数学的性質を利用した処理方法最適化機構をもつ情報フィルタリングシステム

An Information Filtering System with an Optimization Mechanism of the Processing Method Based on Mathematical Properties

小寺 拓也♦ 澤井 里枝♦
寺田 努♠ 塚本 昌彦♠
西尾 章治郎♠

Takuya KODERA Rie SAWAI
Tsutomu TERADA Masahiko TSUKAMOTO
Shojiro NISHIO

近年，必要なデータのみを自動的に抽出する情報フィルタリング技術に対する注目が高まっている．フィルタリングの最適な処理方法は受信機の数や空きリソース，ネットワークの負荷などの環境により変動するため，状況に応じて処理方法を変更する必要がある．しかし，そのような処理方法の変換を実現するためには，システム稼動中に処理方法を変更しても一貫したフィルタリング結果が得られることを保証する必要がある．そこで本稿では，数学的性質を利用した処理方法最適化機構をもつ情報フィルタリングシステムの構築を目的とする．本システムは，状況に応じてより効率的な処理方法を自動的に選択する機構をもつため，さまざまな環境において常に効率的なフィルタリング処理が行える．

In recent years, there is a strong demand for filtering techniques that automatically extract only necessary data. The optimum processing method of filtering changes according to several environmental factors such as computational capability, the number of receivers, and network load. However, to change the processing method according to the environment, it should be assured that the filtering results are consistent among multiple processing methods. In this paper, we show the implementation of an information filtering system that optimizes the processing method based on mathematical properties. This system can automatically change the processing method to the optimum method according to the environment.

1. はじめに

近年，新たな衛星の打ち上げや放送のデジタル化によって，従来のただ見るだけの放送に加えて，番組表やHTMLファイルなど多種多様なデジタルコンテンツを放送するデータ放送サービスが一般的になると予想される．しかし，大量の受信データから必要な情報を探し出すことはユーザにとって非常にコストの高い作業であるため，ユーザが必要とするデータを自動的に選択する情報フィルタリングシステムが求められている[2]. これまでに提案されたフィルタリングシステムでは，データを受信する度に取捨選択する逐次処理を行うのが一般的であるが，逐次処理では継続的に受信する大量のデータを逐一処理しなければならないため，受信機の処理コストが高くなる．このようなフィルタリングの処理コストを軽減するためには，複数の受信機で処理を分散する並列処理や，ある程度データをためておき一度に処理する一括処理が有効である．最適な処理方法は受信機の数や空きリソース，ネットワーク負荷など環境により変動するため，状況に応じて処理方法を変換する必要があるが，処理方法の変換を実現するには，システム稼動中に処理方法を変更しても一貫したフィルタリング結果が得られることを保証する必要がある．

そこで本研究では，数学的性質を利用した処理方法最適化機構をもつ情報フィルタリングシステムの構築を目的とする．本システムは，逐次処理，一括処理などフィルタリングの処理方法を動的に変換する機能をもち，状況に応じて最適な処理方法を自動的に選択するフィルタリングシステムである．処理方法選択の際には，フィルタリングを関数として表現したフィルタリング関数[7, 8]の数学的性質を利用することで，フィルタリング結果の一貫性を保証する．また，本システムは，フィルタリング処理にアクティブデータベース[11]の動作記述言語であるECAルールを用いて，ネットワーク帯域やクライアントのCPU利用率の変化を自動的に検出し，コストが低くなる処理方法を選択する機構を実現する．本システムを用いることで，さまざまな環境において常に効率的なフィルタリング処理が行えるようになる．

以下，第2章では，本研究で構築するフィルタリングシステムの設計について説明し，第3章でシステムの実装について述べる．第4章で本システムについて考察し，最後に第5章でまとめを行う．

2. システムの設計

本研究で構築する情報フィルタリングシステムは，ユーザの要求に対してフィルタリング結果が等価となる処理方法を判断し，状況に応じて最適な処理方法を自動的に選択する．本システムの概要を図 1 に示す．本システムは以下の流れで動作する．

