マルチメディア通信と分散処理 78－6
（1996．9．27）

NetNewsにおけるスレッド単位の情報フィルタリングシステムの構築

出雲 正尚, 知念 賢一, 山口 英
奈良先端科学技術大学院大学 情報科学研究科

概要

現在 NetNews で交換される情報は膨大であり，一人のユーザが全ての記事を閲覧することは不可能になっている．この問題に対する解決方法として情報フィルタリングが研究されている．しかしながら，これまでの研究では，個々の記事に対するフィルタリングについての研究が中心であり，関連した他の記事の情報を得ることができない点が解決されていなかった．そこで本研究では，NetNews で交換される記事群での構造としてスレッドに注目し，スレッド単位のフィルタリングシステムの設計を行ない，その実装について述べる．

An Implementation of Thread-based Information Filtering System for NetNews

Masanao Izumo, Ken-ichi Chinen, Suguru Yamaguchi
Graduate School of Information Science,
Nara Institute of Science and Technology

Abstract

In the NetNews system on the Internet, the huge numbers of articles are posted everyday so that it is impossible for a single user to retreive all articles on it. In order to decrease the number of accesses for find out articles a user wants, several systems with the "information filtering" technique have been proposed and developed. However, these systems are too simple because their filtering mechanisms evaluate each articles separately. Articles on the NetNews has its own logical strucutre called "thread." It is obious that what users want after filtering is not a single artcle but a thread of the article. We've devleoped yet another information filtering system for NetNews which employs the concept of "thread filtering." Futheremore, we confirmed that this system improved user environment of the NetNews system through its evaluation process. In this paper, we report design, implementation, and evaluation of the thread filtering system.

1 はじめに

インターネット上のニュースシステム (以下 NetNews) は，USENET 環境で開発され，インターネット上の情報共有サービスとして広く普及している．ユーザは記事 (article) を投稿したり，その記事に対する意見や反論の記事を投稿できる．さらに，ユーザが目的の記事を探し出せるように記事はニュースグループと呼ばれるグループに分かれている．

しかし，現在ではニュースグループの数，そして記事の数や量は膨大になり，一人のユーザが全ての記事を参照することは不可能となっている．文献 [1] によれば，NetNews によって新たに提供される記事は，一日平均 74,000 通，約 200 MB もの情報が交換されている．このため，ユーザは自分が求める記事を探し出すのに，多くの時間をさかなければならなくなっている．

このようなことから，NetNews に投稿された記事から，効率よく有用な記事を探し出す手法がこれまで数多く提案されてきた．その一つの手法として，記事の内容によってフィルタリングを行ない，ユーザが参照する記事数を絞り込む手法が研究されている．例えば文献 [2] では，ユーザが指定した単語を含む記事だけを抜き出すようなシステムについて報告されている．

ところが，NetNews にはスレッド (thread) と呼ばれる記事間の関係構造がある．NetNews においては，ある記事に対して，別のユーザが意見や反論を記事として投稿することが多い．元になる記事をイニシャル記事，それに続く意見や反論をフォロー記事と呼ぶ．さらに，それらをまとめてスレッドと呼ぶ．文献 [3] によれば，イニシャル記事が有用であれば，その後に続くフォロー記事もまた高い確率で有用であることが示されている．この事実は，NetNews を対象とするフィルタリング機構において，スレッド単位でフィルタリングを行なうべきであることを示唆している．

そこで，本研究では，NetNews でのスレッドを膨大な記事群から効率よく抽出する方法と，スレッドに着目したフィルタリング機構を提案し，その設計と開発を行なった．また，このシステムをユーザが使用した場合に，良好な利用者環境を与えることを確認した．

