

プッシュ型放送システムにおける送受信方式の モニタリングおよびチューニングが可能なツールの設計と実装

前田 和彦[†] 内田 渉^{††} 原 隆浩^{††} 西尾章冶郎^{††}

[†] 大阪大学工学部情報システム工学科 〒 565-0871 吹田市山田丘 2-1

^{††} 大阪大学大学院情報科学研究科マルチメディア工学専攻 〒 565-0871 吹田市山田丘 2-1

E-mail: [†]maeda@ise.eng.osaka-u.ac.jp, ^{††}{wataru,hara,nishio}@ist.osaka-u.ac.jp

あらまし 近年, 有線および無線の通信技術の発展に伴い, 放送型通信を用いてデータを配送するプッシュ型情報システムに関する研究の関心が高まっている. 本稿では, プッシュ型放送を行うシステムにおいて, モニタリングやチューニングが可能なサーバツールおよび, クライアントツールの設計を行う. このツールでは, サーバの管理者やクライアントの利用者が, 双方でシステムの動作や応答時間などの性能指標を視覚的なモニタリングにより調査し, 必要に応じてスケジューリングやキャッシング方式の変更およびチューニングを行うことによって, システム環境や各々のユーザの要求に応じた性能を得ることを可能とする. さらに本稿では, このような設計に基づいたツールのプロトタイプの実装について述べる.

キーワード プッシュ型情報システム, データ放送, モニタリング, チューニング

Design and Implementation of Tools with Monitoring and Tuning Mechanisms for Scheduling and Caching Strategies in Push-based Broadcast Systems

Kazuhiko MAEDA[†], Wataru UCHIDA^{††}, Takahiro HARA^{††}, and Shojiro NISHIO^{††}

[†] Dept. of Information Systems Eng., Faculty of Engineering, Osaka Univ.

^{††} Dept. of Multimedia Eng., Graduate School of Information Science and Technology, Osaka Univ.
2-1 Yamadaoka, Suita, Osaka 565-0871, Japan

E-mail: [†]maeda@ise.eng.osaka-u.ac.jp, ^{††}{wataru,hara,nishio}@ist.osaka-u.ac.jp

Abstract Recently, there has been an increasing interest in research of push-based information systems that deliver data using broadcast in both wired and wireless environments. In this paper, we design server and client tools with monitoring and tuning mechanisms for scheduling and caching strategies in push-based broadcast systems. These tools enable server administrators and client users to monitor visually the behavior of the system and the performance criterias such as response time and to tune scheduling and caching strategies based on the monitoring results. As a result, they can improve the performance of the system according to the system environment and each user's requirements. We also describe our implementation of the prototypes of the designed tools.

Key words push-based information system, data broadcast, monitoring, tuning

1. はじめに

近年, 有線および無線の通信技術の発展に伴い, 放送型通信を用いてデータを配送するプッシュ型情報システム(以下では単にプッシュ型情報システムと呼ぶ)に関する研究の関心が高まっている. プッシュ型情報システムでは, クライアントはデータアクセスの際, アクセス要求をサーバへ送信せずにサー

バの放送帯域を監視し, そのデータアイテムが放送された時点でアクセスを完了する(図1). クライアントからの要求に応じて個々のデータを放送するオンデマンド型情報システムとは異なり, プッシュ型情報システムでは, サーバは各クライアントから離散的に発生する要求を一度の放送により満たすことができるため, クライアント数が増加してもシステム全体の負担コストはほとんど変わらない. また, アクセス要求がサーバへ送

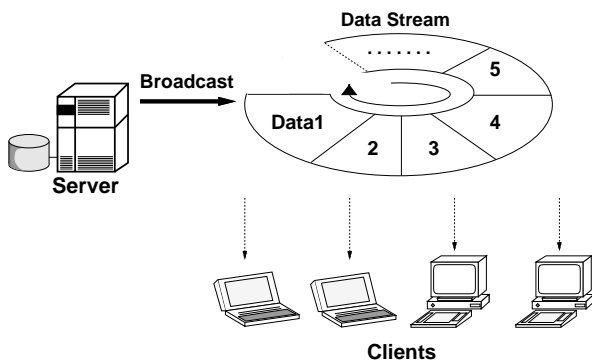


図 1 プッシュ型情報システム

Fig. 1 A push-based information system .

