呼ばれる方法[8][9]がある．また，近年ではその他に，分散ハッシュテーブルと呼ばれる方式がある．

2.2 分散ハッシュテーブル

分散ハッシュテーブルとは，P2P における情報検索方法の一つである．分散ハッシュテーブルは P2P においてサーバを必要としない．コンテンツとノードの両方に同じハッシュ関数のハッシュ値を割り当て，コンテンツのハッシュ値を受け持つピアへ分散配置する．検索をする際は，検索したいコンテンツのハッシュ値を計算し，そのハッシュ値を受け持つノードに順番に問い合わせを行うことで検索を可能にする．その後，ピア間で直接通信を行う．分散ハッシュテーブルは，ハイブリット型 P2P やピュア型 P2P よりもスケーラビリティが高く，ネットワーク負荷はそれほど上がらず，ネットワーク上のコンテンツを漏れなくかつ高速に探索することを可能にする．

分散ハッシュテーブルを利用した検索アルゴリズムとして代表的なものに Chord や Kademlia がある．

2.3 Bloom フィルタ

Bloom フィルタとは，ある要素がサイトに存在することをハッシュ関数と一定のビット列を用いて表現するデータ構造である．Bloom フィルタを構成するには，まず n ビットのビット列を用意し，全ビットを“0”に初期化する．次に，0 から n-1 の値を返す k 個のハッシュ関数を用意する．そして，キーワードから用意した全てのハッシュ関数を用いてハッシュ値を求める．求めた各々のハッシュ値に該当するビット位置のビットを“1”に変える．Bloom フィルタを用いて要素の有無を判断するには，検索したい要素のハッシュ値に対応する位置の Bloom フィルタのビットを調べる．対応するすべてのビットが“1”であれば，要素が存在すると判断できる．ただし，ハッシュ値が衝突することにより，要素が存在しないにもかかわらず，要素が存在すると判断する可能性がある．逆に，要素が存在するのにもかかわらず，要素なしと判断することは起こりえない．Bloom フィルタには，複数の Bloom フィルタの論理和をとることで，それぞれの Bloom フィルタに含まれるすべての要素を表現する Bloom フィルタが得られるという特徴がある．このとき，要素数にかかわらず Bloom フィルタのビット長は変化しない．

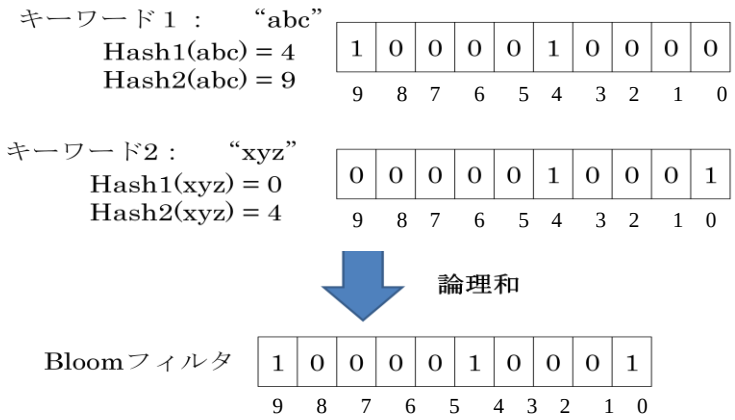

図 2.1　Bloom フィルタの構成例

2.4　B 木構造に基づく Bloom フィルタ

2.4.1　B 木構造

各ノード（P2P インデックス情報を管理している実際のノード）は，分散型の B 木によって管理されている．B 木におけるノードは物理的なノードとは異なるため，論理ノードと呼ぶ．B 木における葉ノードは，物理ノードの ID が格納されているものとする．また，木の中間ノードには，木の接続関係（親ノード，子ノードへの分岐の状況を示すノード ID の配列など）や情報のバージョン番号が管理されている．

