システムソフトウェアと
オペレーティング・システム 81－22
（１９９８．５．７）

NetNewsにおける必要な記事の自動選択法

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のトラフィックは日々増加しており、1日20 Gbytesを越える日も少なくない。その結果、記事の保持期間の短縮、回線帯域の浪費、ディスクアクセスの大量発生といった問題が起きている。本研究ではこれらの問題を解決するために、末端のサーバをキャッシュシステムに置き換え、アクセスされる可能性の高い記事を予めキャッシュに取り込み、それ以外の記事は要求に応じて取得する方法を提案し、評価する。提案手法はニュースグループ単位で予めキャッシュに取り込むことを考え、どのニュースグループを取り込むかを過去のアクセス状況を用いて決定する。本稿では、アクセス状況を参考にする期間の長さ、利用率の閾値をさまざまに変えてニュースグループ選択の効果に関する実験を行ったので、その結果を報告する。

Automatic Selection of Required Articles for NetNews

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

Abstract

The traffic of NetNews is increasing day by day, and often exceeds 20Gbytes per day. This causes the following problems. Time span of articles which can be stored shrinks, network band is wasted, and disk accesses increases heavily. In this paper, in order to solve above-mentioned problems, a method that an end server is replaced with a cache system, and articles which may be requested are fetched into cache in advance, and other articles are fetched on demand, is proposed and evaluated. Taking account of past access pattern, the method decides which newsgroups are to be fetched in advance. Moreover, applying various span for referring access pattern and threshold of access frequency, experiment for investigating effectiveness of selection is carried out and its results are shown.

1　背景

インターネット上のトラフィックが年々増加している。インターネット上ではさまざまなアプリケーションで利用されるデータが流れているが、ここではNetNewsのトラフィックについて注目する。NetNewsの記事はサーバ間でやりとりされており、最近では1日20Gbytes、50

万通を越える日も少なくない[1]。このように従来のシステムが想定していなかった大量の記事が流通することにより、以下のような問題が発生する。

1. 記事の保持期間の短縮
2. 回線帯域の浪費
3. ディスクの読み書きの大量発生

第1の問題点は記事の保持期間の短縮である。例えば一日 20 Gbytes のトラフィックの場合、20 Gbytes のディスクを用意しても一日分の記事しか保存できない。保持期間が短いと、ユーザが話の流れをさかのぼって読みたいときに古い記事が少なく不便である。また利用頻度の少ないユーザは多数の記事を読み損ねることになる。一定のディスク容量しか利用できない場合、トラフィックが多いほど深刻な問題になる。

第2の問題点は回線帯域を大量に消費することである。一日 20 Gbytes というのは平均して約 2 Mbps に相当しており、T1 とよばれる回線 (1.5 Mbps) でも帯域が足りないほどの流量である。この回線帯域の消費は他のサービスに遅延を引き起こす可能性がある。

第3の問題点はディスクの読み書きの大量発生である。流れる記事数は 1 日 50 万通を越える日も多く、記事のディスクへの保存およびディスクからの読み出しの回数もそれだけ多くなる。これはディスク装置を酷使することとなり、トラブルの発生を誘発すると考えられる。

すべての記事を必要とするサーバは以上のような問題を避けられない。しかしながらすべてのサーバがすべての記事を必要とするわけではない。世界中のサーバ間で記事をやりとりするための中継を主な目的としたサーバ (以下、基幹サーバと呼ぶ) は、すべての記事を保存し転送しなければならない。しかしながら、ユーザに閲覧、投稿させることを主な目的とするサーバのうち、配送相手を一つしか持たないサーバ (以下、末端サーバと呼ぶ) はすべての記事を保持する必要はない。例えば、あるサーバではユーザから要求された記事は、全流入記事の約 10%しかないことがわかっている [1]。つまりこの 10%の記事のみを保持できれば、上記の 3 つの問題に対処できる。

そのため本研究ではこれらの末端サーバをキャッシュシステムに置き換え、選択的に記事を保存することを考えている。この際、予め取得されていない記事は基幹サーバまで取りにいかなければならないため、要求される記事の大半を予め取得しておかないと応答性が悪化する。我々は既にこの条件において、必要と予想されるニュースグループの記事を選択的に予めキャッシュしておき、残りの記事を要求に応じて基幹サーバから取得するという方式の有効性を確認している [2]。そこでは選択するかしないかを問題としてきたため、選択方法を単純にしていた。このため、選択方法をより精緻にすることにより、資源をより有効に活用できるキャッシュシステムが実現できると予想される。

本稿では、限られた資源でできる限り要求される記事だけを取得するキャッシュシステムの構築を目標とし、そのための有効な記事選択方法を検討する。ニュースグループを選択することの有効性はわかっているので、本稿ではその選択基準をより詳細に調べる。具体的には、時々刻々変わると考えられる読者の記事選択状況に対応できることと、管理者の管理コストの軽減を考え、選択を自動的に適切に行うことができるアルゴリズムを検討する。

