

P2P 技術を用いた効率的な電子掲示板システムの提案

櫻木 信輔[†] 山之上 卓^{††}

[†]鹿児島大学大学院理工学研究科情報工学専攻

^{††}鹿児島大学学術情報基盤センター

概要 P2P 型電子掲示板システム(BBS)は C/S 型に比べコストや匿名性という面において優れている。しかしその反面、ノード管理や書き込まれたデータの保持方法など P2P 特有の問題が生じる。また、BBS として不都合無く運用する為には書き込み順を前後させない機構を持つ事が望ましい。そこで本稿ではこれらの点を解決した、BBS に適した P2P システムを提案する。

A Proposal of an Effective Bulletin Board System using a P2P Technology

Shinsuke SAKURAGI[†] Takashi YAMANOUE^{††}

[†] Graduate School of Science and Engineering, Kagoshima University

^{††} Computing and Communications Center, Kagoshima University

Abstract A P2P-based Bulletin Board System (BBS) is more efficient than client/server model's one in terms of cost and anonymity. But, P2P-based BBSs have several problems such like node management and data maintaining etc. Moreover, the P2P based BBS is hard to have the mechanism that the unique serial number is given to an article in a thread as same as a common client/server BBS. We propose a P2P system witch solves such problems.

1. はじめに

現在、電子掲示板システム(BBS)は C/S 型が主流となっている。C/S 型 BBS は構造が単純であり管理が容易であるという点において優れているものの、その構造上どうしても避けられない問題を持っている。それは、ネットワーク負荷や保存容量、CPU 速度といった物理的な限界が存在するという点と、それによって管理コストが非常に大きなものとなっているという点である。また、管理者が存在するという点も、情報の中立性という観点から見た場合、好ましいとは言えない。

そこで本研究では、P2P 技術を用いることによって負荷分散、効率化、柔軟性の確保などを実現し、物理的・金銭的コストの軽減を図ると共に、管理者の特権的機能についてもこれを排除することによって編集の介入を防止するという事を設計の理念とした。また、書き込み順の同期という従来の P2P 型 BBS ではあまり考えられていなかった問題についても解決方法を示す。

2. 既存システム

本システムを提案したのは、現在使われている P2P 型 BBS では C/S 型 BBS の代替として使おうとした時に、デメリットの部分が無視できないのではないかと考えたからである。現在、一定数以上ユーザ数を獲得していると思われるシステムに、Winny2 BBS, Alpha2(RiGOch), 新月などが存在するが、ここではこれらについてごく簡単に紹介する。なお、これらの他にも Freenet(Frost)などが存在するが、運用目的が異なる為ここでは割愛する。

・ Winny2 BBS[1]

スレッド(話題)ごとに管理ノードを持たせることにより、同期問題を解決している。しかし、管理ノードがネットワークから抜けると該当スレッドは閲覧不可能となる、スレッドの認識できる範囲が Winny のブロードキャストが到達する範囲に限られる、などの問題がある。

・ Alpha2(RinGOch)[2]

キャッシュにある種の思考ルーチンを持たせることにより効率的にファイルを分散させるシステム。BBS, Wiki, チャットといった機能を実装している。ただし汎用性に重点を置いている為、書き込み反映までの遅延が大きく書き込み順の同期も取れないなど、純粋に BBS として利用するにはやや不都合な点がある。

・ 新月[3][4][5]

2008 年 8 月現在、最も活発に開発されているシステムのひとつ。常時接続のコンピュータ同士を接続し、ブロードキャストによってデータの共有や負荷分散を行う。書き込み順の同期については考えられていない。

3. 提案手法

本章では、本システムの構成や通信に関する処理、既存の問題の解決方法等について述べる。

3.1 ネットワークトポロジ

本システムでは 3 種類のノード群を持たせ、それらをリング型と二分木型に分けるハイブリッド型とした。それぞれのノード群の役割を以下に述べる。また、これらの関係を図 1 に示す。

・ 管理ノード

検索、認証、ノードの参加や離脱に関する処理など、システムの管理機能をデータ保持ノード群に対して提供する。管理ノード 1 つにつき、1 グループのデータ保持ノード群を管理する。管理ノード同士はリング型構造で結びついており、お互いに自身が管理するカテゴリ名と自身の位置情報を結びつけたデータを交換する。リング型構造を取っているのはトラブルが発生した場合の再接続処理が簡易であり、また交換するデータが頻繁に変わる性格のものでは無いためである。