この B 木の情報は，参加する物理ノードによって分散管理される．各物理ノードは，B 木の部分情報を持っている．B 木に変更が生じた場合は，他のノードに変更情報を通知することで，全物理ノード間の一貫性を管理している．各物理ノードが持つ部分木の情報は，対応する葉ノードから根ノードまで，木を遡った経路に存在する各論理ノードが持つ情報と，それらに隣接する兄弟ノードが持つ情報である．したがって，m 分木を想定すると，物理ノード数を N としたときに，およそ $\log_m N$ の高さの B 木ができる．各論理ノードには，m 個の分岐を管理するためのサイズ m の配列がある．そして，その各論理ノードに隣接する兄弟ノードの情報を持つと仮定すると，各物理ノードは $m\log_m N$ に比例する個数の情報を持つ．

なお，根ノードおよび各中間ノードは，下に連結された中間ノードの中の一番小さい ID を分岐の区切りとして，配列に保存して管理することとする．この配列中の一番小さい ID を持つ物理ノードを，代表ノードと呼び実際にその配列を管理する責任を持つものとする．

Vol.2011-MBL-57 No.13
Vol.2011-UBI-29 No.13
2011/3/7

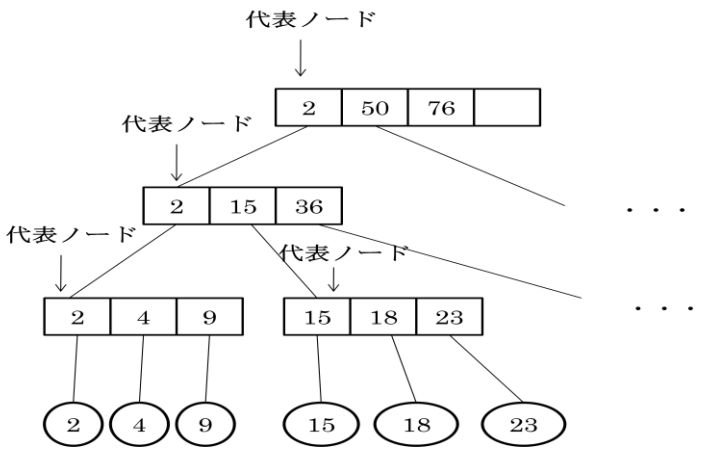

図 2.2　B 木によるノードの管理

また，各物理ノードでは，インデックス情報が管理されおり，各インデックス情報は Bloom フィルタを持っている．各ノードは，ノード内の Bloom フィルタをすべて集約した集約 Bloom フィルタを持っている．また，B 木の構造に基づいて，上位の中間ノードは，下位のノードの Bloom フィルタを集約した集約 Bloom フィルタを持っている．最上位の根ノードには，すべての Bloom フィルタを集約した全体の集約 Bloom フィルタを持っている．

B 木の情報と同様に，Bloom フィルタの情報も各ノードに分散されている．各物理ノードは，B 木の葉から根に至る経路上にある中間ノードに存在する集約 Bloom フィルタと，その隣接する兄弟ノードの集約 Bloom フィルタを持っている．

