

適応型ピアツーピアシステムにおける動的検索誘導 Dynamic Search Redirection in Adaptive P2P System

佐藤 崇[†]
Takashi Sato

内田 良隆[‡]
Yoshitaka Uchida

吉田 紀彦[†]
Norihiko Yoshida

1. はじめに

現在インターネットの主流であるクライアント・サーバシステムは、サーバにアクセスが集中するという問題を抱えている。これを改善するものとして注目されているのが、コンテンツを各ノードに分散させる P2P(Peer to Peer) システムである。P2P を具現化したシステムとして Napster[1] や Gnutella[2] などが有名である。

P2P ではコンテンツの所在を検索することが重要である。Napster では所在を一元管理するインデクスサーバを置いているが、そのサーバへのアクセス集中を招いている。Gnutella では問合せをブロードキャストすることで所在検索を行っているが、ネットワークトラフィックの増大を招いている。そこで当研究室では必要に応じてインデクスサーバを動的に生成し、適宜再配置することで、負荷一局集中・トラフィック増大をともに回避する P2P システム「AmorphicNet」[3],[4] を設計し、小規模な実験でその有用性を実証した。しかし時間経過に伴いサーバ負担が高まる点や、サーバ生成時のネットワーク負荷がさほど減少しないなどの問題点があった。

そこで本研究では、それらの問題を解決するため、サーバ生成時に広範囲にわたりサーバ方向へと検索パケットを誘導する方式を提案し、シミュレータを作成してその効果を実証した。

2. 背景と関連システム事例

P2P はコンテンツ管理の面から二つのグループに分類できる。一つはコンテンツの制御から行う方式であり、CAN[5] や FreeNet[6] などがある。もう一つはコンテンツの所在(インデクス情報)のみを制御する方式である。前者の方式は、自分のコンピュータの中に必要としないコンテンツが配置されるが、後者の方式では自分の要求するコンテンツのみが配置されるようになるため、当研究室は後者の方式を採用した。現在、後者の方式には以下の二つの有名な方式がある。

(1)HybridP2P

Napster で導入された方式で、コンテンツの所在すなわちインデクス情報を一元的に保持するサーバを中央に置く。各ノードはそのサーバに問合せを行う。この方式では、検索数が多くなるとサーバの負荷が大きくなるというクライアント・サーバシステムと同じ問題が発生する。これを解決する試みとしては、インデクスサーバのミラーサーバを置く FastTrack[7] があり、KaZaA[8] などに用いられている。

(2)PureP2P

Gnutella で導入された方式で、全てのノードが対等に接続されている。各ノードは自身に接続している全ノードに問合せを送る。この方式では、ブロードキャストに

よって大きなトラフィックが発生する。トラフィック削減のため問合せやコンテンツの転送を各ノードでキャッシュする方式もある。

当研究室が提案した AmorphicNet とは、これらの技術に基づき初期状態を PureP2P で構成し、負荷が高まったところにインデクスサーバを動的に生成・配置するシステムである。

サーバは負荷が高まると子サーバを生成することにより負荷を分散させる。逆に負荷が低くなると平ノードと名付けたサーバでない元のノードへと自動的に戻る。

このように AmorphicNet では、動的にサーバを生成・配置し、負荷が低くなれば平ノードに戻ることににより、常に最適なネットワーク構造を保持することを目的としている。

しかし、時間経過に伴いサーバ負荷が高まる点や、サーバを生成しても初期段階ではネットワーク負荷がそれほど減少しないといった問題点もある。

3. インデクスサーバへの検索パケット誘導

提案方式の目的を明確にするために、現在の方式を述べ、その問題点を挙げる。

3.1 現在の方式

現在の方式では、生成されたサーバが近隣ノードに自分の IP アドレスを知らせる。サーバの IP アドレスを知らないノードの検索パケットがサーバに届いた場合、サーバは検索結果と共に自身の IP アドレスを知らせるパケットをそのノードに送る。検索結果と IP アドレスを受け取ったノードは、次回以降直接サーバに検索パケットを送るようになる。しかしそのような方式をとると、サーバの IP アドレスを知るノードが急激に増えるためサーバの負荷も急激に高まる。サーバは負荷の高まりに合わせ子サーバを生成し、自分は親サーバとなる。親サーバと子サーバの間ではインデクス情報の同期が取られるため、インデクス情報の更新が頻繁になると同期を保つための負荷が高くなっていく。時間経過に伴いサーバの IP アドレスを知るノードが増えるため、サーバは検索パケットに回答する負荷に加え、インデクス情報の同期を保つための負荷も高まっていく。