1. ユーザのフィルタリングに関する要求をプロファイルとして記述する．本システムは，プロファイル記述言語としてフィルタリング SQL[6]を用いる．
2. 1.の要求を実現するフィルタリングに対し，フィルタリング関数[7, 8]の数学的性質を利用することで，フィルタリング結果が等価となる処理方法を決定する．
3. 2.で決定した等価な各処理方法について，それらの処理を実際に行うための ECA ルール群を生成する．
4. 2.で決定した等価な処理方法のうち，クライアントの CPU 利用率やネットワークの負荷など現在の環境において最適な処理方法を選択する．そして，3.で作成した ECA ルールのうち，選択した処理方法を行う ECA ルールを用いて受信データをフィルタリングする．システム稼動中に環境が変化した場合，その変化に対応して処理方法を変更するルールにより，常に最適な処理方法を選

♦学生会員 大阪大学大学院情報科学研究科
{kodera,rie}@ist.osaka-u.ac.jp
♠ 正会員 大阪大学大学院情報科学研究科
{tsutomu,tuka,nishio}@ist.osaka-u.ac.jp

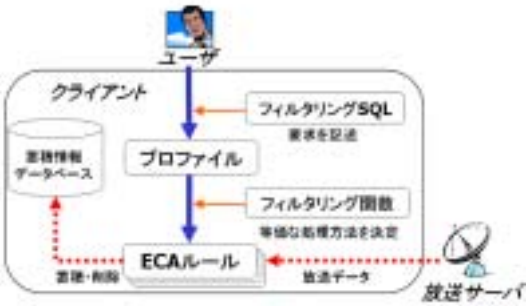

図 1　システムの概要

択する．

5. フィルタリングにより必要と判断されたデータを蓄積情報データベースに格納する．

以下，システムの設計について，上記の流れに沿って詳細に述べる．

2.1 フィルタリング SQL

本システムでは，ユーザの要求を記述するためのプロファイル記述言語として，フィルタリング SQL を用いる．フィルタリング SQL とは，データベースへの問合せ言語である SQL(Structured Query Language)に，フィルタリングのための記述構文を加えたものである．以下の記述例は「A 放送または B 放送から受信したデータのうち，属性“GENRE”の値が“ニュース”であるデータのみを蓄積したい」というユーザの要求を表す．

```
EXTRACT    *
FROM       A 放送, B 放送
WHERE      GENRE = “ニュース”
```

2.2 フィルタリング関数

本システムは，逐次処理や一括処理といった処理方法を変換してもフィルタリング結果の一貫性を保証するために，フィルタリング関数[7, 8]の数学的性質を利用する．フィルタリング関数の枠組みでは，関数が満たす制約条件によってフィルタリングの性質が表され，逐次処理と一括処理のフィルタリング結果が等価である(逐次等価性)といったように，処理方法に関するいくつかの性質が定義されている．したがって，例えば，あるフィルタリング手法を表すフィルタリング関数が逐次等価性を満たすならば，そのフィルタリング手法では，逐次処理と一括処理の間で処理方法を変換しても一貫したフィルタリング結果が得られることを保証できる．

フィルタリング関数の体系では，現在のところ次の4つの処理方法を取り扱っている．データが受信機に到着するたびに，その受信データと前回までに蓄積した結果を合せてフィルタリングする処理方法を**逐次処理**と呼ぶ．また，受信データをある程度ためておき，一括してフィルタリングする処理方法を**一括処理**と呼ぶ．複数の受信機でデータを分けて受信し，並列にフィルタリングした後，それらの結果を合せたものをフィルタリング結果とする処理方法を**分配処理**，分配処理の結果をさらにフィルタリングする処理方法を**並列処理**と呼ぶ．