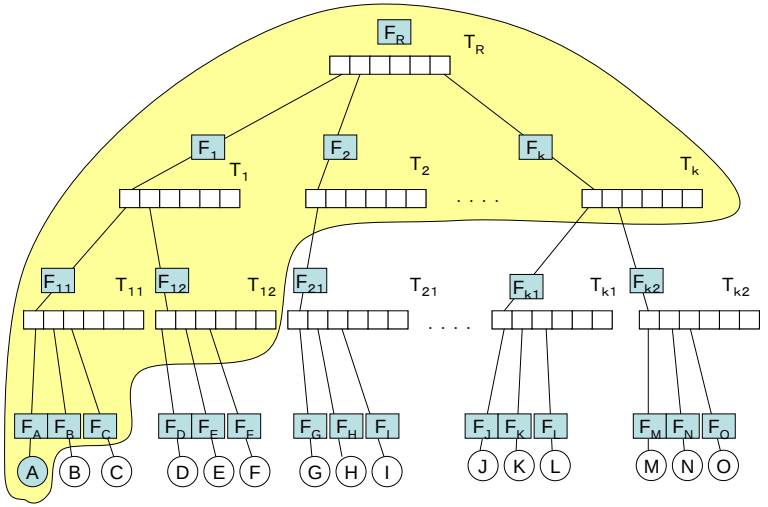

図 2.3　ノード A が持つべき情報の範囲

2.4.2　ノードの参加方法

新規の物理ノードの参加方法は，すでに参加しているノードのどれかにアクセスして自身の ID を含む参加要求を送付するものとする．アクセスされたノードは，根ノードの代表ノードに要求を転送する．要求された根ノードの代表ノードは，要求の ID に基づいて，どの中間ノードに管理されるべきかを判断し，要求を下位の中間ノードに転送する．これを繰り返すことで，管理されるべき最も下位の中間ノードの代表ノードに参加要求が到達する．その中間ノードの配列に空きがある場合，その中間ノードの配列に新規に ID が登録されることで，参加処理は終了となる．

もし，空きがない場合は，中間ノードは B 木におけるノードの分割作業を行う．分割作業は，そのレベルの中間ノードの分割が上位のレベルの分割を引き起こすことがあり，最後には根ノードの分割と，その上位に新たな根ノードの作成が行われることがある．

B 木への新規の参加があったとき，そのノードを収容する中間ノードの代表ノードは，集約 Bloom フィルタの変更を計算する．もし，集約 Bloom フィルタの変更がある場合，その変更が伝搬する最上位の中間ノードの代表に通知し，そこからマルチキャストによって各ノードに伝搬する．

Vol.2011-MBL-57 No.13
Vol.2011-UBI-29 No.13
2011/3/7

2.4.3 ノードの脱退方法

既に参加している物理ノードの脱退方法は，脱退するノードが属する最下位の中間ノードの代表ノードに脱退要求を送付する．代表ノードは，中間ノードの配列から対象ノードが脱退した場合に，要素数がm/2未満にならないなら，ただ配列から対象ノードを削除するだけで脱退処理を終了する．

もし要素数がm/2未満になる場合は，隣接する兄弟ノードから配列の要素を移動する．隣接ノードから配列の要素を移動するときm/2未満になる場合は，その隣接ノードと中間ノードを合併する．この合併操作は，必要に応じて根ノードまで遡り，場合によっては最上位の根ノードを削除する操作までを行う．

B木からの脱退があったとき，そのノードを管理する中間ノードの代表ノードは，集約Bloomフィルタの変更を計算する．もし，集約Bloomフィルタの変更がある場合は，その変更が伝搬する最上位の中間ノードの代表ノードに通知し，そこからマルチキャストによって各ノードに伝搬する．

2.4.4 検索方法

情報の検索では，まず検索用のキーワードから検索用のBloomフィルタを作成する．検索用Bloomフィルタを含む検索要求を，B木のどこかのノードに送付する．検索要求が到達したノードでは，そのノードが保持するB木の部分情報について，根ノードから比較を開始する．B木の各ノードに付随する集約Bloomフィルタと検索用のBloomフィルタとのAND演算を実施して，検索用Bloomフィルタが残る（情報が見つかった）最も下位（つまり，部分木中の葉）のノードを見つける．そのノードが中間ノードであれば，その下の各中間ノードの代表ノードに問い合わせを送る．また，そのノードが葉ノードであれば，対応する物理ノードに直接検索要求を送付する．

転送された検索要求を受け取った中間ノードの代表ノードは，その中間ノード以下のノードに対して検索要求に含まれるBloomフィルタを使って照合を実施する．そして，最も下位のノードに到達するまで，繰り返し検索要求を行う．

図2.4に検索要求の転送例を示す．ノードAに検索用のフィルタ010101が送られてきた場合，ノードAは自身が持つフィルタの情報から，F_kとF_{12}に検索が合致することがわかる．そのことから，その集約されたフィルタに関連するノード群を下位に持つ部分木の代表ノードであるノードDとノードJに検索要求が転送されることになる．