また、インデクス情報のキャッシュはサーバのみに溜るため、サーバ退出後それらのキャッシュは消え、元の PureP2P で構成された状態に戻りネットワーク負荷が再び高まってしまふ。

3.2 提案方式の概要

本研究が提案する方式は、サーバ方向へ検索パケットを誘導するというものである。サーバが生成した時、そのサーバは自身の IP アドレスを添付した TTL の小さいパケットを隣接するノードへと送信する。そのため、サーバの IP アドレスを知るノードはそのサーバの近隣ノードのみであり、その数は時間が経過しても増えない

[†]埼玉大学 Saitama University

[‡](株) NTT ドコモ NTT Docomo

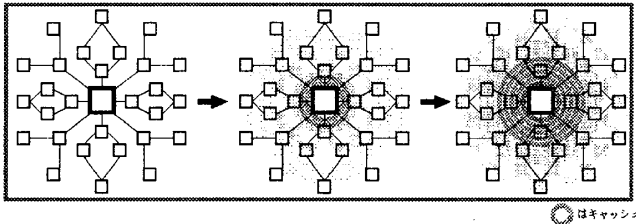


図1: 時間経過に伴うキャッシュ範囲の移行

ものとする。それと共に、サーバは自身の方向を広範囲に知らせるため TTL の大きいパケット送信する。それにより、サーバの IP アドレスを知らないノードは二つのグループに分けられる。一つ目は、サーバに関し何の情報も持たないノード群である。これらは PureP2P の動作を行う。二つ目は、サーバの方向を知らせるパケットを受け取ったノード群である。あるノードが検索パケットを送ろうとした場合、サーバ方向にある一つのノードにだけ検索パケットを送る。よって PureP2P と比べてネットワークトラフィックが削減される。サーバ方向へと送られた検索パケットはサーバのもとへと一つ一つ近付きながら転送され、最終的にはサーバにたどり着く。サーバには膨大なインデクス情報が保持されているため、要求していたコンテンツの所在が見付かる可能性が高い。

3.3 詳細設計

3.2 節で挙げた方式を実現するための具体的な動作を詳述する。

(1) 誘導テーブルの作成

インデクスサーバ生成時にサーバは広範囲にわたり自分の方向を知らせるパケットを隣接ノードに送信する。パケットを受け取ったノードは送られてきたノードの IP アドレスとそのパケットのホップ数を誘導テーブルと名付けたテーブルに登録する。ここでいうホップ数とはオーバーレイネットワーク上でのホップ数のことである。

平ノードは、パケットを送ってきたノード以外の隣接ノードにホップ数を一つ足しパケットを送る。それをホップ数が一定値以上になるまで続ける。これによりサーバを中心とした同心円上にサーバ方向へとパケットを誘導する経路ができあがる。

(2) 誘導テーブルの更新

インデクスサーバがいくつも生成された時、各ノードにさまざまなところからサーバ方向へ検索パケットを誘導するための情報が集まる。そのため一定の条件で誘導テーブルを作成していく必要がある。以下にその条件を示す。

1. 誘導テーブルに何も登録されていなければ情報を登録する。
2. 既に登録されている情報のホップ数のほうが大きければ、その情報を破棄して新しい情報を登録する。
3. 既に登録されている情報のホップ数と等しければ現在の情報を保持したまま新しい情報を追加する。
4. 既に登録されている情報のホップ数のほうが小さければ新しい情報を破棄する。

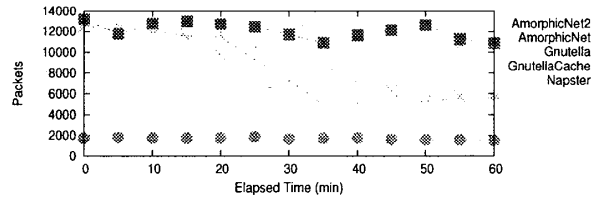


図2: ネットワーク内の総パケットの推移

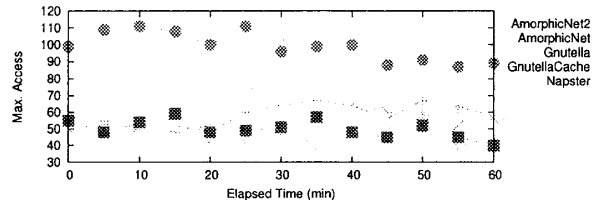


図3: ネットワーク内のノードの最大検索パケット処理数

(3) 検索パケット転送

誘導テーブルを持っているノードに検索パケットが送られてきた場合、そのノードは自分がそのコンテンツに関するインデクス情報を持っていない場合は誘導テーブルより IP アドレスを引きそこに検索パケットを送信する。

