

P2P における Web サービスの検索について

2000MT081 堀 善則, 2000MT084 真田 幸利, 2000MT087 鈴木 智也

指導教員 青山 幹雄

1.はじめに

本研究では P2P アーキテクチャを用いた Web サービスの検索システムを実装し、実装したシステムの性能評価を行った。

2.研究の背景, 問題, 課題

P2P は一部に集中して使用される資源を分散し問題を解決することを期待されている。

P2P で検索を行うにはいくつかの問題が存在する。すべてのサービスを探し出すことが困難であること。全体的にネットワークトラフィックが増大すること。サービスを提供するピアがいつでもオンラインとは限らないことなどがある。

P2P アーキテクチャを用いて、以下のことに重点を置き Web サービスを検索するシステムを開発し検証した。

- (1) できるだけ多くのサービスを発見する
- (2) 少ない検索時間で発見する
- (3) リアルタイムに検索を行う

3.P2P の定義

本研究において以下の 3 つを P2P と定義する。

- (1) ピアはクライアントとサーバ両方として機能できる
- (2) すべてのコンテンツは分散している
- (3) ピアは固定 IP アドレスを必要としない

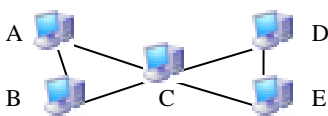
4.解決方法の提案

4.1 ネットワークの提案

P2P を用いて Web サービスを検索する際、最も重視するのは検索できるサービスの数である。このような観点から我々は 3 つの P2P ネットワークを提案する。

(1) フラットなネットワーク

最も単純なネットワーク形式である。ピアが検索を行う時には、直接接続されたピアに



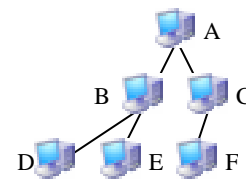
検索要求を出す。また、検索要求を受けたピアは同じように、

直接接続されたピアに検索要求を出す。このようにパケットレールのように検索要求が出され、目的のサービスを探す。

各ピア同士の通信経路は、A が検索要求を送信すると B と C に要求が伝達される。さらに、C は自分と直接つながった D と E に検索要求を伝達する。

(2) 木構造ネットワーク

図 2 に示すネットワークは、木構造をとる。各ピアは自分自身の情報を持ち、下位のピアは上位のピアに、自分の情報を一定時間ごとに伝える。その結果、ネットワークの上位に行けば行くほど、より多くのピアの情報を持つことになる。

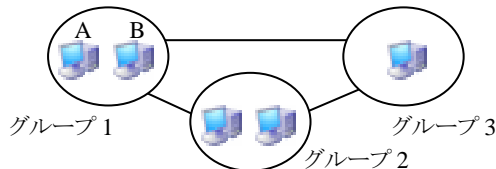


各ピア同士の通信経路は、D が検索要求を送信すると B に要求が伝達される。さらに、B は自分と直接つながった A に検索要求を伝える。同様に、E と F のような下位のピアの検索要求は上位のピアに伝達される。

(3) グループ化したネットワーク

図 3 に示すネットワークは、提供するサービスが類似したピア同士をグループ化して、グループ全体でサービスを提供する。

各ピア同士の通信経路は、A が検索要求を送信すると B に要求が伝達される。さらに、B はグループ 1 で検索要求を送信しても発見できなかった場合、グループ 2 やグループ 3 に要求を伝達する。



4.2 提案したネットワークの比較

提案した 3 つのネットワークの特徴を表 2 示す。我々はネットワーク上での多くの検索結果を少ない検索時間で得ることを目標にし、木構造のネットワークが最も適していると

考え、木構造ネットワークを構成する。ネットワークのトラフィックに関しては各ピアの帯域幅や接続時間を考慮した木構造の作成で軽減できると考える。

表1 提案したネットワークの特徴

	フラット	木構造	グループ
検索効率	×	△	○
拡張性	○	△	△
ネットワーク負荷	×	△	×

4.3 木の生成

我々は木の生成方法を2種類提案する。

(1) パイプバインディングプロトコルを用いた木の作成

JXTAのパイプバインディングプロトコルを使用して木を作成する。この提案の特徴を以下に示す。

- (i) パイプアドバタイズメントによって接続するピアを特定する
- (ii) 初期ノードに接続してきたピアの情報を格納しておくので、負荷が大きい
- (iii) 独自でルーティングを定義しなければならない