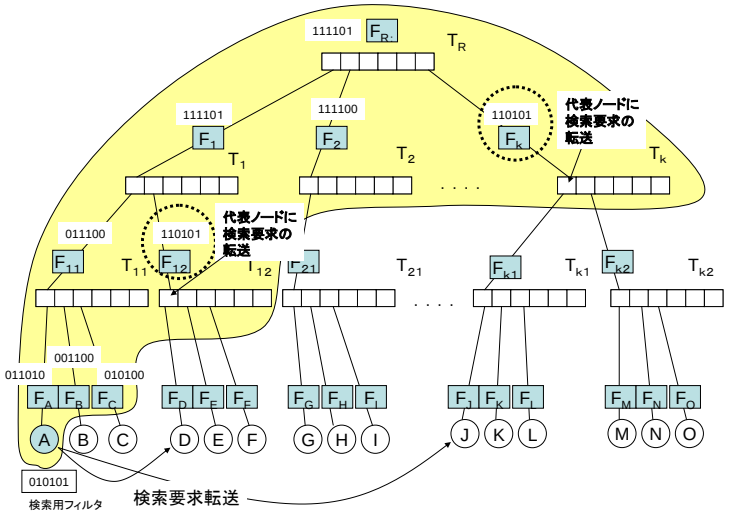

図2.4　ノードAへの検索要求の転送例

3. 提案方式

2.4章で述べられたB木構造に基づくBloomフィルタの構成は，ノードの参加・脱退時に，ノードにつけられたIDの順番になるように木を構成している．そのため，検索時の遅延時間を考慮していない．

本研究では，検索時の遅延時間を短縮するノードの参加方法を二つ提案する．

3.1 提案方式1

3.1.1 新規ノード参加方法

提案方式1での新規ノードの参加する方法は，すでに参加しているノードのどれかにアクセスして参加要求を送付するものとする．アクセスされたノードは，根ノードの代表ノードに要求を転送する．要求された根ノードのそれぞれの下位の中間ノードの代表ノードと新規のノードとの遅延時間を求める．要求された根ノードの代表ノードは，その遅延時間に基づいて，遅延時間の一番短い中間ノードに参加要求を転送する．これを繰り返すことで，管理されるべき最も下位の中間ノードの代表ノードに参加要求が到達する．最も下位の中間ノードに参加要求が到達したら，代表ノードから