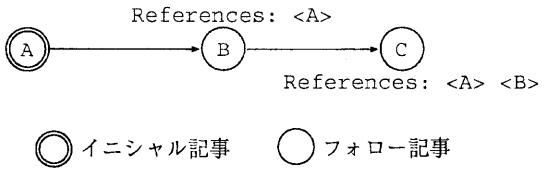

図 1: イニシャル記事とフォロー記事

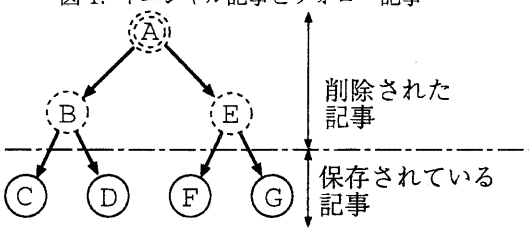

図 2: 削除された記事を含むスレッド

2 NetNews でのスレッドの抽出

NetNews の記事はヘッダと本文から構成される．フォロー記事のヘッダの References 行には，イニシャル記事の ID (Message-ID) が記載される．イニシャル記事 A に対するフォロー記事 B があった場合，B に対するフォロー記事 C の References 行には，A, B の順序で Message-ID が記載される．さらに C に対するフォロー記事では A, B, C の順番になる (図 1)．しかしながら，スレッドに含まれる記事が増えると References 行の記載も長くならざるをえない．そのため，NetNews では，References 行の縮退を行なうことを許している．この場合，直前のフォロー記事の ID は必ず残すことが規定されている．このことから，記事の References 行を解析することによってスレッドを抽出できることが分かる．

ところが，NetNews システムが使用するディスク領域は容量に制限があるため，記事をシステム内に保存する期間や総量に限界がある．したがって，単にサーバに蓄積されている記事だけを使ってスレッドを抽出したとしても，完全なスレッドの再構成は不可能である (図 2)．つまり，スレッドを完全に抽出するためには，過去におけるスレッドに関する情報をすべて保持することが必要となる．

スレッドに注目したニュースリーダは，GNUS を代表として，これまでいくつかのシステムが開

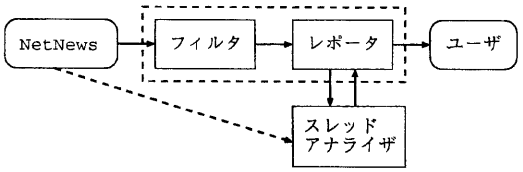

図 3: システム構成

発されてきた．しかしながら既存のシステムでは，スレッドの抽出をニュースグループ内に限定していたり，あるいは，記事のサブジェクトからスレッドを構成していた．本システムでは，より正確なスレッドの抽出と，複数のニュースグループにまたがったスレッドの抽出を行なうために References 行のみを参照してスレッドを構築している．

3 設計と実装

本章では，NetNews におけるスレッド単位の情報フィルタリングを実現するための設計と実装について述べる．

3.1 システム構成

本システムが実現すべき機能としては，

1. 有用な記事を取り出す．
2. 有用な記事を含むスレッドを取り出す．
3. 得られた記事をユーザに提供する．

がある．そこで本システムでは，機能毎にフィルタ，スレッドアナライザ，レポータの 3 つのモジュールに分割し，設計を行なった．図 3 にシステム全体の構成を示す．

3.2 フィルタ

フィルタは有用な記事を取り出すモジュールである．新しい情報を随時取り寄せ，ユーザの求める (有用な情報) 条件で記事を検査して，適合した記事を取り出す．前述のように，フィルタは，1 日 約 200 MB という膨大な量をフィルタリングしなければならない．投稿された記事全てを各々のユーザへ配送し，フィルタリングを行なう場合，ネットワークの負荷が高くなってしまう．そこで，複数のユーザのフィルタリングを一括して処理することにする．具体的には，フィルタは NetNews サーバ毎に起動されるデーモンとして構成する．

```
(a)  condition                { action }
(b)  JWordMatch("京都")      { MessageID }
```

図 4: フィルタの文法記述

3.2.1 フィルタの動作記述

ユーザの求める記事の条件に適合した記事をどう処理するかをフィルタに伝える必要がある．この条件と処理方法を動作記述と呼ぶ．

フィルタはニュースサーバから新着記事を取り寄せ，与えられた動作記述にしたがってフィルタリングを行う．動作記述はパターンとアクションに分かれる．フィルタは，記述されたパターンに記事が適合した際に，アクションを実行する．本システムでは図 4(a) に示した文法にしたがって記述する．パターンは文字列の比較，単語の比較，それらの比較結果を AND, OR, NOT の論理演算で記述する．アクションは適合した旨を通知するために用いられ，アクションの実行結果はフィルタに通知される．