(2) ランデブープロトコルを用いた木の作成

JXTAのランデブープロトコルを使用して木を生成する。この方法の特徴を以下に示す。

- (i) ランデブーアドバタイズメントによって接続するピアを特定する
- (ii) ピアの情報を管理するピアが存在しないのでボトルネックがない
- (iii) JXTAで提供されているランデブーピアの機能を使用するので独自でルーティングを定義する必要はない

4.4 木の生成方法での提案の比較

前節での2つの方法の比較を表2に示す。ランデブープロトコルを用いて木を作成する方が、特定のピアに負荷がかかることがない。さらに、プログラムを作成する時に効率がよいので、木の作成方法は提案2のJXTAのランデブープロトコルを使用する方法を用いる。

表2 2つの提案の比較

	提案1	提案2
ネットワークのトラフィック量	×	○
初期ノードにかかる負荷	×	○
実装の容易さ	×	○

5.設計

5.1 木構造の作成

我々は木構造を以下の4つ組として定義する。

- (i) <親>は1個である

(ii) <子>は0個以上である

(iii) 優先度の高いピアほど木の上位に位置する

(iv) フェイアウォール内のピアは基本的に<葉>になる

木の生成アルゴリズムには3つの場合に分けられる。1つ目は、ネットワークに参加する時に行われる木の生成のアルゴリズムである。2つ目は、優先度の変化した時に<親>と<子>が交換するアルゴリズムである。最後に、接続先のランデブーピアがネットワークから切断したときのアルゴリズムである。木の生成時における状態の変化を図4に示す。



図4 木の作成のステートチャート図

5.2 優先度の算出のアルゴリズム

木構造を作成するために、優先度を定義する。この優先度は3つの基準に基づく。帯域幅、接続時間と処理能力である。優先度の算出方法を以下に示す。

$$\text{優先度} = A \times \text{帯域幅} + \text{接続時間} + B \times (1/\text{処理能力})$$

6.開発と実証

6.1 ピアの構成

機能を連携させた、ピアの構成を図5に示す。

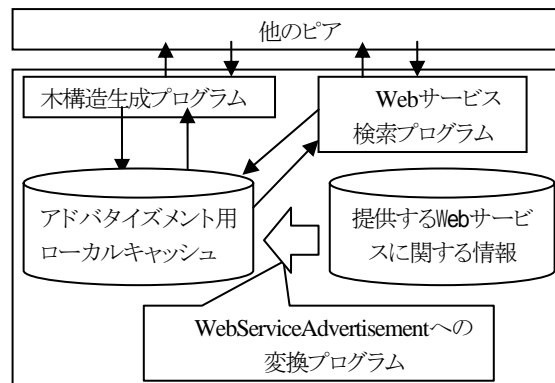


図5 ピアの構成

機能を以下に示す。

- (1) 木構造生成プログラム
- (2) Web サービス検索プログラム
- (3) アドバタイズメント用ローカルキャッシュ
- (4) 提供する Web サービスに関する情報
- (5) アドバタイズメントへの変換プログラム

6.2 P2P アーキテクチャ上の Web サービス検索システム

ピアの構成内での Web サービス検索プログラムのユースケース図を図 6 に示す。

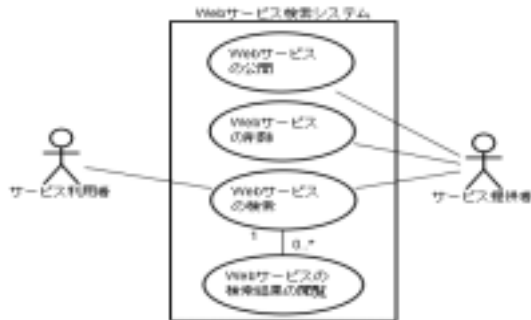


図 6 Web サービス検索システムのユースケース図

Web サービスの検索を行う際、少ない検索時間で多くの検索結果を得るため、常に最新の Web サービスの情報を得るために JXTA を利用してシステムを構築する。また、検索の効率を上げるために、ピアの優先度を算出し、その優先度を基準に接続先を決めることによって木構造を動的に作成する。