Vol.2011-MBL-57 No.13
Vol.2011-UBI-29 No.13
2011/3/7

の遅延時間をそれぞれ求め，すでに参加しているノードと新規のノードとの遅延時間を比較して，順番になるように新規ノードを追加する．

変更情報の通知は，変更が生じた最も上位のノードの代表ノードに依頼して実施してもらう．

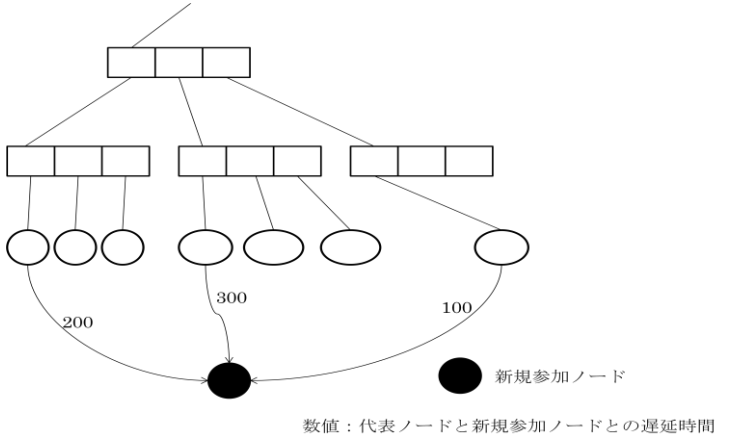

図 3.1　新規ノードと代表ノードとの遅延時間比較例

図 3.1 の場合，新規参加ノードは，一番右の中間ノードに参加要求を転送することになる．

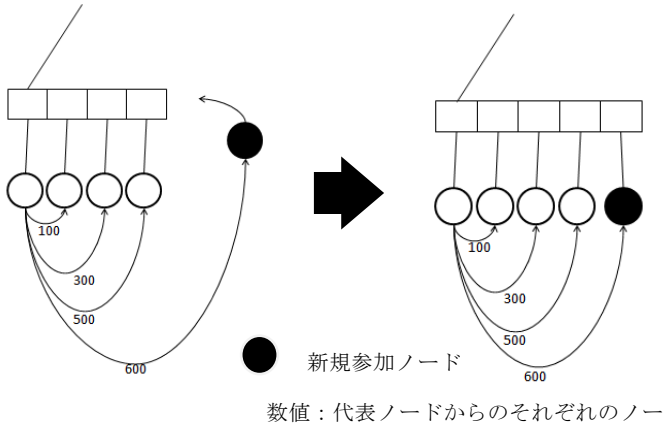

図 3.2　新規ノード参加例

図 3.2 のように，最も下位の中間ノードに要求がきたら，最も下位の中間の代表ノードからそれぞれのノードの遅延時間を求め，遅延時間の順番に並ぶように新規ノードを追加する．

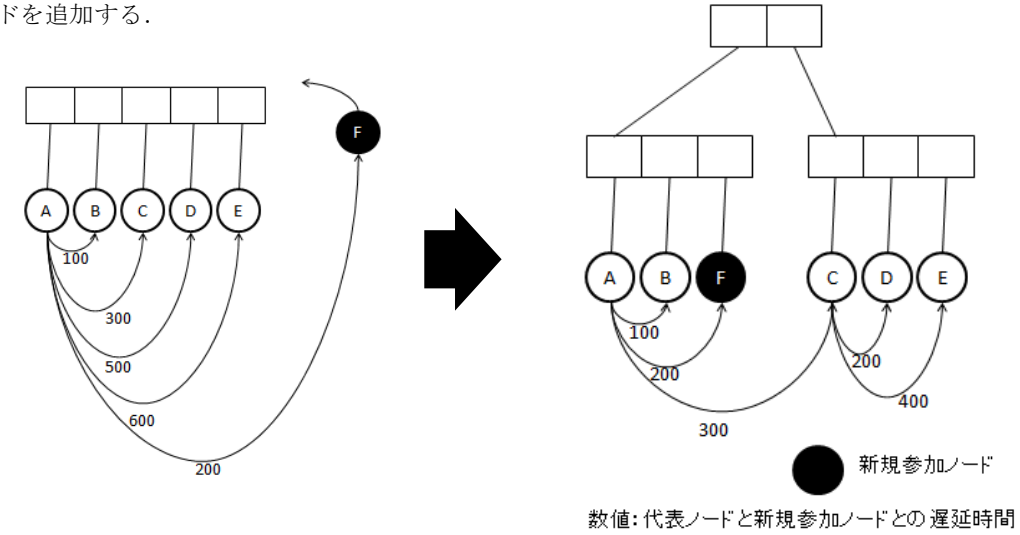

図 3.3　中間ノード分割の例（5 分木の場合）

m分木のＢ木の場合，中間ノードが保持できる子ノード数はm個なので，ノード追加時にその上限を超えたときは中間ノードを 2 つのグループに分ける必要がある．

中間ノードの分割方法は，分割する前に中間ノードが保持する子ノードm個の中から新しく入る子ノードを代表ノードからの遅延時間が短い順に（m+1）/2　－1 個選び，新しく入る子ノードと選ばれた子ノード，それ以外の子ノードで 2 つのグループに分ける．

図 3.3 の場合，代表ノードからの遅延時間はノード B，ノード F，ノード C，ノード D，ノード E という順番になるので，ノード A，ノード B，ノード F のグループとノード C，ノード D，ノード E のグループに分かれる．分かれた時に，ノード C が代表ノードになり，ノード C とノード D，ノード C とノード E の遅延時間を求め，遅延時間の短い順に並び替える．

3.1.2　ノード脱退・検索方法

ノード脱退・検索方法は従来方式のノード脱退・検索方法と同じ操作を行う．

Vol.2011-MBL-57 No.13
Vol.2011-UBI-29 No.13
2011/3/7

3.2 提案方式２

3.2.1 新規ノード参加方法

新規ノードの参加する方法は，すでに参加しているノードのどれかにアクセスして参加要求を送付するものとする．アクセスされたノードは，根ノードの代表ノードに要求を転送する．ここまでは提案方式１と同じことを行う．