例えば，4(b) の動作記述をフィルタに与えると，単語「京都」と記事の比較を行い，適合した場合にその Message-ID を返す．ここで JWordMatch は，指定された単語「京都」が記事の中に含まれていれば真となる．本フィルタは，形態素解析プログラム JUMAN[4] を用いて単語単位の比較を実現しているので，記事の中に「東京都」という部分文字列が含まれていても「京都」には適合しない記述が指定できる．

3.3 スレッドアナライザ

スレッド単位でフィルタリングするために，スレッドアナライザは与えられた記事からその記事が含まれるスレッドを取り出す．新しい情報を随時取り出して，スレッドの構造を記録する．これによってサーバ側で記事が削除されたとしても，スレッドアナライザでは正確にスレッドを抽出できる．

本システムのスレッドアナライザは，References のみからスレッドを作成しているので，複数のスレッドにまたがったスレッドを作成できる．

なお，スレッドアナライザは，一定期間フォローのなかったスレッドをスレッドごと削除する．

3.4 レポータ

レポータは，フィルタからの結果をユーザに提供するための処理を行なう．レポータは，フィルタリングの条件をユーザから受けとり，フィルタの動作記述を生成してフィルタを駆動する．次にフィルタ結果を基に，スレッドアナライザからフィルタされた記事を含むスレッドを取り出し，得られたスレッドを整形してユーザに提供する．

現在のレポータの実装は，記事を HTML に変換し，ユーザの指定した場所に保存するようになっている．記事の中に URL が含まれていれば，HTML のリンクを自動的に作成する．ユーザは Netscape などの WWW ブラウザを通して記事を読むことができる．

3.5 実装

フィルタやスレッドアナライザは NNTP[5] を用いて，ニュースサーバと通信している．そのため，ニュースサーバと通信可能な環境下であれば，本システムを利用することが可能である．

フィルタとスレッドアナライザはデーモンとして動いており，TCP/IP により，通信可能である．

スレッドデータベースには，gdbm を用いて実現されている．

4 評価

4.1 性能評価

この節では，スレッドアナライザ，およびフィルタの処理能力を調べるために，性能を評価する．なお，測定は以下の環境で行なった．

ハードウェア: `Pentium/166, Memory/32MB`
`Hard Disk 4G,`
ソフトウェア: `OS BSD/OS 2.1`
`DBMS gdbm, Compiler gcc`

