

NetNewsのためのキャッシュシステム

舟阪淳一 最所圭三 福田晃
奈良先端科学技術大学院大学

概要

NetNewsの配送記事量が増加してきており、1日20Gbytesに達することもある。そのため記事の保存および配送には大量のディスクおよび回線資源が必要となってきた。一方、読まれている記事が全記事の10%に満たないサーバが存在することが分かっている。従来のNetNewsシステムは全記事をコピーするので、90%が無駄な配送になる可能性がある。そこで本稿では、資源の効率的な利用のため選択的コピーを提案する。この際サーバをキャッシュシステムに置き換え、NetNewsのアクセスパターンを考慮したキャッシュアルゴリズムを調査する。実システムで実験を行うと、再現性が乏しくアルゴリズムの比較に向かないので、シミュレーション実験を行った。その結果、できるだけユーザの利便性を損わずに資源を節約するには、プリフェッチにおいてニュースグループによる選択をすればよいことが分かった。

Cache system for NetNews

Junichi Funasaka, Keizo Saisho, Akira Fukuda
Nara Institute of Science and Technology

Abstract

NetNews traffic is increasing up to 20Gbytes per day. Huge amount of disk storage and high speed network line are required. It is, however, investigated that read articles are 10% of all articles on some NetNews servers. Since traditional NetNews systems copy all articles, 90% of all articles may waste resources. In this paper, a selective delivering system is proposed. In the system, a NetNews server is replaced with a cache system. Caching algorithms, that take account of access patterns, are researched. Because reappearance is hard to observe on actual systems, experiment was carried out with simulation. As a result, prefetch selecting newsgroups uses resources effectively and makes users' usability not so bad.

1 はじめに

利用者の増加により、インターネットのトラフィックが増大してきている。NetNewsの場合、サーバ間で転送される記事の量が1日20Gbytes

に達する日もある¹(図1参照)。このため全ての記事を保存し配送するためには、膨大なディスク資源と広帯域の回線が必要となる。一方で、

¹news:tnp.netnews.stats

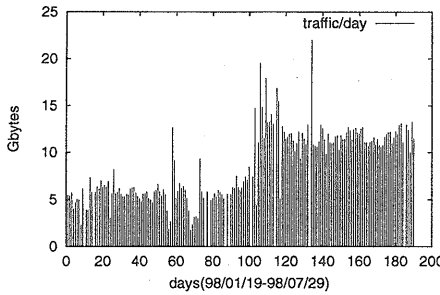


図 1: Tokyonet における 1 日あたりの配送記事量

NetNews の全記事が読まれるとは限らない。例えば奈良先端科学技術大学院大学のニュースサーバで実際に読まれた記事を調べた結果、約 10% しか利用されていないことが分かった [1][2]。ここで問題となるのは、利用されない記事を配送することによる資源の無駄使いである。これは NetNews システムがニュースサーバ間で、互いに自分が持っていない記事を全てコピーすることが原因であると考えられる。

そこで本稿では、利用される記事を選択しできるだけ資源を有効利用できる NetNews システムを提案する。ニュースサーバをキャッシュシステムに置き換えることにより、ディスクの利用効率とネットワークの利用効率を向上させることを考える。ユーザの利便性が損われては意味がないので、できるだけユーザの利便性を損わずにディスク資源とネットワーク資源を節約する キャッシュシステムの実現を目指す。

先述したように実際に読まれている記事の割合は 10% 程度なので、この局所性を有効に利用できれば、ユーザの利便性を保ったまま資源の節約が期待できる。

本稿では特に記事の選択に注目したキャッシュアルゴリズムをいくつかとりあげ、シミュレーションにより評価した。NetNews においてどのような記事選択基準が、ディスクおよびネットワーク資源を効率的に利用できるかを調査した。

2 NetNews

NetNews はグループ通信のためのシステムである。記事が情報の単位であり、ユーザの間でやりとりされる。NetNews のサーバは自分が持っていない記事を他のサーバと互いにコピーすることで、同じ記事を保持するようにする。また NetNews のクライアントはサーバに接続し、記事の閲覧や投稿を行う。ある記事はクライアントからあるサーバに投稿され、サーバ間でバケツリレーのようにコピーが繰り返され、最終的に世界中のサーバで閲覧が可能になる。

