

階層型 Bloom Filter を用いた分散ファイル管理

三橋 孝平 森本 亮 山田 卓矢 三浦 孝夫
法政大学工学部電気電子工学科

概要

本稿では P2P ネットワーク上のファイル機構を提案し、ファイルの位置独立性を確立し、またこれらのマイグレーションを効率よく行うことができることを述べる。とくに、効率よくファイル探索を行うため、部分構造を指定したディレクトリ構造に対して、階層的に Bloom Filter を用いる管理方式を提案する。ここでは、JXTA による、サーバなし、マルチキャスト環境での通信網を想定する。この結果、ネットワークへの参加・離脱が自由に行え、少ない通信量を保つことができる。本論文では、Java/JXTA を用いて提案手法を試作しその有効性を実証する。

1 P2P 環境下での情報管理

P2P の狙いは、主にデータ共有としての応用、分散計算による負荷軽減を狙った応用、公開の自由・制限を可能にする自立自治システムへの応用である。また、P2P 環境におけるデータ検索では、管理・格納・検索の観点からファイル単位での管理が現実的である。

Napster に代表されるサーバ方式では、ファイルのインデックスを中央のサーバが管理することで、探索の高速化、セキュリティ管理を簡単に行えるという利点がある。しかし、サーバに機能が集中するため、拡張性と耐久性に乏しいという問題がある。Gnutella に代表される対称方式では、サーバ同士が同等に結合する。そこでは、検索クエリを次々にブロードキャストするため、帯域の大量消費という問題がある。また、事実上のスーパーピアが導入されてしまうという問題がある。

ファイル探索全般における要件としては、ファイルのほとんどが読み出しのみ、サイトの参加・離脱が自由、そして、大域名がなく、大域的整合性を考えないということが言える。また、P2P における検索要件としては、なるべく多くの対象を検索し再現性の向上を図ること、そして、なるべく無駄の少ない対象の検索をおこない適合性の向上を図ることが言える。

2 P2P 分散システムの機能要件

ここで想定する P2P 環境での探索要件について論じる。各ピアは "Give and Take" を受け入れる。つまり、他のピアに対し自由に探索要求を出せる代わりに、他のピアからの非探索要求も常に受け付け、処理を行う。また、ネットワークは小規模のものを扱う。その特色としては、各資材部での情報提供など、あちこちに情報が点在し、それらを目的に応じて気軽に収集できる。通信に関してはパケットループなし、ブロードキャストなしであり、また外部に対しては厳しい機密管理を行うが内部ではそれぞれのピアは同格である。データの更新頻度が大きい場合、レプリカを配置することは同期の面で難しく、回線の負荷を増大させるのみである。さらに、ファイルの移動を受け入れるので、昨日まで東京本社で管理

していた情報が、今日から川崎支社で管理されているかもしれない。最後に、各ピアは自治権（公開制限の自由な設定を含む）を持つ。どの情報をいつ公開するか、などは各ピアで管理可能である。

ファイル名探索手法については [2] でそのいくつかについての性能評価がされている。ブルームフィルタを用いた探索ではファイル構造を扱えないため現実のファイルシステムに適応することは難しい。また、集約フィルタはフィルタのグループ化により階層構造を扱うことができるが、中間ノード置かれたファイルを扱うことができない。

探索手法については、サーバ方式ではデータそのものの、もしくはそのインデックスを一箇所に集中して管理するシステム構成を持つため、データの点在、更新頻度が大、ファイル移動あり、自治権を持つという点で適さない。対称方式では全てのピアが同等の機能を有しており、システムへの参加・離脱が自由なため、要件を満たすことができる。そのため、我々はこの方式を採用する。完全対称 P2P を採用したシステム構成において、パケットループなし、ブロードキャストなしの通信を確立する。そのためには、全てのピアに対するマルチキャスト的機構が必要となる。これは、TCP/IP 方式では実現不可能である。そこで、我々は JXTA を採用する。[3] 階層型ファイル構成を扱うためには、グループ化を扱う集約フィルタと 1 レベルのブルームフィルタを組にして管理する必要がある。そうすることで、完全な再現性を保ちつつ高い適合性を得ることができる。また、他のピアの最高階層の集約フィルタのキャッシュを各ピアが持つことで、探索クエリを送る相手を限定することもできる。この場合、キャッシュを保持しマルチキャスト制御を行うことの負荷と、それによって得られる適合性とのトレードオフを調べる必要がある。