実測は 1996 年 8 月 27 日，奈良先端大のニュースのスプールにあった 50 万記事を用いた．

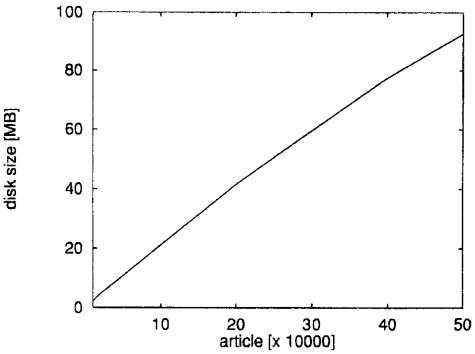

図 5: 登録記事数と必要なディスク容量

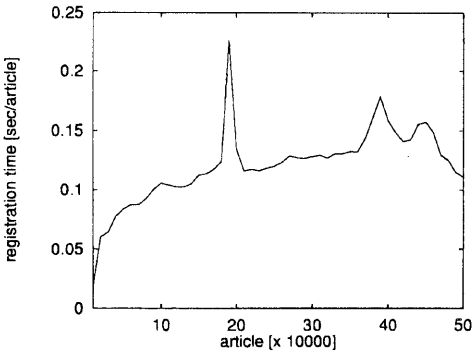

図 6: 現在の登録記事数に対する，1 記事当たりの登録時間

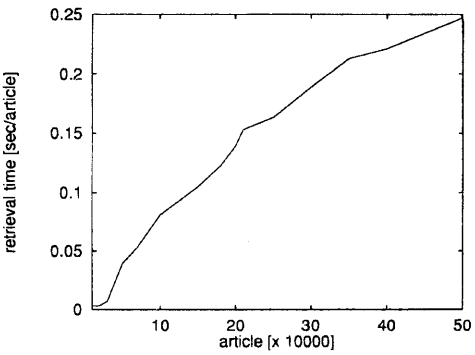

図 7: 現在の登録記事数に対する，1 記事当たりの検索時間

4.1.1 スレッドアナライザ

スレッドアナライザは，データベースを用いて大量のデータを処理している．そのため，処理速度とファイル容量が問題となる．スレッド登録数が増加した場合に，データベースが使用できるファイル容量が十分かどうか，あるいはデータベースに対する登録や検索速度が実用に耐える速度で動作するかどうかを検討する必要がある．そこで，登録記事数に対する必要なファイル容量，および登録・検索速度を測定した．

図 5 にスレッドアナライザが使用しているスレッドデータベースのファイル容量と登録記事数の測定結果を示す．

図 5 から必要なファイル容量はスレッドアナライザに登録されている記事数に比例することが分かる．現状の NetNews では 1 日平均約 74,000 記事が投稿されている．したがって，1 ヶ月分のスレッドデータを保持するのには約 400 MB のディスク容量が必要となる．

図 5 には，データベースに登録されている記事数と，新たな記事を登録するための時間についての相関を示している．現在の流速は，約 1 ～ 2 記事/秒 であるので，登録に必要とされる時間は必ず 0.5 秒以下でなければならない．

よって，図 6 から，記事の登録は十分間に合う速さで行なわれる．なお，登録記事数 20 万 近傍でピークが見られるが，実験中に他のプロセスが動いて，一時的に性能が低下したと考えられる．

検索速度については，図 7 のようになる．データベースに 1 週間分 (約 50 万記事) の記事が登録されているとき，1 つのスレッドを得るために約 0.25 秒を要している．

4.1.2 フィルタ

NetNews の記事は，平均して 1 秒間に 約 1 記事，多い時には 2 記事到着する．よって，フィルタは 1 秒間に 2 記事以上を処理できなくてはならない．本システムが要件を満足しているかを確認するために，フィルタの処理時間を測定した．

「インターネット」と「スレッド」という単語を条件に与えて，それぞれ，処理時間を測定した．対象とした記事は，fj ニュースグループの 1000 記事，約 1.3 MB である．1000 記事中,「インターネット」は 29 記事,「スレッド」は 9 記事に含まれている．フィルタリングの条件を同じにした複数のユーザで測定した場合に，単語「インターネット」,「スレッド」についての検索速度のグラフが図 7 である．この図から，ユーザ数に比例してフィルタリングの時間がかかることが分かる．

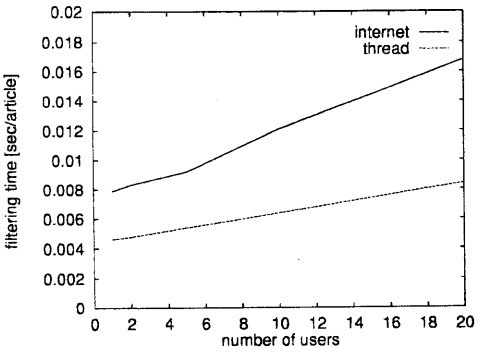

図 8: フィルタの性能

図 7 から，20 人のユーザが「インターネット」という単語を指定した場合，フィルタリング処理をするのにかかる時間は 1 記事当たり 0.017 秒であり，フィルタとして十分な処理速度であることが分かった．