このように NetNews システムでは、サーバ間がプッシュ型の配送であり、サーバ・クライアント間がプル型の配送である。

プッシュ型配送とは、情報提供側主導で情報を配送することである。たとえばサーバに新規に発生した情報を発生時に送る場合が相当する。一方プル型配送とは、情報取得側主導で情報を配送することである。たとえばクライアントが要求を出して情報を取得する場合である。

従来、サーバ間はすべてプッシュ型で配送されていたが、配置によってはプル型が適したサーバもあり得る。

また NetNews の記事にはニュースグループという属性がある。ニュースグループは話題によって記事をふりわける単位である。必要なニュースグループを選べば、利用者の興味にあった記事だけを取得することができると考えられる。

NetNews では次々と記事が発生するため、不要な記事は削除することになる。必要な記事だけに絞って時間あたりの記事量を減らすことができれば、それだけ古い記事まで保存でき、ユーザにとっても利益となる。

サーバには記事のやりとりをする相手がひとつしかないものと、複数持つものがある。前者を末端のサーバと呼び、後者を幹線のサーバと呼ぶ。末端のサーバでは他の相手への配送がないので、単純に記事のアクセスパターンだけを考慮にいたれた配送ができる。

そこで本研究では末端のサーバをプル型で記事を取得するキャッシュシステムに置き換え

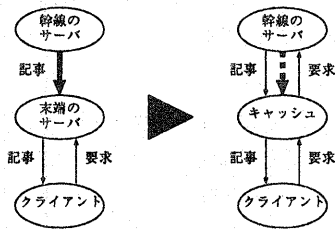


図 2: キャッシュのモデル

ることを提案し、それに適したキャッシュアルゴリズムを調査する。

3 NetNews のためのキャッシュシステム

本節では NetNews システムのためのキャッシュシステムについて述べる。はじめにモデルを示し、次にアルゴリズムについて説明する。

3.1 キャッシュモデル

図 2 の左のように、幹線のサーバと末端のサーバがあり、末端のサーバにクライアントが接続する状態を考える。幹線と末端のサーバ間は全コピーなので大量の記事が流れる。そこで図 2 の右のように、末端のサーバをキャッシュシステムに置き換える。

このキャッシュシステムはクライアントからの要求に対してサーバの役目をする。すなわち代理サーバとして働く。キャッシュに存在する(ヒットする)記事が要求された場合はその記事を返し、キャッシュにない記事はサーバから取得してクライアントに返す。全要求に対するヒットの割合をヒット率と呼ぶ。

またヒット率を向上させるために、右図に太い点線で示したように、一部の記事に対してサーバ間と同様のコピーを行うことを考える。

3.2 キャッシュアルゴリズム

キャッシュシステムを運用する場合には、さまざまなアルゴリズムを適用できる。まず大きくわけて次の二つがある。

- キャッシュ保存のアルゴリズム
- キャッシュ削除のアルゴリズム

ここではどのような基準で記事をキャッシュするのが有効であるかを調べるために、キャッシュ保存のアルゴリズムのみを考える。

以降の節では、キャッシュ保存のアルゴリズムを分類する選択項目について詳細を述べる。

3.2.1 記事の取得のタイミング

- プリフェッチ

要求の有無にかかわらず、サーバからキャッシュへ記事のコピーを行うことを、プリフェッチと呼ぶことにする。必要と予想される記事をあらかじめ配送することでヒット率の向上が期待できる。ただし不必要な記事を取得してディスク容量を無駄使いする欠点がある。これは先に述べたプッシュ型配送に対応する。

定期プリフェッチ、記事到着時プリフェッチ等が考えられる。

- オンデマンド

キャッシュシステムは代理サーバの役目を果しているため、キャッシュにない記事はサーバから取得して返す。経由するキャッシュシステムでこの記事をキャッシュすることで、同じ記事の要求に対してキャッシュした記事を返すことができる。必要最小限のディスク容量で済むが、記事に対して複数回の要求がないとヒット率があがらない。これは先に述べたプル型配送に対応する。

なおプリフェッチとオンデマンドは併用が可能である。

3.2.2 キャッシュする記事の選択方法

記事の属性によってキャッシュシステムの動作を変えることを考える。例えば、ニュースグループや記事の大きさを基準にしてキャッシュ保存する等が考えられる。

ある記事が読まれると、その記事に関連した記事が読まれる可能性が高いと判断し、まとめて記事を取得する方法も考えられる。このときニュースグループの全記事を移動したり、スレッド(参照記事のつながり)を構成する記事全部を移動することが考えられる。