以上の作業を行うことで、誘導テーブルを持つノードはサーバに直接問合せるのではなく、サーバに近いノードのみに検索パケットを送信するようになる。それにより、以下のような利点が得られる。

- 広範囲のノードがサーバに近いノードのみに検索パケットを送るので、現在の方式より PureP2P で構成された部分が削減されネットワーク負荷が下がる。
- 時間が経過するにつれて図1のように、サーバを中心とした同心円上にキャッシュが溜っていくため、サーバに到達する前に検索パケットに回答できる割合が増え、サーバの負荷が低下する。
- サーバが退出した場合においても、サーバ付近のノードにキャッシュが残っているため現在の方式よりサーバ退出時のコストが低下する。

4. 実装と評価

本研究では AmorphicNet の実装コードを改良しシミュレータを作成した。シミュレータではノード数 300 個、初期状態で保持しているコンテンツ数を 50 個以内のランダムな数で与え、サーバ格上げの契機をキャッシュ数 200 個、コンテンツ数 676 個、近隣ノードとの接続数 2 ないし 3 として実験を行った。検索パケットはネットワーク内でランダムに選ばれたノードが 0.1 秒間隔で送信することとする。比較対象は現在の AmorphicNet の動作を行うものと、Napster、Gnutella、キャッシュ付きで動作する Gnutella の合計 4 つである。

図2と図3はそれぞれ毎分あたりのネットワーク内の総パケット数の推移とノードの最大検索パケット処理数の推移を示している。シミュレータでの AmorphicNet

の動作において、サーバの生成が始まるのは20分～30分後である。図においては提案方式をAmorphicNet2と表示している。

図2よりサーバの生成が始まってから提案方式の方が現在のAmorphicNetよりネットワーク全体の負荷が低下していることがわかる。これは以前PureP2Pで構成されていた部分が削減されたためだと考えられる。また図3より、サーバが生成し始めた頃の負荷が現在のAmorphicNetより低くなっていることがわかる。これは現在のAmorphicNetは、サーバが少数の場合、そのサーバにのみ検索パケットが集中してしまうのに対し、提案方式ではサーバが少数の場合でも、検索パケットは一定の経路を辿っていくので、その過程で検索コンテンツが見つかればサーバに到達する前に動作が完結する。そのためサーバに対して高負荷がかかることが無い。

5. まとめ

本論文では、AmorphicNetにおいてネットワーク負荷、サーバ負荷を低下させるために、サーバへと検索パケットを誘導する方式を提案した。実験の結果、サーバ生成時にPureP2Pで構成された部分を削減することができ、その結果、ネットワークの負荷が減少した。さらにサーバに直接問合せるノードを削減することができ、ネットワーク内でのサーバ負荷が減少した。

なお、今回の実験ではノード数・コンテンツ数が限られていたため時間経過に伴い現在の方式との差が無くなっている。しかしノード数・コンテンツ数を増やすことにより、現在と提案方式との差が顕著になると考えられる。その理由はコンテンツ数が少ないと、時間経過に伴い両方の方式とも、各ノードがほとんどのコンテンツを持っている状態になってしまうためである。そのような状態になると、検索パケットを隣接ノードに転送する回数が減り、提案した方式の効果が顕著に現れない。またノード数が少ないと、検索パケットの到達する範囲に限りがあるため、PureP2Pで構成された部分の削減、ならびにサーバにかかる負荷の比較がしづらい。

今後はサーバ同士の連携をとり、サーバ間での検索パケットの転送を行うことや、コンテンツ自体に着目し、インデクス情報だけでなくコンテンツの分散管理に関する手法も取り入れることを検討している。

参考文献

- [1] Napster website, "http://www.napster.com/".
- [2] Gnutella website, "http://www.gnutella.com/".
- [3] 吉田紀彦, 内田良隆, 楯崎修二, 瀬河淳一, 下川俊彦. インデクスサーバの自動形成によるピアツーピアシステムの動的効率化. 電子情報通信学会論文誌, Vol.J86-B, No.8, pp.1445-1453, 2003.
- [4] 内田良隆. ピア・ツー・ピア・システムにおけるインデクス情報管理の動的再構成. 長崎大学学士論文, 2002.
- [5] CAN website, "http://www.icir.org/sylvia/".
- [6] Brandon Wiley Ian Clarke, Oskar Sandberg and Theodore W. Hong. Freenet: A distributed anonymous information storage and retrieval system. *Lecture Notes in Computer Science, vol.2009, pp.46-67, Springer, 2001.*
- [7] FastTrack website, "http://www.fasttrack.nu/".
- [8] KaZaA website, "http://www.kazaa.com/".
- [9] 佐藤崇. 適応型ピアツーピアシステムにおける動的検索誘導. 埼玉大学卒業論文, 2004.