NetNews の記事選択についての関連研究として、見込み計算を用いて必要と予想される記事を予め取得して、ニュースリーダの応答性を向上する試みがある [3]。この研究では、一週間ごとの購読率の高い読者が多く存在するニュースグループの記事を予め取得する手法 (アクティブキャッシュ) を採用している。応答性を主に評価の対象としているものの、ニュースグループ単位のキャッシュの有効性も示している。

キャッシュの有効利用についての研究は World Wide Web について盛んに行われている ([4][5][6][7])。World Wide Web ではクライアントの

[1] すべてのニュースグループの記事を配送している Tokyo-Net における全流入記事の統計 (news:tnn.netnews.stats)

要求があってはじめて配送されるプル型配送を採用しているため、サーバ主導で配送されるプッシュ型配送も用いるNetNewsにはWorld Wide Webにおける研究成果をそのまま適用することはできない。しかしながらクライアントから要求のあったオブジェクトだけを保存する受動的キャッシュだけでなく、あらかじめ必要と予想されるオブジェクトを保存する能動的キャッシュもあわせて利用していく必要性が叫ばれて来ており[8]、本研究も着想において参考としている。

2 利用傾向

適切な選択アルゴリズムを考えるためには、読者の利用傾向を調べる必要がある。ここでは次の項目を調査した。

1. 利用傾向の周期性
   職場や研究室でNetNewsを利用する場合、生活における一週間の周期が利用傾向に反映されると考えられる。適切な選択アルゴリズムを決めるためには、この周期を考慮に入れる必要がある。

2. 要求されるニュースグループのばらつき
   常時アクセスされるニュースグループが決まっていれば、そのニュースグループの記事だけをキャッシュしておけばよい。ときどきアクセスされるニュースグループがどれだけあるかを把握して、選択基準の参考とする。

上記項目を、奈良先端科学技術大学院大学(以下NAIST)と九州大学情報処理教育センター(以下ec.kyushu-u)の各ニュースサーバについて調査した。調査にはサーバへの流入記事のログとクライアントからの要求記事のログを用いた。各ログの特徴を表1に示す。all req.は全要求数および記事量を示し、uniq. req.は要求された記事数および記事量を示す。例えば、同じ記事が複数回要求された場合、uniq. req.としては1記事だがall req.には要求された回数分計上される。またall incomeは全流入記事数および記事量のことである。

表1: ニュースサーバの記事処理量

	NAIST		kyushu-u
	4/10 - 5/20 (41日)	8/13 - 9/22 (41日)	6/27 - 9/30 (96日)
all req.	379,447	268,895	99,741
(bytes)	10,701M	4,027M	237M
uniq. req.	139,543	104,631	57,465
(bytes)	4,609M	1,865M	142M
all income	6,693,879	7,134,151	1,266,474
(bytes)	27,633M	23,945M	5,765M

表1をみると明らかなように、NAISTの流入記事数は冒頭で述べた1日20 Gbytes、50万記事という流量に比べると小さく、ec.kyushu-uの流入記事数はNAISTの1/10程度である。これはNAISTに流入する記事の多くは研究機関を経由してきたものであり、研究に不必要な記事は途中で中継されないためであると考えられる。またec.kyushu-uにおいてさらに流入記事数が少ないのは、取得するニュースグループを少なくしているためである。このように限られたニュースグループしか購読しない運営方針は、基幹サーバとしては不適当であるが、末端サーバとしては十分考えうるので、今回の研究対象とした。

以下では、主にec.kyushu-uにおける利用傾向と実験結果について詳細を述べる。

2.1 利用傾向の周期性

一日あたり利用されるニュースグループ数を示したのが図1である。図の□印が一日あたりの利用ニュースグループ数である。これをみると、縦線を入れた土曜、日曜に利用ニュースグループ数が大きく減ることがわかる。そこで一週間の移動平均をとって図中に＋印で示した。