3.3 関連研究

World Wide Webにおけるキャッシュの運用についてはさまざまな研究がなされている。Tatarinovらの研究では、Static cachingというキャッシュアルゴリズムの有利を示している[3]。Static cachingとは前日利用されたオブジェクトをあらかじめキャッシュしておくアルゴリズムである。本研究と同様に利用ログを用いたシミュレーションの研究である。この研究では削除のアルゴリズムについても調べられている。

NetNewsにキャッシュシステムを導入する提案はすでになされている[2]。しかしながら要求のあった記事をキャッシュするという単純な仕組みしか考えられていない。

4 実験

シミュレータを用いて、前述のアルゴリズムを比較する。シミュレータは各種アルゴリズムを採用できるようにしてあり、利用ログと記事流入ログを入力とし、キャッシュの使用量、ヒット率、ネットワーク使用量を出力する。実験では、奈良先端大のニュースサーバで1998年4月10日の一日で得られたログを入力に用いた。

本稿では以下のアルゴリズムを用いて実験を行った。なお全ての実験において記事は記事単位で移動する。

- キャッシュなし

この場合キャッシュシステムは代理サーバとして動作する。

- プリフェッチ
ここでは到着時プリフェッチだけを対象とし、「全記事選択」と「特定NG選択」について実験を行った。「全記事選択」とはニュースグループに関らず全ての記事をキャッシュにプリフェッチする方法で、「特定NG選択」とは特定のニュースグループに属する記事のみをプリフェッチする方法である。実験では前日に読まれたニュースグループを特定のニュースグループとした。
- オンデマンド
記事単位のオンデマンドキャッシュを行う。「全記事選択」ではサーバから取得した記事をニュースグループに関らずキャッシュに保存するが、「特定NG選択」では保存対象のニュースグループに属する記事のみをキャッシュする。プリフェッチと同様に前日読まれたニュースグループを保存対象とした。
- 併用
到着時プリフェッチとオンデマンドキャッシュを併用する。到着時プリフェッチを行っているので、「全記事選択」「特定NG選択」はプリフェッチの場合と全く同じになっている。そこで特定のニュースグループに属する記事のみをプリフェッチし、それ以外の記事はオンデマンドキャッシュする方法のみを調べた。

5 結果と評価

本稿で評価の対象とした事項を次に挙げる。

- ディスク使用量
ディスク使用量が少ないほど、古い記事まで保存できる。
- ヒット率
これは応答時間に関わる量であり、ヒット

表 1: シミュレーションの結果

98/4/10 1日分の結果		ディスク 使用量 (Mbytes)	ヒット 率	ネットワーク 使用量	
				S-Ca (Mbytes)	Ca-Cl (Mbytes)
キャッシュなし		0	0.0	99	99
プリ フェッ チ	全記事 選択	593	1.0	593	99
	特定NG 選択	200	0.93	204	99
オン デマ ンド	全記事 選択	66	0.49	66	99
	特定NG 選択	64	0.46	67	99
併用		203	0.96	203	99
理想的 プリフェッチ		66	1.0	66	99

率が高いほど応答時間は短くなる。従ってヒット率が高いほどユーザの利便性を増すと考えられる。特にサーバ・キャッシュ間の所要時間が大きいときに効果が期待できる。

● ネットワーク使用量

幹線のサーバとキャッシュの間の転送量を少なくすることにより、他のサービスにも帯域を利用できる。

結果を表 1 に示す。なおこの 1 日で流入した総記事量は 593Mbytes である。

5.1 ディスク使用量

まず全記事選択の場合をみる。必要なものだけを取得するオンデマンド (66Mbytes) は余分なものを含むプリフェッチ (593Mbytes) よりもディスクの使用量が少ない。プリフェッチの場合のディスク使用量はサーバに到着した記事の総量にあたり、オンデマンドのディスク使用量はそのうち要求された記事の量を表している。このことから記事の利用率が 10% 程度であることが再確認できる。

なおキャッシュしないときのディスク使用量は 0 である。

特定 NG 選択の効果を見ると、プリフェッチではディスク使用量が全記事選択の約 30% になっている。一方オンデマンドのみの場合はほとんど変化がなかった。

5.2 ヒット率

プリフェッチの場合、全ての記事がキャッシュに保存されているので、ヒット率は 1 である。オンデマンドの場合、約 0.5 になった。なおキャッシュしない時は 0 である。

ニュースグループを選択した場合、ニュースグループに関らずキャッシュする場合の 90% 以上を達成しており良い結果がでている。これはオンデマンドとプリフェッチのどちらについても言える。