信されないため、クライアント数が増加した場合の上り帯域での競合も発生しない。したがって、クライアント数が非常に多い環境において、データアクセスのスループットの向上が期待できる。

これまでに、プッシュ型情報システムの性能を向上するためにサーバ側でのスケジューリング方式 [1], [3] ~ [6], [10], [13], [?], [14], [16] ~ [18] および、クライアント側でのキャッシング方式 [2], [11], [12], [15] が多数提案されている。クライアントのデータアクセスの平均応答時間を短縮するための方式として、最も代表的なものに、Vaidya らが提案したスケジューリング方式 [16] や、Acharya らが提案した PT 法と呼ばれるキャッシング方式 [2] がある。

これらの方式は、統計により得られる既知のアクセス特性に基づいてスケジューリングおよびキャッシングを行う。例えば、Vaidya らが提案したスケジューリング方式では、各データアイテムに対するシステム全体のアクセス確率に基づいて、頻繁にアクセスされるデータアイテムを頻繁に放送することによって平均応答時間を短縮する。PT 法では、アクセス確率と次回の放送までの時間の積で表される応答時間の期待値が大きいデータアイテムをキャッシュし、平均応答時間を短縮する。

ここで、このような従来研究では、下記に示すように、ある仮定の下で行った評価によってのみ、性能向上を確認している。

- 放送プログラムは固定とし、サーバはそれに従って周期的に放送を行う。
- データアイテムのサイズは均一である。
- クライアントのアクセス確率は時間的に変化しない。
- データアイテムの追加・削除は行わない。

しかし、これらの個々の仮定は、実環境では適当でない場合があるため、各方式を実環境へ適用したときに、期待される性能を示すとは限らない。

また、これらの方式は、平均応答時間など特定の性能指標の改善を行うため、特定の方式によってスケジューリングおよびキャッシングを行った場合、必ずしもシステム環境やユーザの要求に応じた性能が得られるとは限らない。例えば、Vaidya らの方式を用いて、システム全体のアクセス確率が高いデータアイテムを頻繁に放送した場合、システム全体と嗜好が異なるクライアントの平均応答時間は非常に長くなってしまふ。データ

アクセスの待ち時間にデッドラインが存在するような場合、そのような状況は望ましくない。

また、統計によって得られたクライアントのアクセス特性に基づいて、従来方式をそのまま使用した場合、システム全体のアクセス特性に関係なく一部のデータアイテムは頻繁に放送したい、今までのアクセス履歴に関わらず特定のデータアイテムをキャッシュしやすくしたい、という要求に対応することができない。例えば、楽曲データをプッシュ型配信をするシステムにおいて、新発売された知名度の低い曲を宣伝のために頻繁に放送したいというサーバ側の意図や、新しく興味をもった楽曲をキャッシュしやすくして欲しいといったクライアント側の意図を反映することができない。

このように、従来の方式を実環境に適用する際、適用する環境によって最適な方式を選択する必要がある。また、サーバ管理者やクライアント利用者などの、ユーザ毎にどの性能指標を重視するか、どのアイテムを重視するかといった嗜好は異なるため、あらゆるユーザの要求に答えられるように、特定のデータアイテムの重要度を変更するなど、方式の動作を調整できるようにする必要がある。しかし、システムの特長やユーザの嗜好は多岐にわたり、さらに動的に変化するため、サーバやクライアントシステムが自動的に調整することは非常に困難である。