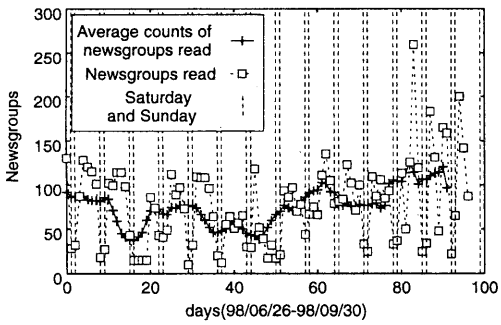

図 1: 利用ニュースグループ数

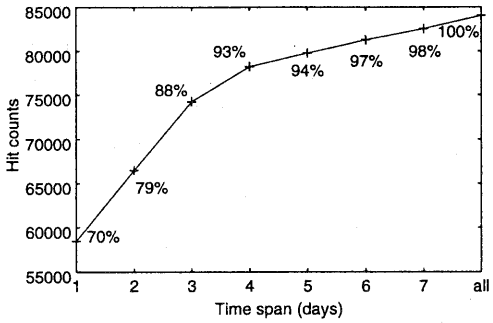

図 2: アクセス頻度で分類したニュースグループとそれらで実現できるヒット数

これをみるとかなり変動が少なくなっており、一週間でアクセスを平均化することにより曜日によるヒット率の低下を減らすことができると予想される。

なお NAIST でも同様に一週間の周期が観測でき、移動平均によって曜日による利用ニュースグループの低下がおさまっている。

2.2 要求されるニュースグループのばらつき

ある一日の全アクセスのうち、過去何日かでアクセスされたニュースグループに所属する記事へのアクセスがどれだけあるかを調査した。具体的にはニュースグループを最後にアクセスされた日によって分類し、各ニュースグループ群がどれだけのヒット数を得られたかを計算した。図 2は期間中のヒット数を総計したものである。

図の横軸は最後にアクセスされた日と計測した日の相対を表し、縦軸はその日以降に分類されるニュースグループで計上したヒット数を示す。all は全ヒット数である。図中の数字は、全ヒット数に対する割合 (ヒット率) を百分率で示している。この図をみると、7 割のヒットが前日にアクセスされたニュースグループの記事に対するものであり、7 日前までを考慮に入れれば 98%までヒット率を向上できていることがわかる。

3 実験

実験は、NAIST と ec.kyushu-u の流入記事のログ、要求記事のログを用い、キャッシュに保持される記事のシミュレーションを行った。ニュースグループを選択するアルゴリズムをかえた場合のヒット率とディスク使用量について以下で説明する。

3.1 期間の長さ

文献 [2] において、選択対象は前日 1 日にアクセスされたニュースグループだけであったが、この期間を広げることから考える。直前の 2 日、3 日、…、10 日間にアクセスされたニュースグループを選択の対象とする。このときのヒット率とディスク使用量の関係を図 3に示す。図 3 の内側のグラフは、前日 1 日から 10 日間の部分を拡大したものである。

図 3をみると、期間の長い方がヒット率が高く、7 日間をこえるとヒット率の増分に対するディスク使用量の増分が大きくなることがわかる。これは、過去 7 日間のアクセス状況のみ考慮すれば良いことを示している。7 日間にアクセスされたニュースグループを選択の対象とすると、ディスク使用量は全記事を予め取得する方法 (All prefetch) の 1/5 以下に抑えられ、ヒット率は要求のあった記事のみを取得する手法 (All

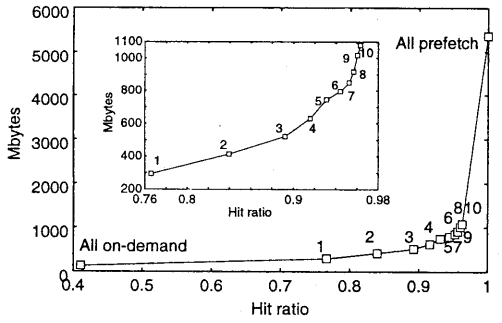

図 3: 期間をかえたときのヒット率とディスク使用量の関係

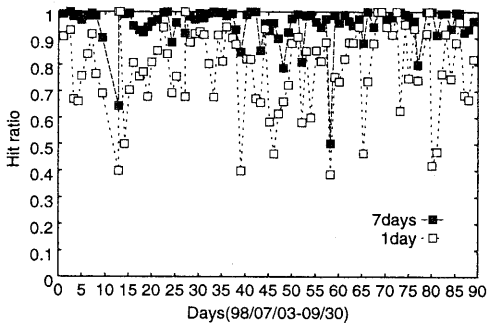

図 4: 1 日ごとのヒット率

on-demand) から 0.5 以上向上している。

さらに、期間を 1 日間、7 日間とした場合の 1 日ごとのヒット率を図 4に示す。利用傾向から予想した通り、ヒット率の周期的な低下がかなりおさまっていることがわかる。