提案方式２では，要求された根ノードの代表ノードと根ノードのそれぞれの下位の中間ノードの代表ノードとの遅延時間を計算する．そして，根ノードの代表ノードと新規ノードとの遅延時間を計算し,根ノードの代表ノードからの遅延時間に基づいて，どの中間ノードに管理されるべきかを判断し，要求を下位の中間ノードに転送する．これを繰り返すことで，管理されるべき最も下位の中間ノードの代表ノードに参加要求が到達する．最も下位の中間ノードに参加要求が到達しても同じように，代表ノードからの遅延時間をそれぞれ求め，すでに参加しているノードと新規のノードとの遅延時間を比較して，順番になるように新規ノードを追加する．

変更情報の通知は，変更が生じた最も上位のノードの代表ノードに依頼して実施してもらう．

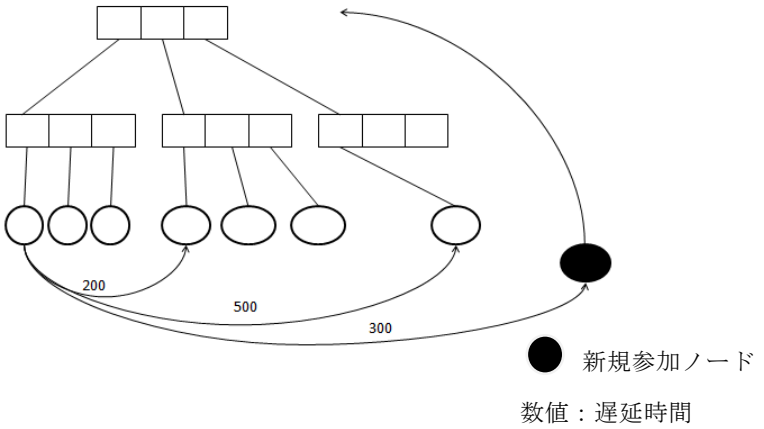

図 3.4　新規ノードと代表ノードとの遅延時間比較例

図 3.4 は，根ノードの代表ノードとそれぞれの下位の中間ノードの代表ノードとの遅延時間と参加ノードと根ノートの代表ノードとの遅延時間を示している.この場合，参加ノードの遅延時間が 300 で，遅延時間 200 と 500 の間なので，根ノードから参加要求が送られる中間ノードは根ノードの真ん中の下位の中間ノードである．最下位の中間ノードまでこれを繰り返し，最下位の中間ノードでも同じように遅延時間に基づいて参加ノードを追加する．

中間ノードが分割する場合は，提案方式１と同じ処理を行う．

3.2.2 ノード脱退・検索方法

ノード脱退・検索方法は従来方式のノード脱退・検索方法と同じ操作を行う．

4. 評価

B 木構造に基づく Bloom フィルタにおいての検索時の遅延時間をシミュレーションを行い，従来方式と提案方式とで結果を比較した．

ネットワークデータを作るツール GT-ITM[11]を使用して，トランジットスタブモデルのネットワークを作り，現実的なネットワークに近いデータを使う．今回のシミュレーションでは，トランジットノード（中継ノード）の数を４，スタブノード（末端ノード）の数を 256 に設定した．GT-ITM ではリンクの遅延がランダムに割り当てられる．シミュレーションにはダイクストラ法を使用し，末端ノード間の最短遅延時間を算出して利用した．

4.1 シミュレーション方法・結果

B 木構造のすべての倫理ノードに Bloom フィルタを持たせたプログラムを使用し，検索要求時に遅延時間を計測するプログラムに改良して，B 木構造に基づく Bloom フィルタの検索要求においての遅延時間を従来方式と提案方式で比較した．

計測方法は，すべてのノードの検索を行い，それぞれに発生する遅延時間を合計して，ノード数で割り，平均の遅延時間を出力する．ノード間の重みをランダムに設定し，数回この処理を行い，その出力された遅延時間の平均をとって比較した．ノード数は 20，50，100，200，500，1000 で固定し，それぞれの遅延時間の平均をだした．

表 4.1 に 5 分木での従来方式と提案方式の遅延時間を計測した結果を示した．図 4.1 に 5 分木での従来方式と提案方式の遅延時間を比較した．表 4.2 に 10 分木での従来方式と提案方式の遅延時間を計測した結果を示し，図 4.2 に 10 分木での従来方式と提案方式の遅延時間を比較した．