また，フィルタリング関数の体系では，さまざまなフィルタリング手法を取り扱っているが，代表的なフィルタリング手法として次のようなものがある[9]. データ間の相関性を考慮せず，コンテンツや属性などからデータごとに取捨選択を決定するフィルタリング手法をセレクションと呼ぶ．例えば，ユーザが指定したキーワードを含むデータのみを蓄積するキーワードマッチングや，評価値が閾値以上であるデータのみを蓄積するフィルタリングなどがセレクションである．また， ユーザの嗜好に応じて重要度の高い順にデータを並べ，より重要度の高いデータから特定の数だけを蓄積するフィルタリング手法をランキングと呼ぶ．さらに，セレクションとランキングを組合せることで，より複雑な処理ができる．例えば，ある単語についてキーワードマッチングを行った後に，別の単語についてキーワードマッチングを行うセレクション同士を組合せた手法や，受信日時が最新100のデータを抽出した後に，評価値がベスト10のデータを抽出するといったような，異なる属性を基準としたランキング同士を組合せた手法などが存在する．

表1 各フィルタリング手法における等価な処理方法

フィルタリング手法			等価処理
セレクション			逐次処理 一括処理 分配処理 並列処理
ランキング			逐次処理 一括処理 並列処理
セレクション→セレクション			逐次処理 一括処理 分配処理 並列処理
ランキング(蓄積数n) →ランキング(蓄積数n')	同じ属性が基準		逐次処理 一括処理
	異なる属性が基準	n≤n'	分配処理
		n>n'	なし
セレクション→ランキング			逐次処理 一括処理 分配処理
ランキング→セレクション			なし

各フィルタリング手法においてフィルタリング結果が等価となる処理方法を表1に示す．証明など導出に関する詳細は文献[7, 8, 9]に記述している．表1の“手法1→手法2”は，手法1で処理を行った後に手法2で処理するフィルタリングを表す．また，“n≤n'”は，1つ目のランキング処理で蓄積するデータ数(n)が2つ目のランキング処理で蓄積するデータ数(n')以下であるフィルタリング手法を表す．

2.3 ECA ルール

本システムでは，ECA ルール[11]を用いて実際のフィルタリング処理を記述する．ECA ルールとは，データベース技術の一つであるアクティブデータベースの動作記述言語であり，発生する事象(Event)，アクションを実行させるための制約条件(Condition)，実行する動作(Action)の 3 つの組からなる．ECA ルールはイベント駆動型であるため，システム環境の変化やデータの到着といった状況の変化に対応した柔軟な処理が記述できる[5]. また，システムの機能は ECA ルールの集合で表現されるため，ECA ルールを部分的に追加・削除することで，フィルタリングポリシーのカスタマイズや処理方法の変更が容易になる．さらに，ECA ルールを送信することで，他の端末に処理機能やフィルタリングのポリシーを移動できる． 表 2，表 3 に本システムのフィルタリング処理に必要なイベント，アクションを示す．これらを基に，CPU 利用率，ネットワークの負荷の変化を検知するイベントなどを追加することで，環境の変化に応じた処理方法変換が可能となる．

本システムでは，受信データを取捨選択するフィルタリング処理と，環境に応じて最適なフィルタリング処理方法へ変換する処理に ECA ルールを用いる．以下，2 つの処理の実現方法について述べる．

表 2 イベント一覧

名称	内容
SELECT	テーブルのデータ参照
INSERT	テーブルのタップル挿入
DELETE	テーブルのタップル削除
META_RECEIVE	メタデータの受信
CONTENT_RECEIVE	コンテンツデータの受信
RULE_RECEIVE	ルールの受信
NET_RECEIVE	データパケットの受信
TIMER	設定したタイマの発火

表 3 アクション一覧

名称	内容
QUERY	データベース操作
SETTIMER	タイマの設定
BROADCAST_SEND	パケットのブロードキャスト
FILE_SEND	ファイル送信
STORE_FILE	ファイルの蓄積
DELETE_FILE	ファイルの削除
ADD_RULE	ルールの追加
ENABLE_RULE	ルールの有効化
DISABLE_RULE	ルールの無効化