・ データ保持ノード

カテゴリ毎に本ノードが 1 つのグループを構

成し、本システムにおいてストレージとしての機能を持つ。本システムの利用者は基本的に本ノードとなる。ノード同士は二分木型構造で結びついており、同一カテゴリ内の全スレッドのデータをお互いに共有し合う。ここで二分木型を採用したのは多数の packets を一斉にやりとりする場合において効率的である為で、この事は平原・山之上らの研究[6]によって示されている。

・一般ノード

P2P ネットワークには参加せず、閲覧や書き込みのみを行うゲストノード。データ保持ノードを Proxy として利用する。データ保持ノードが自身の属するカテゴリのスレッドを放送のように随時受け取ることが出来るのに対し、このノードは必要に応じて取得要求を出す必要がある。

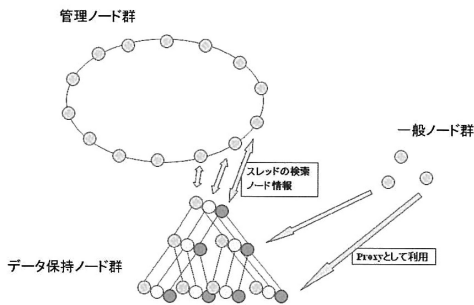


図1 ネットワークトポロジ

3.2 通信・検索アルゴリズム

利用者(データ保持ノード)がスレッドを閲覧する際には、2つのケースが考えられる。それは、自身の属するカテゴリ内のスレッドを閲覧する場合と、それ以外のスレッドを閲覧する場合である。前者の場合は単純に自身の持つキャッシュを閲覧すれば良いのだが、後者の場合は該当のスレッドを保持するノードを検索し、取得要求を行う必要がある。ここでは、この時の手順について説明を行う。

前提として、管理ノードは管理下のデータ保持ノード群の位置情報を保持しており、また他

の管理ノードの位置情報とそれらが保持・管理するボード名の情報を持っている。

一連の流れを図2に示す。まず、利用者は自身の管理ノードからボードリストを取得する。利用者はこのリストの中から閲覧したいボードを選択し、管理ノードにスレッドリストの取得要求を出す。次に、それを受けた管理ノードは該当のボードを管理するノードに対してこの要求を橋渡しする。すると、要求を受けた管理ノードは要求元の利用者に対してランダムに選択した自身の管理するデータ保持ノードの位置情報を返す。これを受けた利用者は、このデータ保持ノードに対して接続要求を行う事によってスレッドリストを得る。また、該当ボードおよびカテゴリ内におけるスレッド本文の取得や書き込みに際してもこのノードを利用する。

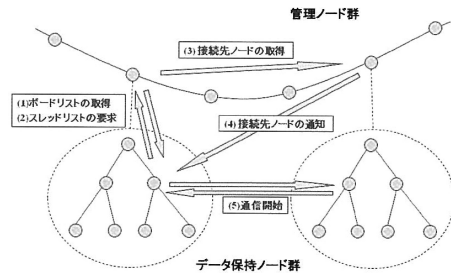


図2 通信・検索イメージ

3.3 書き込み

3.3.1 書き込みの伝搬

利用者が書き込みを行う際には、スレッドを閲覧する場合と同様に、対象が自身の属するカテゴリ内のスレッドかそれ以外という2つのケースが考えられる。しかし、後者の場合でも書き込みを行うのは既に3.2の手順により経路が確立した後であるため、異なるのは書き込み要求が自分自身から発生したものなのか他のノードから取り次いだものなのかという程度である。書き込みを実際に行おうとすると、まず利用者は自身もしくは接続先のデータ保持ノード内のスレッドデータを書き換え、次に書き込みデータを二分木型のネットワーク内にパ

ケツリレーのように伝搬させる。これによって、最悪でもノード数の対数の2倍程度の遅延のうちに同一グループ内の全ノードに書き込みが反映される。

3.3.1 匿名性

書き込みデータを配信する際、その情報の第一発信者が誰であるのかを明示しないため、他のノードは誰が書き込んだデータなのかを知る事が出来ない。また、一般ノードや他のデータ保持ノードが間接的に書き込む場合もあるため、より複雑性が増し、この経路を暗号化した場合情報の第一発信者の特定はより困難となると考えられる。