以上のことを踏まえて、我々の基本アイデアについて述べる。構成方式としては、Java による P2P 構成を実装し、利用者の制御と分散ファイル管理を行う。通信方式は JXTA を利用することでマルチキャスト的機能を獲得、そのプラットフォーム非依存性から Unix, Win 間での通信も可能である。そして、探索手法としてはディレクトリ内のファイルをシグニチャコーディングした LBF(Local Bloom Filter) とそれを階層的に管理するための GBF(Global Bloom Filter) を導入する。各ピアは自分が提供するディレクトリ構造についての <ディレクトリ, LBF, GBF> のリストを所持しており、これを階層的に使用することによって、高度な質問に対応することができる。

例 1 : 人事部/アルバイト/住所録

世界中に散らばった支社の人事部が管理している住所録を集めることができる。

例 2 : /東京本社//住所録

東京本社が管理している住所録を全て集めることができる。先頭の "/" はユーザが設定したトップディレクトリの直下であることを表す。

例 3 : /東京本社/*//住所録

「東京本社」ディレクトリから任意の 1 レベルを経由したところに置かれている住所録だけに限定して集めることができる。

"Peer-to-Peer File Management using Hierarchical Bloom Filters" : Kohei Mitsuhashi, Ryo Morimoto, Takuya Yamada, Takao Miura : Hosei University, Dept. of Elec. and Elec. Eng. Kajino-cho 3-7-2, Koganei, Tokyo, JAPAN

3 システムの構成

3.1 全体構成

本システムにおける処理の流れは以下の様になっている。

1. あるピアがシステムに要求を出す
2. 要求を各ピアへ同時に送信する。
JXTA を使用することで、各ピアにマルチキャスト的な通信を行うことができる。
3. 各ピアで要求を処理する。
4. ファイルが見つければ要求元に応答を返す。
5. 応答があった相手に ftp 要求を出しファイルのダウンロードを行う。

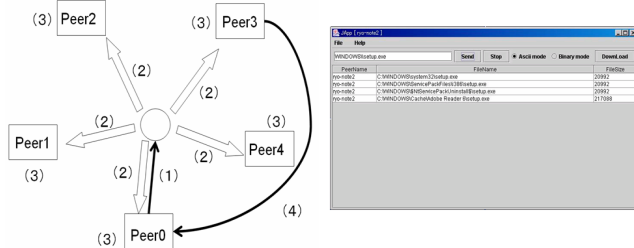


図 1: システムの流れと GUI 画面

3.2 階層型 Bloom Filter

LBF(Local Bloom Filter) は、あるディレクトリに含まれる全てのファイルのファイル名を要素として作られる Bloom Filter である。また、GBF(Global Bloom Filter) はそのディレクトリの LBF と全てのサブディレクトリの GBF とのビットごとの OR 演算を行って得られる Bloom Filter である。そして、全てのディレクトリに対しての<ディレクトリ,LBF,GBF>の組をリスト化したものが、階層型 Bloom Filter である。

実行例として”C://xyz”という質問に対しての処理の流れを図 2 を参考にしながら述べる。

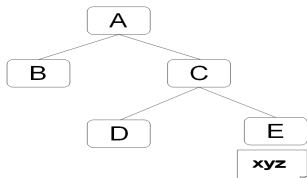


図 2: 階層型 Bloom Filter

”C”, ”xyz”の Bloom Filter を BF(C),BF(xyz) とする。BF(C) と BF(xyz) の両方がディレクトリ A の GBF に含まれていれば、システムにそのディレクトリの名前が”C”であるか聞く。ここではそうでないので、次にサブディレクトリ B と C に GBF が BF(C) と BF(xyz) の両方を含むか聞く。もし含んでいればそのディレクトリの名前が”C”であるか聞く。この場合、C がそうであるので、そのディレクトリの LBF が BF(xyz) を含むか聞く。ここでは含まない。次に C のサブディレクトリ D と E に対し GBF が BF(xyz) を含むか聞く。E がそうであるので、LBF が BF(xyz) を含むか聞く。もし、そうであればシステムに本当にファイル xyz があるか聞く。もし、あれば応答を返す。