これまでに、プッシュ型情報システムを実運用することが可能なミドルウェアも提案されているが [8]、実運用中でモニタリングを行ったり、放送やキャッシングのチューニングを行う機能は実現されていない。

そこで本稿では、プッシュ型情報システムにおけるモニタリングやチューニングが可能なサーバツールおよびクライアントツールを提案する。このツールでは、サーバ管理者やクライアント利用者が、双方でスケジューリングやキャッシングの動作や応答時間、ヒット率などの性能指標を視覚的にモニタリングし、必要に応じてスケジューリング、キャッシング方式の変更やチューニングを行うことができる。これにより、システム環境や個々のユーザの要求に応じた性能を得ることが可能となる。さらに本稿では、この設計に基づいて実装した提案ツールのプロトタイプについても述べる。

以下、2. でこれまでに提案されているスケジューリング方式およびキャッシング方式について述べる。その後、3. で提案するツールの設計について述べ、4. で提案ツールのプロトタイプの実装について説明する。最後に 5. で本稿のまとめを行う。

2. 従来方式

プッシュ型情報システムの性能向上を目指す研究の一環として、これまでに様々なサーバ側におけるスケジューリング方式やクライアント側におけるキャッシング方式が提案されている。本章では、それらのうち、Vaidya らのスケジューリング方式 [16] や放送ディスク [1], PIX 法 [1], および PT 法 [2] の概要、そしてこれらの方式を実環境に適用したときの問題点について説明する。以下では、各データアイテムを $1, \dots, M$ の識別子を用いて区別する。 M はサーバが放送するデータアイテムの総数である。

2.1 スケジューリング方式

2.1.1 Vaidya らのスケジューリング方式

(1) 各データアイテムの放送開始時に、全てのデータアイテム j に次式で表される評価関数 $G(j)$ を与える．

$$G(j) = (Q - R(j))^2 \cdot P_j / l_j, \quad 1 \leq j \leq M \quad (1)$$

ただし、 Q は現在時刻、 $R(j)$ はデータアイテム j が一番最後に放送された時刻、 P_j はデータアイテム j に対するシステム全体のアクセス確率、 l_j はデータアイテム j のサイズとする．

(2) 放送するデータアイテムの中で $G(j)$ が最大となるアイテム i を探す． $G(j)$ が最大となるデータアイテムが複数存在する場合は、それらの中から無作為に一つのデータアイテム i を選択する．

(3) 時刻 Q に、データアイテム i を放送する．

(4) $R(i) = Q$ とする．

ここで、 $Q - R(j)$ はデータアイテム j が一番最後に放送された時刻から経過した時間を表している．すなわち、前回放送されてからの経過時間が大きくなれば、評価関数 $G(j)$ が大きくなるため、放送されやすくなる．また、評価関数 $G(j)$ に P_j / l_j が含まれているため、システム全体のアクセス確率が大きく、データサイズが小さいものが優先的に放送される．

2.1.2 放送ディスク

放送ディスクでは、次の手順によって放送プログラムを作成し、それに従って周期的な放送を行う．

(1) 各データアイテムを、システム全体のアクセス確率に基づいて、降順に整列する．

(2) アクセス確率が近いデータアイテムをまとめ、全データアイテムを N 個の集合に分割する．各集合をディスクと呼び、 D_1, D_2, \dots, D_N と表す．

(3) 各ディスク D_i 内に含まれる全アイテムの放送頻度を整数で決定し、 D_i の放送頻度を f_i とする．例えば、全アイテムが2つのディスク D_1, D_2 に分割され、 $f_1 = 3, f_2 = 2$ となるとき、 D_1 が3回放送されるごとに、 D_2 は2回放送される．