なお,「インターネット」と「スレッド」の検索速度やグラフの傾きの差は単語の適合する回数が異なることに起因するものである．

4.2 ユーザによる評価

スレッド単位のフィルタリングの有効性を示すためにアンケートに調査を行なった．アンケートでは，実際に本システムを利用しているユーザに対し，以下の調査を実施した．

- 得られたスレッド数
- 各々のスレッドについて，有用だと思った記事数．

被験者は，奈良先端大に所属している 5 名である．

図 9 は，スレッド内の記事「全て有用」,「一部有用」,「全て不用」の割合を表した図である．この図からフィルタリングした結果得られたスレッドの約半数が有用な記事を含むことが分かる．

また，ある実験者について，指定した単語に適合しない有用記事数を調査した結果が (表 10) で

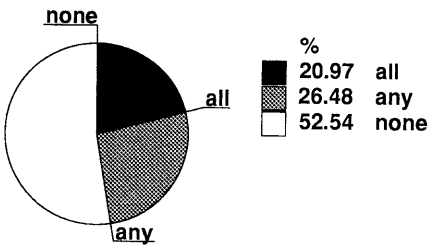

図 9: スレッド内の有用記事数

	有用	不用	計
キーワードを含む記事数	26	132	158
キーワードを含まない記事数	35	194	229
計	61	326	387

図 10: スレッド内の記事の内分け

ある．この表によると，有用記事中，57% は，指定した単語が含まれていない有用記事であることが分かった．つまり，記事単体のフィルタリングでは得られなかった有用な記事が，スレッド単位でのフィルタリングにより得られるることが分かった．

4.3 考察

本システムでは，スレッド単位でフィルタリングすることにより，記事毎のフィルタリング以上に有用な情報を得ることができた．また，日本語形態素解析による日本語の単語単位の比較処理も十分処理が追い付くことも分かった．

ところが，フィルタリング結果にはユーザの意図しない記事も多く含まれた．シグネチャなど，ユーザの意図しない場所に指定した単語が含まれていたことが原因であると考えられる．そのため，フィルタリングの前処理として，必要ないシグネチャを切り出すなどの処理が必要である．また，単語だけではユーザが望んでいる記事を表現することはできないので，単語によるフィルタリング以外の手法も考えなければならない．例えば連想キーワードベクトルの利用 [6]，曖昧な検索など，より高度な手法を用いたフィルタリングが必要である．

本システムのスレッドは，スレッドアナライザのスレッド管理により，サーバ側の記事の削除とは関係なくスレッド情報を保存できる．図 5 から，1G のディスクで数ヶ月分のスレッド情報を保存できることが分かった．また，登録数やレポータのアクセスが増加すると，スレッドアナライザの応答が劣化する可能性があることも分かった．

5 おわりに

NetNews の記事は膨大で，一人のユーザが全ての記事を参照することは不可能となっている．本研究では，その解決方法として情報フィルタリングを用いた NetNews フィルタリングシステムを提案した．また，NetNews は単一の記事だけでなく，スレッド単位で利用されていることに着目し，スレッド単位のフィルタリングを実現した．フィルタリングした結果をスレッド単位でまとめることにより，多くの有用な記事を得ることができた．

参考文献

[1] <URL:news:tnn.netnews.stats>.

[2] Hector Garcia-Molina Tak W. Yan. Sift - a tool for wide-area information dissemination. *In Proceedings of the 1995 USENIX Technical Conferences, pages 177-86*, 1995.

[3] 森田昌宏 篠田陽一. 情報洪水の緩和のための情報フィルタリングの実現. *JAIN Symposium*, 1994.

[4] 松本裕治 黒橋禎夫 宇津呂武仁 妙木裕 長尾真. 日本語形態素解析システム juman 使用説明書 2.0. 奈良先端大技術報告書 *NAIST-IS-TR94025*, 1994.

[5] P. Lapsley B. Kantor. Network news transfer protocol: A proposed standard for the stream-based transmission of news. *RFC 977*, 1986.

[6] 森康真 國藤進. 情報フィルタリング機能をもつ発散的思考支援環境の試作. 情報処理学会研究会，マルチメディア通信と分散処理 *63-18* グループウェア *5-18, P.133-140*, 1994.