4 実験

4.1 実験手順

今回用いる計算機と探索のために組み込んだデータの仕様を表 1 に示す。今回の実験では同一スペックの 3 台の計算機を用いた。

ディレクトリの深さが 2 レベルと 5 レベルのところに置かれたファイルをそれぞれ無作為に選んで探索し、JXTA 通信および階層型 Bloom Filter の処理にかかった時間を計測する。質問形式は 2 レベルのファイルに対しては”ディレクトリ/ファイル”であり、5 レベルのファイルに対しては”ディレクトリ 1/ファイル 1”と”ディレクトリ 2/ファイル 1”の 2 通りで探索を行う。

OS	FreeBSD4.6.2-REREASE
CPU	Pentium IV 1800MHz
メモリ	256MByte
LAN	100Mbps
言語	j2sdk1.3.1
実験に用いたデータサイズ	600MB

表 1: 実験環境

4.2 実験結果

測定結果の中から代表例としていくつかを図 3 に示す。階層型 Bloom Filter の処理にかかる時間は同じ探索については常に安定している。JXTA 通信の時間は多少ばらつきがあるが、ほぼ安定している。

深さ 2 のファイル		BF		JXTA	
ファイル		BF	JXTA	BF	JXTA
ファイル 1	1	306	271		
	2	308	369		
	3	313	284		
ファイル 2	1	167	269		
	2	169	267		
	3	167	369		
ファイル 3	1	307	460		
	2	302	426		
	3	303	273		

深さ 5 のファイル		D1/F1		D2//F1	
ファイル		BF	JXTA	BF	JXTA
ファイル 1	1	1094	262	625	343
	2	1092	274	623	344
	3	1119	267	629	193
ファイル 2	1	208	181	768	194
	2	203	194	769	194
	3	202	196	768	195
ファイル 3	1	312	192	748	190
	2	312	192	741	193
	3	311	201	747	193

BF: 階層型 Bloom Filter の処理時間, JXTA: JXTA の通信時間

図 3: 測定結果

4.3 考察

計測結果より本システムの応答時間が実用可能な範囲であると言える。また、3 台のマシン（3 つピア）の間でそれぞれが他のピアのデータへの探索を行えることから、ピア同士が”Give and Take”の関係であり、内部のピアが同等であると言える。さらに、たとえあるピアがシステムから離脱しても、残りのピアの間でシステムが正常に機能すること、その後、いつでもシステムに参加できることから、参加・離脱の自由が確認できる。ただし外部機密性については、今回は実験室内のみの実験であるため検証することは難しい。

5 結び

本論文では、Java/JXTA を用いて本提案手法を試作し、実際にその使いやすさを確認するために実験を行った。これにより、パケットのループなし、ブローキャストなしの通信の実現、ピア同士の”Give and Take”な関係、内部の同等性、参加離脱の自由を確認することができた。また、階層的 Bloom Filter が高度なファイル名探索要求に対応可能であることを示した。最後に、JXTA 自身の問題からシステムに使用しているパイプをなかなか発見することが出来ず、待たされることがあった。このような、JXTA の気ままな性質から現時点で使用するためには、時間にゆとりを持った利用法が望ましいと言える。

参考文献

- [1] Balakrishnan, et al: ”Looking Up Data in P2P System”, Comm.ACM 46-2, p.43-48,2003
- [2] Janathan Ledlie, et al: ”Scaling Filename Queries in a Large-Scale Distributed File System”, Harvard Univ. TR03-02,2002
- [3] Brendon J.Wilson, et al: ”JXTA の全て” 日経 BP 社; ISBN4-8222-8156-6(2003)