5.3 ネットワーク使用量

キャッシュシステムの導入によりサーバ・キャッシュ間 (S-Ca)、キャッシュ・クライアント間 (Ca-Cl) の転送記事量がどのように影響をうけるかについて調べる。

キャッシュ・クライアント間の転送量は、アルゴリズムに関わらず同じ要求を処理しているので変わらない。一方、サーバ・キャッシュ間の転送量には大きな違いがみられる。

まずプリフェッチの場合は必要ない記事も転送するので転送量が多い (593Mbytes)。

キャッシュなしの場合はキャッシュ・クライアント間と同量が流れる (99Mbytes)。

またオンデマンドの場合は、キャッシュでヒットしなかった記事のみが流れるので、キャッシュ・クライアント間の転送量よりも少なくなる (約 70Mbytes)。全記事選択の場合の 66Mbytes という値は要求された記事の重複のない総量なので、これが最小限量である。

ニュースグループを選択すると、ディスク使用量と同様に、プリフェッチの場合は約 30% になり、オンデマンドの場合はほぼ不変である。ただしオンデマンドの場合は、キャッシュの対象としないニュースグループの記事が重複して転送されることがあるので、ニュースグループ

に問わずキャッシュする場合よりネットワーク使用量は増える。

5.4 アルゴリズムの評価

オンデマンドではディスク使用量を小さくできるがヒット率が小さくなる。プリフェッチでは逆にヒット率が大きくなるがディスク使用量が大きくなる。ユーザの利便性の指標であるヒット率を大きく保ったままディスク使用量を小さくするには、特定のニュースグループに属する記事を選んでプリフェッチするのが有効である。なぜならプリフェッチの全記事選択と特定NG選択を比較すると、ディスク容量が30%に減少しながらも、ヒット率は93%に達しているからである。

一方オンデマンドでは、属するニュースグループによってキャッシュする記事を選んでも効果は小さい。これは特定のニュースグループに利用が集中しているためと考えられる。

本研究の目的を理想的に達成するのは、要求される記事があらかじめ分かっているそれらが全てプリフェッチされている場合である。表1の理想的プリフェッチで各評価量を示す。ニュースグループを選択するプリフェッチでは理想的プリフェッチの3倍のディスク容量を必要とするが、ヒット率はあまり変わらない。

プリフェッチで保存の対象としなかったニュースグループをオンデマンドキャッシュする場合(併用)、プリフェッチの特定NG選択に比べて若干ヒット率が増大しており、理想的プリフェッチに少し近づいたアルゴリズムといえる。

以上のことから選択的コピーが有意義であることがわかる。また細かい選択制御によりさらに理想に近づける可能性があると考えられる。

6 おわりに

NetNewsの利用ログを用いてシミュレーションを行い、プリフェッチ、オンデマンド、記事の選択を組み合わせたアルゴリズムの比較を行った。キャッシュの使用量、ヒット率、ネットワー

ク使用量の評価から、記事の属するニュースグループによって選択的にプリフェッチすることにより、ユーザの利便性を損わずに資源を効率利用できることが分かった。

今回の結果は特定の一日についてであるから、普遍的な結果であるとは限らない。他の日についても調べる必要がある。また実験期間を長くした場合の調査が課題としてあげられる。アクセスパターンは奈良先端大のログから得られているので、局所的な性質が現れている可能性がある。今後は他のサーバのデータも考慮に入れる必要がある。なお削除のアルゴリズムも考慮に入れたより複雑な場合の実験も残っている。これらを調べてから、応答時間やサーバの負荷を考慮に入れて、総合的に評価する。

謝辞

IIJ 技術研究所の山本和彦氏には本研究に対して適切なアドバイスをいただきました。特に記して謝意を表します。

参考文献

1. 舟阪淳一、山本和彦、福田晃: 「ニュースグループごとに配送方式を自動変更するNetNewsシステムの評価」、電子情報通信学会技術研究報告 CPSY97-33、pp.49-54、1997
2. 知念賢一、山口英、山本平一: 「NetNewsにおける利用状況を反映する記事配送モデルの提案」、情報処理学会マルチメディア通信と分散処理研究会研究報告 No.71、pp.49-54、1995
3. I. Tatarinov, A. Rousskov and V. Soloviev: "Static Caching in Web Servers", Technical Report, North Dakota State University, NDSU-CSOR-TR-97-04, 1997