2.3.1 フィルタリングのための ECA ルール

フィルタリング SQL 記述は，その記述構文に従って ECA ルールへ変換する．構文のパターンはフィルタリング関数の体系によっていくつかに分類されるため，パターンに応じて各処理方法を実現するための ECA ルールを生成することは容易にできる．2.1 節で記述したフィルタリング SQL はセレクションによるフィルタリングに分類されるため，逐次処理，一括処理，分配処理，並列処理が等価となり，例えば一括処理で実現する ECA ルールは次のようになる．

```
ルール 1    E  META_RECEIVE
            A  QUERY(“INSERT INTO 一時テーブル”)
ルール 2    E  TIMER
            A  QUERY(“SELECT * FROM 一時テーブル”)
ルール 3    E  SELECT 一時テーブル
            C  NEW.RESOURCE = A 放送
               OR NEW.RESOURCE = B 放送
               AND NEW.GENRE = ニュース
            A  QUERY(“INSERT INTO 蓄積テーブル”)
               STORE_FILE
```

“NEW”とは，メタデータに対するイベントが発生したときに，そのメタデータの内容が格納される変数であり，“NEW.属性”によりメタデータの属性値を参照できる．ルール 1 は，メタデータを受信したとき，その内容を一時テーブルに格納することを表す．ルール 2，ルール 3 は，設定したタイマが発火したとき，一時テーブルの中から“RESOURCE”が“A 放送”または“B 放送”であり，“GENRE”が“ニュース”であるメタデータを蓄積テーブルに格納し，コンテンツデータをディスクに蓄積することを表す．

逐次処理は，上記の ECA ルールにおいて，タイマを用いず，メタデータを受信するたびにフィルタリング処理を行うことにより実現できる．また，分配処理，並列処理は，依頼先端末に依頼処理のための ECA ルールを追加することで同様に実現できる．

2.3.2 処理方法変換のための ECA ルール

本システムでは，フィルタリング関数の数学的性質により明らかとなった等価な処理方法に対して，最適な処理方法を選択するための処理も ECA ルールで実現する．以下に処理方法変換のための ECA ルールの例を示す．

```
E  TIMER
C  CPU_USAGE >= 50%
A  DISABLE_RULE 逐次処理
   ENABLE_RULE 一括処理
```

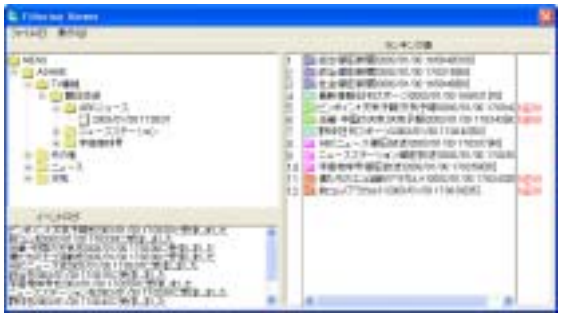

図 2 蓄積データのビューア

このルールは，逐次処理を行っているとき，タイマで一定時間ごとに CPU 利用率をチェックして，CPU 利用率が 50% 以上であれば一括処理へ変換する ECA ルールである．

本システムでは，CPU 利用率以外にも，受信機の受信コストなど処理方法変換のためのさまざまな環境パラメータを設定することで，環境に応じて柔軟に処理方法を変換できる．

3. 実装

前章で説明した情報フィルタリングシステムのプロトタイプを実装した．実装は， WindowsXP 上で Microsoft 社の VisualBasic6.0，Visual C++6.0 を用いて行った．蓄積情報データベースとしては Microcoft 社の Access 2000 を用いた．また，放送するコンテンツとしては，地上波データ放送である ADAMS[1]と bitcast[3]を用いた．