Vol.2011-MBL-57 No.13
Vol.2011-UBI-29 No.13
2011/3/7

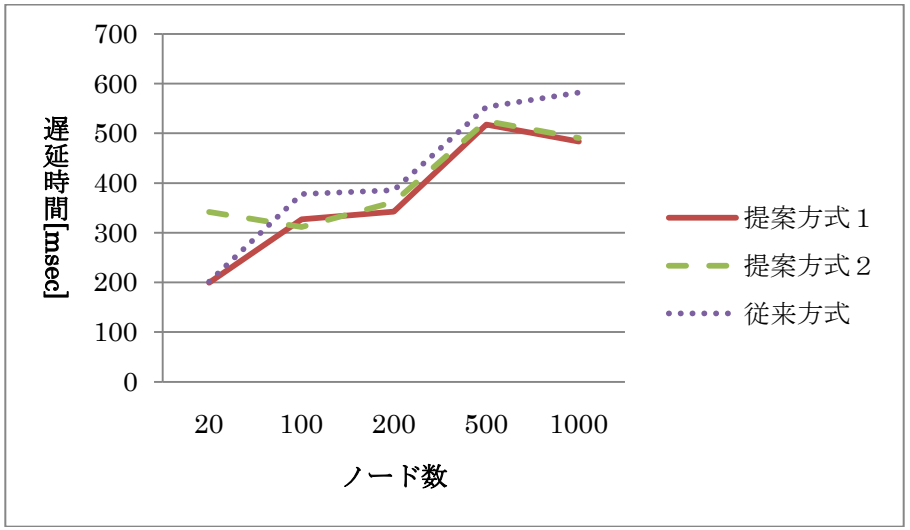

図 4.1　5 分木での遅延時間の比較

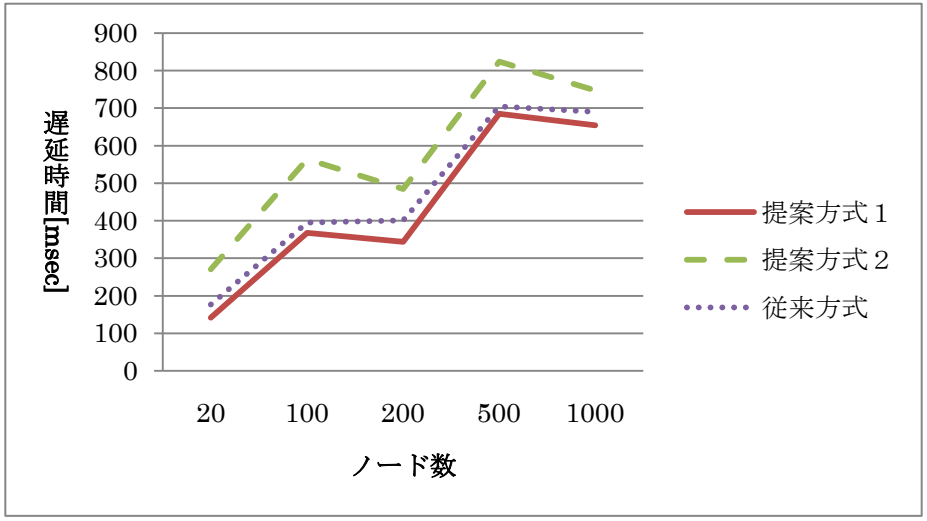

図 4.2　10 分木での遅延時間の比較

4.2　考察

提案方式２は従来方式と比べて，分岐数が少ない時は，提案方式２のほうが遅延時間が短くなった．しかし，分岐数が多い時は，提案方式２のほうが従来方式より遅延時間が長くなった．提案方式２の場合，遅延時間の測定点を根の代表ノード一点にして，遅延時間を計り木を構成していた．したがって，1 ホップ目まではあまり差はないが，2 ホップ目以降が最適ではないので，従来方式のほうが遅延時間が短くなる場合があったと考えられる．