3.2 利用率

より細かく購読状況を選択に反映させるため、利用率という値を導入する。あるニュースグループ n の利用率 $U(n)$ を、

$$U(n) = a(n)/i(n)$$

と定義する。ここで $a(n)$ はアクセス数、$i(n)$ は到着記事数である。ただし、$i(n) = 0$ のときは

$$U(n) = \infty \qquad (\text{for } a(n) > 0)$$

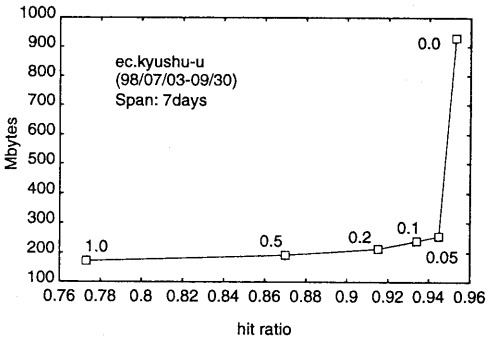

図 5: 閾値をかえたときのヒット率とディスク使用量の関係

$$U(n) = 0 \qquad (\text{for } a(n) = 0)$$

とする。

ここで閾値 U_T を決め、$U(n) > U_T$ を満すニュースグループを選択する。これまでの「前日に 1 回以上アクセスされたニュースグループ」は、期間を前日の一日として、$U_T = 0$ の閾値に相当する。

U_T を 0 から 1 まで変化させて、ヒット率とディスク使用量の関係をグラフにしたものが図 5である。

この図は ec.kyushu-u のログを用いて、期間を 7 日とし各 U_T に対する実験結果を示す。今回の実験では、0.05 の閾値を用いることにより、ヒット率を約 0.95 に高く維持したまま、ディスク使用量を 300 Mbytes 程度 ($U_T = 0$ のときの約 1/3) に抑えることができている。

4 議論

期間の長さによるヒット率とディスク使用量の変化をみると、ヒット率をあげるためには期間が長い方が有利である。しかしながら、1 日と 7 日の結果を比べると、ヒット率は 0.16 しか向上していないのに対し、ディスク使用量は 700 Mbytes 増えている。これは 1 日分の記事 300 Mbytes で 0.76 というヒット率に比べて効率が悪い。ニュースサーバの持つ資源は各サイ

トによってさまざまであり、ここであげた結果はその環境によって評価する必要がある。

そこで利用率をニュースグループ選択の基準に用いることにより、この 700 Mbytes の増加を押さえることを試みた。7 日の期間を固定したまま利用率を基準としてニュースグループを選択すると、図 5 のような結果が得られた。0.05 の閾値を用いれば、7 日分の結果 (0 の閾値) とほぼ同じ 0.95 という高い閾値を保ったまま、ディスク消費を 1 日分の結果である約 300 Mbytes に抑えることができた。

5 今後の課題

利用率を導入することでディスクの使用量を抑えつつ高いヒット率を得ることができたが、より複雑な処理を行っているため作業量は増えている。この処理量を従来の単純な方法と比較して、評価する必要がある。

また、記事を読む場合は到着順や参照する記事のつながり (スレッド) の順で読むことが多い。これらの情報を用いて記事を選択する方式も検討する必要がある。しかしこの方法は、ニュースグループよりも細かい単位で処理するため、ヒット率をより向上させる可能性はあるものの、処理が繁雑になることもあり効果が得られるとは限らない。

ニュースサーバを持つ環境は組織により、利用できる回線帯域もディスク容量も異なり、ユーザの利用傾向も異なる可能性がある。また基幹サーバとキャッシュシステムの間、キャッシュシステムとクライアントの間のネットワーク所要時間は、ユーザが許容できる応答時間と密接に関係する。今後はこれらを定量的に評価し、それぞれに合った記事選択方式を提案していく予定である。

謝辞

九州大学情報処理教育センターには流入記事のログと利用記事のログを提供していただきました。ここに記して感謝の意を表します。

参考文献

1. 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.

2. 舟阪淳一, 最所圭三, 福田 晃: NetNews のためのキャッシングアルゴリズム, 電子情報通信学会論文誌 (掲載予定).

3. 池口祐子, 村山和宏, 上田和紀: 見込み計算を用いたニュースリーダの応答性改善法, 情報処理学会論文誌, Vol.38-6, pp.1235-1244, 1997.

4. I. Tatarinov, A. Rousskov, and V. Soloviev: Static Caching in Web Servers, Proc. 6th Int'l Conf. on Computer Communications and Networks, 410-17, 1997.

5. J. Pitkow and M. Recker: A Simple Yet Robust Caching Algorithm Based on Dynamic Access Patterns, Proc. the Second International WWW Conference, 1994.

6. B. M. Duska, D. Marwood, and M. J. Feeley: The Measured Access Characteristics of World-Wide-Web Client Proxy Caches, Proc. the USENIX Symposium on Internet Technologies and Systems, pp.23-35, 1997.

7. S. Williams, M. Abrams, C. Stanbridge, G. Abdulla, and E. Fox: Removal Policies in Network Caches for World-Wide Web Documents, Proc. ACM Sigcomm '96, pp.293-305, 1996.

8. Z. Wang, J. Crowcroft: Prefetching in World Wide Web , Proc. IEEE Globecom'96, 1996.