クライアントシステムは，ECA ルールを処理するルールエンジン，メタデータおよび蓄積データを管理するデータベース，蓄積データを参照するためのビューア(図 2)，ユーザが要求を入力するエディタから構成される．ルールエンジンには，プラグインを追加することで記述できるイベントやアクションを自由に拡張できる A-wear[5]を用い，本システムで新たに必要となる機能であるメタデータの解析，受信ファイルの識別，ファイルの格納・削除を実行するプラグインを実装した．

4. 考察

4.1 処理方法が処理コストに与える影響

逐次処理では，継続的に受信する大量のデータを逐一処理しなければならないため，処理に要する計算時間や計算能力などのコストが高くなる．したがって，逐次処理の途中に受信機の処理能力が下がったときは，継続的に受信データをフィルタリングする必要がない一括処理に変換することで，処理コストを低減できる．また，複数の端末が利用可能なときは，分配処理や並列処理に変換することでデータの受信処理を分散できるため，受信のためのコストやネットワークの負荷を軽減できる．逆に，一括処理や分配処理では，ある一定期間に受信したデータをためておいた後にフィルタリングするため，結果が利用できるようになるまで時間がかかるが，逐次処理に変換することで，ユーザはリアルタイムに受信データを利用できる．このように，本システムでは，環境に応じて最適な処理方法を選択できる．

4.2 関連研究

ユーザと放送源の中間に位置するサーバがベクトル空間モデルを用いてフィルタリングを行う方式[10]がある．[10]のシステムは，転置リストに改良を加えてプロファイル検索を効率的に行う手法や，演算の順序を変え，行列を入れ換えることにより複数のファイルを一括して取捨選択する手法を組合せることで，CPU 利用率やメモリ使用量を減らしている．また，ベクトル空間モデルの次元数を減少させることで，処理の効率化を図ったモデルもある[4]．これらのフィルタリングはアルゴリズムの面から処理コストを軽減しているが，本システムでは，処理方法の面から処理コストを軽減する方法をとっている．また，これらの関連研究は，一括処理や分配処理といった処理方法を動的に変更することを考慮していないため，環境の変化に対応して常に効率的な処理をするのが困難である．本システムは，環境の変化に対応して常に処理コストが低くなる処理方法へ動的に変換する機構をもつ．さらに，フィルタリング関数の数学的性質を利用することで，処理方法変換後も一貫したフィルタリング結果が得られることを保証している点がこれまでの研究と異なる点である．

5. おわりに

本研究では，フィルタリング関数の数学的性質を利用することで，フィルタリングの処理中により効率的な処理方法へと変換しても一貫したフィルタリング結果が得られる情報フィルタリングシステムを構築した．本システムは，受信機の処理能力が低下したときは一括処理，複数の端末が使えるときは分配処理といったように，環境に応じて動的に処理方法を変換することで処理コストを軽減できる．

今後は、大量のデータを受信する環境やモバイル環境などさまざまな環境において，処理に要する計算時間や計算能力などの変化を測定し，処理方法を動的に変換しない従来のシステムと比較することで，本システムの有効性を示す予定である．また，現在のシステムは，処理方法変換のための環境パラメータに経験的に求めた値を用いている．そこで，最適な処理方法変換機構を実現するため，さまざまな環境における処理コストを統計的・数学的に明らかにすることで，環境パラメータの最適値を求め，その妥当性を検討する予定である．

[謝辞]

本研究は，文部科学省振興調整費「情報フィルタリングの数学的基盤の確立」,「モバイル環境向 P2P 型情報共有基盤の確立」，科学研究費補助金(基盤研究(B)(2))「大規模な仮想空間システムを構築する放送型サイバースペースに関する研究」(プロジェクト番号:15300033)，および文部科学省 21 世紀 COE プログラム(研究拠点形成費補助金)の研究助成によるものである．ここに記して謝意を表す．

[文献]

[1] ADAMS Homepage: http://www.tv-asahidata.com/.

[2] N. J. Belkin and W. B. Croft: “Information filtering and information retrieval: two sides of the same coin?,” *Comm. ACM*, vol. 35, no. 12, pp. 29-38 (1992).