(4) 各ディスクを、いくつかの小部分に分割する．各々の小部分をチャンクと呼ぶ．まず、全ディスクの放送頻度 f_i の最小公倍数をとり、 T_{max} とする．各ディスク D_i のチャンク分割数を $T_i = T_{max} / f_i$ とする． D_i を分割したチャンクのうち、 j 番目のものを $C_{i,j}$ と表す．チャンクに含まれるデータアイテム数 $N_{chunk}(C_{i,j})$ は $|D_i| / T_i$ とする．ここで、 $|D_i|$ は、ディスク D_i に含まれるデータアイテム数である．

(5) 次のアルゴリズムにしたがって、放送プログラムをディスク内のチャンクを1つずつ配置することで決定する．

```
01 for i:= 1 to  $T_{max}$ 
02   for j:= 1 to  $N$ 
03     Broadcast chunk  $C_{j,(i \bmod T_j)}$ 
04   endfor
05 endfor
```

ここで、Broadcast chunk $C_{j,(i \bmod T_j)}$ とは、ディスク D_j の $i \bmod T_j$ 番目のチャンクを放送することを示す．

放送ディスクでは、スケジューリングにより放送プログラム

を決定されれば、アクセス確率が変化しない限り、同じ放送プログラムが繰り返し放送される．そのため、アイテムが放送される度にスケジューリングを行う Vaidya らの方式に比べて、計算量は少ない．しかし、手順 (2) でのディスクの分け方やディスク数 N の決定方法、そして手順 (3) の放送頻度の求め方については特に定義されていない．

2.1.3 これらのスケジューリング方式の問題点

これらの平均応答時間の短縮に着目したスケジューリング方式は、クライアントのアクセス確率を一定と仮定している．しかし、実環境のようにアクセス確率が時間的に変化する場合、それを考慮してアクセス変化が起こる度にシステムに登録されているアクセス確率の情報を再調整するか、時間的な変化を考慮する他の方式への変更を行う必要がある．また、放送ディスクは各データアイテムのサイズも一定と仮定しているため、実環境のように各データアイテムのデータサイズがそれぞれ異なる場合に、期待されるような性能を示すとは限らない．そのような場合、必要に応じてデータサイズを考慮した放送頻度の調整を行う必要がある．

2.2 キャッシング方式

2.2.1 PIX 法

(1) 受信開始時に、全てのデータアイテム j に次式で表される評価関数 $K(j)$ を与える．

$$K(j) = p_j \cdot x_j \quad (2)$$

ただし、 p_j はクライアントのデータアイテム j に対するアクセス確率、 x_j はデータアイテム j の放送間隔である．

(2) すべてのデータアイテムの中で、 $K(j)$ の値が高いものをキャッシュサイズだけキャッシュする．

PIX 法では、クライアントのアクセス確率が高く、放送間隔が大きいデータアイテムをキャッシュすることで、頻繁にアクセスするが、あまり放送されないデータアイテムの応答時間の短縮を図っている．また、クライアントのアクセス確率が一定と仮定しているため、一度データアイテムをキャッシュしてしまえば、キャッシュ内に存在するデータアイテムは変化しない．しかし、データの更新が頻繁に発生するような環境においては、それを考慮してキャッシュ内のデータアイテムの更新を行う必要がある．

2.2.2 PT 法

(1) 各データアイテムの放送開始時に、キャッシュ内のデータアイテムおよび放送されるデータアイテムに PT 値と呼ばれる値を与える．データアイテム j に与える PT 値は次式で表される．

$$L_j = p_j \cdot (u_j(Q) - Q), \quad 1 \leq j \leq M \quad (3)$$

ただし、 $u_j(Q)$ は時刻 Q におけるデータアイテム j の次回の放送時刻とする．

(2) 放送されるデータアイテム b の PT 値 L_b が、キャッシュ内で PT 値が最小となるデータアイテム j の PT 値 L_j より大きい場合、データアイテム b と j を置き換える．キャッシュ内で PT 値が最小となるデータアイテムが複数存在する場合は、