3.3.2 処理の流れと同期問題

書き込みを行うにあたり、データ保持ノードは最初に自身の所属する管理ノードに対して書き込み要求を行う。ここで、管理ノードはグループ内のスレッドの書き込み数を記憶しており、書き込み要求に対してスレッド毎に排他制御を行った上で番号を割り振る。よって、要求元のノードはユニークな書き込み番号を得る事が出来るため、書き込み番号の衝突や前後の入れ替わりは起こらない。

これは、書き込み番号のみを同期させ、書き込み本文についてはブロードキャストが到達し次第、部分ごとに更新するというイメージである。この仕組みによって、断片的にしかスレッド本文が読めない場合でも、書き込み番号を明示してレスポンスをする事が可能となる。

4. 性能評価

本章では、本システムの負荷分散の性能および書き込み反映までの遅延について考察する。

4.1 負荷分散

ネットワークの負荷について、C/S型BBSとの比較を通して本システムの有効性を検証する。今回は比較対象として総合掲示板サイト「2ちゃんねる掲示板[7]」を用いた。

まず、同サイトのネットワーク負荷について

推定を行う。表1に同サイトの概要を、図3に同サイトのトラフィックの推移[8]を、図4に同サイトのPV数の推移[9]を示す。なお、アクセス数・書き込み数・新規スレッド数は2008年8月22日の、転送量およびPV数は2008年8月23日15時まで1週間分のデータである。また、推定利用者数は、ネットレイティングスによる2007年2月時点での調査数[10]であり、2008年8月現在と当時のPV数がほぼ同じである[9]ことから推定を行った。

表1 「2ちゃんねる掲示板」の概要

アクセス数[9]	190,949,191 PV/日
書き込み数[11]	2,932,498 件/日
新規スレッド数[11]	22,497 スレッド/日
推定利用者数	930 万人
サーバ数[12]	57 台
平均転送量[8]	282.668Mbps
最大転送量[8]	598.394Mbps

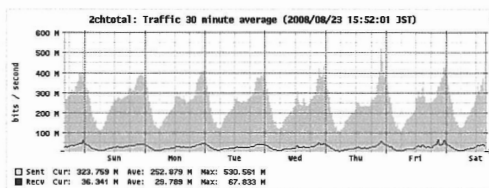


図3 「2ちゃんねる掲示板」のトラフィック推移

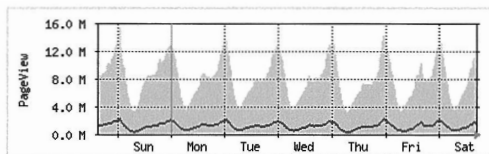


図4 「2ちゃんねる掲示板」のPV数推移

ここで、図3と図4から、トラフィックとPVには強い相関関係が認められる。よって、次の式から1PVあたりのおおよその転送量を求める事が出来る。なお、1スレッドあたり1000件まで書き込めるという同サイトの制限から、書き込み1件あたりのコストは1PVあたりのコストの500分の1程度と仮定した。

$$282.668 \times 1024 \times 1024 \times 60 \times 60 \times 24 \div$$

$$(190949191 + 2932498 \div 500) \div 8 \text{bit} \div 16764 \text{Bytes}$$

よって、同サイトの 1PV あたりの転送量は 16764Bytes、書き込み 1 件あたりの転送量は 34Bytes と仮定することが出来る。

次に、本システムを同サイトと同規模の運用をする場合について、ネットワーク負荷の推定を行う。ここで、システムの構成は次のように仮定した。また、各種転送コストの仮定は表 2 に示す。

- ✓ アクセス数・書き込み数・新規スレッド数は「2ちゃんねる掲示板」と同値
- ✓ 管理ノードは 57 台
- ✓ 利用者の 2 割がデータ保持ノードとして参加し、PV の 8 割を占める
- ✓ ユーザは所属カテゴリとその他のカテゴリを 5 割ずつの頻度で閲覧し、平均 3 カテゴリ利用する
- ✓ スレッドは 1 カテゴリにつき 1 万件保持
- ✓ スレッドの平均書き込み数は 500
- ✓ データ保持ノードは 1 日に 1 時間参加
- ✓ 通信障害は発生しない
- ✓ 全カテゴリは同頻度で利用される