[3] Bitcast Homepage: http://www.bitcast.ne.jp/.

[4] 原田晃史，川越恭二: “ベクトル空間圧縮モデルによる WWW 検索処理の効率化,” 情報処理学会研究報告(データベースシステム研究会 99-DBS-119)，vol. 99, no. 61, pp.91-96 (1999).

[5] 宮前雅一，中村聡史，寺田努，塚本昌彦，西尾章治郎: “ウェアラブルコンピューティングのための拡張可能なルール処理システム,” 情報処理学会研究報告(情報家電コンピューティング研究グループ 2002-IAC-3), pp. 41-46 (2002).

[6] 澤井里枝，寺田努，塚本昌彦，西尾章治郎: “フィルタリング SQL: フィルタリングのためのユーザ要求記述言語,” 電子情報通信学会第 11 回データ工学ワークショップ(DEWS2000)論文集 (CD-ROM) (2000).

[7] R. Sawai, M. Tsukamoto, Y. H. Loh, T. Terada, and S. Nishio: “Functional properties of information filtering,” in *Proc. VLDB2001*, pp. 511-520 (2001).

[8] 澤井里枝，塚本昌彦，寺田努，Loh Yin Huei，西尾章治郎: “情報フィルタリングの関数的性質について,” 電子情報通信学会論文誌 D-I, vol. J85-D-I, no. 10, pp. 939-950 (2002).

[9] 澤井里枝，塚本昌彦，寺田努，西尾章治郎: “フィルタリング関数におけるセレクションとランキングについて,” 情報処理学会論文誌:データベース, vol. 43, no. SIG12(TOD16), pp. 80-91 (2002).

[10] A. H. Timothy and M. Alistair: “The design of a high performance information filtering system,” in *Proc. SIGIR1996*, pp. 12-20 (1996).

[11] J. Widom and S. Ceri: “Active database system,” *Morgan Kaufmann Publishers Inc.* (1996).

小寺 拓也 Takuya KODERA

2003年大阪大学工学部電子情報エネルギー工学科卒業. 現在, 同大学院情報科学研究科マルチメディア工学専攻博士前期課程に在学中．日本データベース学会学生会員.

澤井 里枝 Rie SAWAI

2002年大阪大学大学院工学研究科博士前期課程修了．現在, 同大学院情報科学研究科マルチメディア工学専攻博士後期課程に在学中．日本データベース学会学生会員.

寺田 努 Tsutomu TERADA

1999年大阪大学大学院工学研究科博士前期課程修了．2000年同大学院工学研究科博士後期課程退学．同年より大阪大学サイバーメディアセンター助手, 現在に至る．2002年より同大学院情報科学研究科マルチメディア工学専攻助手を併任．アクティブデータベース，データ放送の研究に従事．電子情報通信学会, 情報処理学会, 日本データベース学会の各会員.

塚本 昌彦 Masahiko TSUKAMOTO

1989年京都大学大学院工学研究科修士課程修了. 同年シャープ（株）に入社，同社研究員．1995年大阪大学大学院工学研究科講師．1996年より同大学院工学研究科助教授，2002年より同大学院情報科学研究科マルチメディア工学専攻助教授，現在に至る．工学博士．モバイルコンピューティング，分散知識ベースシステムの研究開発に従事．ACM，IEEE，日本データベース学会等9学会各会員.

西尾 章治郎 Shojiro NISHIO

1980 年京都大学大学院工学研究科博士後期課程修了．工学博士．京都大学工学部助手，大阪大学基礎工学部および情報処理教育センター助教授，大阪大学大学院工学研究科情報システム工学専攻教授を経て，2002 年より同大学院情報科学研究科マルチメディア工学専攻教授となり，現在に至る．2000 年より大阪大学サイバーメディアセンター長を併任．データベース, マルチメディアシステムの研究に従事. 現在, ACM Trans. on Internet Technology などの論文誌編集委員. 本学会理事，情報処理学会フェロー含め，9 学会の会員.