6.3 実証を行うための基準と実行環境

P2P アーキテクチャによる Web サービス検索システムによってどの程度検索の効率が向上したのか実証する。そのための基準を以下に示す。

(1) 目的の WebServiceAdvertisement の発見までの時間

発見までの時間はピアが検索要求を送信し、他のピアに検索要求が伝わり、検索の目的となる Web サービスの WebServiceAdvertisement が見つかるまでの時間である。

(2) 各ピアに送受信されたトラフィック量

システムを起動してからピア間で送受信されたメッセージの量を、ツールを用いて計測することでどれだけトラフィック量が変化したのかを調べる。

実証を行う実行環境は Pentium4 2GHz、メモリ 640MB のコンピュータ 4 台と Pentium3 500MHz、メモリ 128MB のコンピュータ 1 台で行った。また、java VM は JDK1.4.2、ライブラリは SunMicroSystems JXTA version2.1 を使用した。

6.4 実証方法

作成したシステムの実証は以下の 4 種類の環境で行う。

- (1) 木構造ネットワークでキャッシュがある場合
- (2) 木構造ネットワークでキャッシュがない場合
- (3) フラットなネットワークでキャッシュがある場合
- (4) フラットなネットワークでキャッシュがある場合

7. 実証結果と考察

7.1 目的の WebServiceAdvertisement の発見までの時間についての結果と考察

目的の WebServiceAdvertisement の発見までの総合の時間を図 7 に示す。

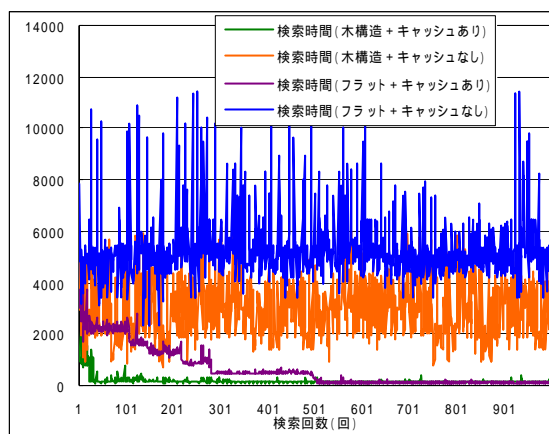


図 7 目的の WebServiceAdvertisement の発見までの時間の総合図

キャッシュのある場合とキャッシュのない場合を比べた際、どちらの場合も初めの検索時間は同程度である。キャッシュがない場合はそのまま平均して一定の値をとるが、キャッシュのある場合は発見までにかかる時間が徐々に減少していく。これは検索要求が目的のピアまで到達していきなくとも、ピア内のキャッシュで目的の WebServiceAdvertisement を発見した場合は、目的のピアまでメッセージを送らずローカルキャッシュから返答する。キャッシュは時間と共に蓄積していくので発見までの時間が徐々に減少していく。最終的には WebServiceAdvertisement が検索をするピアにすべて集まり、ローカル内だけで検索が行われるので検索の時間が安定する。

木構造の場合とフラットなネットワークの場合を比べた際、検索を始めた時点で発見時間が木構造の場合が速くなっている。これは検索要求 1 回に対して集められる WebServiceAdvertisement の量がフラットなネットワークに比べ多いので、目的の WebServiceAdvertisement を見つけられる確率が高く、結果として検索時間が短くなったと考えられる。また多くのアドバタイズメントを集められるという考え