表 2 転送コストの仮定値

書き込み 1 件あたりデータ量	34Bytes
スレッドタイトルの長さ平均	50Bytes
ノードの位置情報	6Bytes
書き込み時の認証	10Bytes
ノードの参加および離脱	20Bytes
ノード間の生存確認	100Bytes/sec

これより、本システムにかかるトラフィックはおおよそ次のように推定することが出来る。なお、データ保持ノード間のノード参加および離脱に係るコストなど、システム全体から見て無視できるほど小さいものはここでは考えない事とする。

管理ノードの総転送量：1738bps

管理ノード間の生存確認：1600bps

データ保持ノード管理：60bps

他カテゴリの検索：30bps

書き込みの認証：48bps

データ保持ノードの総転送量：4376bps+ α

データ保持ノード間の生存確認：2400bps

一般ノード(Proxy)間の通信：400bps

書き込みのプロロードキャスト：488bps

他カテゴリ間の検索(含 Proxy)：928bps

他カテゴリ間の閲覧(含 Proxy)：160bps

※ α は参加時のスレッド取得分 (1.75MByte)

このことから、本システムは C/S 型 BBS に比べて非常にネットワーク負荷が小さく、負荷分散の性能について、十分有用であると評価出来る。なお、これらの条件下で本システムを動作させるには、170MBytes 程度のキャッシュ容量が必要とも推定できる。

4.2 書き込み反映までの遅延

あるノードが書き込みを行ってからシステム全体にそれが反映されるまでの遅延を計算する。

本システムにおいて書き込みデータを管理しているデータ保持ノードは二分木型構造である。ここで 4.1 と同様の仮定を利用すると、データ保持ノード 1 グループあたりのノード数は平均 1360 となる。つまり、ツリーの高さは $\log_2 n$ より 11 となり、遅延は最悪の場合でも 22 ノード分の経由時間で済むと推定できる。

5 まとめと今後の課題

今回設計したシステムでは、書き込み順の同期をとりつつ、特定のノードに高い負荷をかけずに大規模な運営を行える可能性がある事が確認できた。また、一般ノードを用意し多数がそれを利用することを想定する事によってシステム参加への敷居を下げるといった配慮も行った。匿名性についてもある程度確保出来ると考えられる。

ただし、今回の論文では、CPU コストの測定、通信のオーバーヘッドの測定、一時的に高負荷

になる場合についての言及がなされていない。
今後はプログラムの完成を急ぎ、これらについても検証を行っていく予定である。

参考文献

- [1] GLOCOM: “「Winny の技術」をもとに当時の到達点を明らかにする”, http://www.glocom.ac.jp/j/chijo/text/2006/06/winny_glocom.html .
- [2] RinGOch, <http://gateway.ringoch.info/> .
- [3] 新月, <http://shingetsu.info/> .
- [4] shinGETsu,
<http://sourceforge.net/projects/shingetsu/> .
- [5] 第 1 回 P2P 勉強会: “P2P 掲示板「新月」のプロトコルとデータ構造”, <http://www.skame.nu/P2Pmeeting/0409fukutommy.pdf> .
- [6] 平原貴行, 山之上卓, 安在弘幸, 有田五次郎: “TCP を利用した分散ネットワーク環境のための電子黒板システム”, 情報処理学会論文誌, Vol.43, No.1, pp.176-184(2002).
- [7] 2ちゃんねる掲示板, <http://www.2ch.net/> .
- [8] 2ちゃんねるの転送量,
<http://server.maido3.com/pie/> , 23 Aug 2008.
- [9] 2ちゃんねる ページビュー観測所,
<http://pv.40.kg/> , 23 Aug 2008.
- [10] ネットレイティングス: “YouTube、“史上最速”で利用者 1000 万人に到達”, http://csp.netratings.co.jp/nnr/PDF/Newsrelease03222007_J.pdf .
- [11] スレッドランキング, <http://www.bbsnews.jp/> .
- [12] 2ちゃんねる/BBSPINK 稼働中の掲示板システム関連サーバー一覧, http://mumumu.mu/server_working/ .