提案方式１は従来方式と比べて，分岐数に関係なく，提案方式１のほうが遅延時間が短くなった．提案方式１の場合，参加するノードに遅延時間が近いノードを選ぶので，2 ホップ目以降の遅延時間が長くなることはない．したがって，提案方式１は従来方式より遅延時間が短くなったと考えられる．

5.　まとめ

従来研究の B 木構造に基づく Bloom フィルタでは，ノード ID によって B 木を構成した場合に，検索時の遅延時間が考慮されていないという問題があった．本研究では，新規ノードを追加する際に，ノード ID によって追加する場所を決めるのではなく，遅延時間を考慮して追加する場所を決めるという方法を提案した．

新規ノード追加方法として，新規ノードと代表ノードとの遅延時間を計算し，適切な場所にノードを追加するというものである．この構成方法によって，検索時の遅延時間を削減する．ノード脱退方法，ノード検索方法は従来方式と同じものを使う．

シミュレーションの結果，ノードをノード ID に基づいて追加する場合よりもノードを遅延時間を考慮して追加した場合のほうが，検索時に発生する遅延時間を削減できることが確認できた．ノード数が少ない場合は，検索時の遅延時間にあまり差はないが，ノード数が多くなればなるほど提案方式のほうが検索時の遅延時間を削減できることがわかる．

今後の課題は，ノードの追加・脱退をするたびに遅延時間を考慮して並び替える必要があるので，B 木を構成する処理に時間がかかってしまう．そのため，その処理時間を削減する必要がある．

謝辞　本研究の一部は，日本学術振興会科学研究費，基盤研究(C)(22500071)の助成を受けたものである．

Vol.2011-MBL-57 No.13
Vol.2011-UBI-29 No.13
2011/3/7

参考文献

[1] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. “A Scalable Content-Addressable Network.” In Proceedings of the ACM SIGCOMM 2001 Technical Conference, San Diego, CA, USA, August 2001.

[2] Antony Rowstron and Peter Druschel. “Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems.” In IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), pages 329-350, November 2001.

[3] Petar Maymounkov and David Mazieres. “Kademlia: A Peer-to-peer Information Systems Based on the XOR Metric.” In Proceedings of the IPTPS 2002, Boston, March 2002.

[4] Ion Stoica, Robert Morris, David Karger, Frans Kaashoek, and Hari Balakrishnan. “Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications.” In Proceedings of the ACM SIGCOMM 2001, San Diego, CA, USA, August 2001.

[5] B. Bloom. “Space/Time Tradeoffs in Hash Coding with Allowable Errors.” Communications of the ACM 13:7, pp.422-426, 1970.

[6] 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”,FIT2007, pp.437-440, 2007.

[7] 若林繁寿, 佐藤文明, “Bloom Filters Based on the B-Tree” 情報処理学会 研究報告 2008-DPS-137 (9) pp43-48,2008.

[8] 星合隆成, 須永宏, “P2P 総論[Ⅰ]～[Ⅳ]”, 電子情報通信学会誌, Vol.87, Vol9, pp804-811, 2004 , Vol.87, No.10, pp887-896,2004, Vol.87, No.12, pp1049-1056, 2004 , Vol.88, No.1, pp46-53,2005.

[9] Hari Balakrishnan, M.Frans Kaashoek, David Karger Robert Morris, and Ion Stoica, “LOOKING UP DATA IN P2P SYSTEMS”, Communications of the ACM Volume 46 Issue 2, February 2003.

[10] S. C. Rhea and J. Kubiatowicz. “Probabilistic Location and Routing.” In Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), Volume 3, pp. 1248-1257, Los Alamitos, CA: IEEE Computer Society, 2002.

[11] Ellen W. Zegura, Kenneth Calvert and M. Jeff Donahoo. “A Quantitative Comparison of Graph-based Models for Internet Topology.” IEEE/ACM Transactions on Networking, pp.770-783, December 1997.

[12] 佐久間洋, 佐藤文明, ”構造型 Bloom フィルタにおける類似性に基づいて集約コスト削減方式の提案”, 情報処理学会「マルチメディア, 分散, 協調とモバイル (DICOMO2010)シンポジウム」,pp.2023-2031, 2010 年 7 月.