は、フラットなネットワークでキャッシュがある場合、検索時間が減少しているのに比べ、木構造でキャッシュがある場合の方が、速く一定の値に収束していることからわかる。

7.2 各ピアに送受信されたトラフィック量についての考察

各ピアに送受信されたトラフィック量の変化を図 8 に示す。

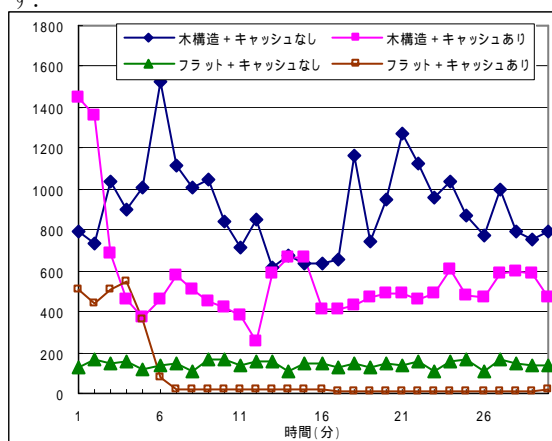


図 8 各ピアに送受信されたトラフィック量の変化の総合図

キャッシュのある場合とキャッシュのない場合を比べた際、キャッシュのある場合の方が初めは多くのトラフィックが発生する。これはキャッシュがある場合は検索をしているピアだけではなく他のピアのキャッシュも溜まるため、ネットワークに流れるトラフィックの量がキャッシュのある場合と比べ初めは多くなっている。ピア内に目的のアドバタイズメントがあった場合はそのピアより先にメッセージを流さなくなるのでトラフィックの量が減少を続け、最終的にすべてのアドバタイズメントが検索を行うピアのキャッシュの中に集まる。

フラットなネットワークの場合はローカルキャッシュの中で発見し検索要求メッセージを送信しなくなるので、トラフィック量はほぼ 0 に収束する。しかし、木構造の場合は 0 にまで減少していない。これは木構造を作るために一定時間ごとに上位のピアに自身の PriorityAdvertisement を送信するためや、優先度の変化によりピアが接続先を変更する処理を行っているためである。そのため全体的なトラフィック量は木構造を作る方が多くのトラフィックが発生している。

今度は同じキャッシュを用いた場合でも木構造を作る場合とフラットなネットワークの場合を比べた際、木構造を作る場合の方が約 2 分早く一定の値に収束している。これは木構造を用いた方が一度の検索でより多くのアドバタイズメントが帰ってくるので、すべてのキャッシュが集まるのが早いためであると考えられる。木構造の場合のほうが一度に多くのアドバタイズメントが帰ってくるという考えは、もう一つの評価基準である目的の WebServiceAdvertisement 発見までの時間のグラフが、フラットな場合より木構造の場合の方が

全体的に少ない時間になっていることから予想できる。また一度の検索で多くのアドバタイズメントが帰ってくることは、フラットな場合に比べ木構造の場合の方が全体のトラフィックの増大につながっていると考えられる。

7.3 2つの結果のまとめ

木構造を用いた方が、フラットなネットワークよりも目的のアドバタイズメントの検索時間が早くなるが、全体的なトラフィックの量は多い。今回は1つのピアが検索を行ったが実際は全てのピアが検索を行うので、複数のピアによる検索要求のトラフィックを考慮に入れるとフラットなネットワークより木構造を用いた方がトラフィックは減少すると考える。

また、キャッシュを用いることによってネットワークに流れる多くのトラフィックの量を減少させることができる。キャッシュを用いると情報が過去の情報になるが、キャッシュを一定時間ごとに変更することで解消できると考える。木構造では一定の時間ごとに情報を上位のピアに送信してキャッシュが更新されているので解消できると考える。

8.まとめ

本研究では P2P アーキテクチャによる Web サービスの検索を行い、ネットワークの構成がフラットな場合と木構造を用いた場合についてのアドバタイズメントの検索速度やトラフィック量を評価の対象とした。

P2P アーキテクチャを用いることによって、C/S アーキテクチャでは得られなかった Web サービスをリアルタイムに検索し、最新なサービスを得ることが可能になる。また、現在の UDDI を用いる Web サービスアーキテクチャでは、UDDI レジストリにサービスの情報を集中して格納しているため、アクセスが集中しサーバに負荷が大きいといえる。しかし、P2P アーキテクチャを用いることによってサービスの情報を各ピアに分散して格納しているため、従来の C/S アーキテクチャのようにアクセスが一部に集中することがないので、負荷が分散する。

本研究は Web サービスの検索を検証したが、ピアの増大によるトラフィックの増大傾向やキャッシュの更新の頻度をどの程度にすると最適になるかなどは今回の実験では検証できなかった。またサービスを使う際の信頼性やセキュリティが今後の課題である。

参考文献

- [1] マイケル・ミラー, トップスタジオ訳:P2P コンピューティング入門, (株)大和総研情報技術研究所 監修
- [2] B. J. Wilson, 倉骨 彰, 佐野 元之 訳:JXTA のすべて, 日経 BP 社(2003)
- [3] TeamJXTA, ProjectJXTA, <http://www.jxta.org/>(2001)