それらの中から無作為に一つのデータアイテムを選択する。

PT 値はあるデータアイテムに対してアクセス要求を発行したとき、そのデータアイテムをキャッシュに保持していない場合に生じる待ち時間の期待値を示している。すなわち、PT 法では、各データアイテムの放送開始時に、各々のデータアイテムをキャッシュから追い出すことにより増加する待ち時間を比較し、最も応答時間の利得が大きいにキャッシュを管理する。

2.2.3 これらのキャッシング方式の問題点

これらの方式では、クライアントのアクセス確率は一定と仮定している。したがって、実環境において、そのような仮定が成立しないアプリケーションに適用する場合、アクセス確率の変化を考慮してチューニングを行う必要がある。

またこれらの方式は、キャッシュ内に蓄えることができるデータアイテム数は考慮しているが、データアイテムのサイズについては考慮していない。データアイテムのサイズはアイテムごとに異なる場合が一般的なため、キャッシングにおける評価値の計算の際には、アクセス確率などに加えて、データアイテムのサイズも考慮し、データアイテム数ではなく、データ容量をキャッシュサイズとして、その中で効率的な管理を行うように調整を行う必要がある。

3. ツールの設計

本章では、提案するツールの設計について説明する。提案ツールを用いたプッシュ型情報システムのシステム構成を図2に示す。ツールはサーバ側およびクライアント側に分かれ、それぞれに通信部とモニタリング・チューニング部が存在する。

サーバ側ツールの通信部は、モニタリング・チューニング部からのスケジューリング方式の変更やチューニングの指示に従ってスケジューリングを行い、各データアイテムをクライアント集合に放送する。クライアント側ツールの通信部は、モニタリング・チューニング部からのキャッシング方式の変更やチューニングの指示に従い、サーバにより放送されたデータアイテムをキャッシュする。また、アプリケーションから発行されたアクセス要求に従い、放送帯域からの受信もしくはキャッシュアクセスにより、要求されたデータアイテムをアプリケーションに渡す。さらに、サーバ側ツール、クライアント側ツールともに、通信部では、通信や方式の動作のログを作成し記録・管理する。

モニタリング・チューニング部は、サーバ側ツール、クライアント側ツールともに通信部で作成されたログを基にそれらの情報を視覚化し、ユーザに表示する。ユーザはそれらの情報によって、現在の通信や方式の動作がユーザの意図する通りに行っているかを確認することができる。また、ユーザは必要に応じて、モニタリング・チューニング部を通して方式の変更やチューニングを通信部に指示することができる。サーバ側とクライアント側の各ツールの仕様を以下に示す。

3.1 サーバ側ツールの仕様

3.1.1 通信部

(1) モニタリング・チューニング部で指定された方式に従い、クライアント集合にデータアイテムの放送を行う。

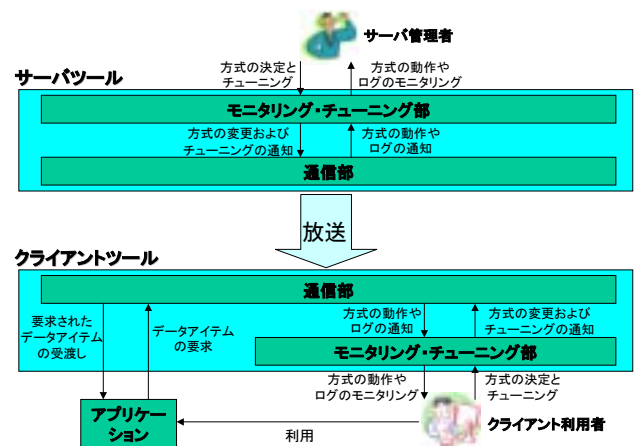


図2 ツールを用いたシステムの構成

Fig. 2 Components of a system using the proposed tools.

(2) 現在の通信状況や指定されたスケジューリング方式の動作、またシステムの性能指標に関するログを作成し記録する。性能指標の例としては、各データアイテムの放送頻度やシステム全体の平均応答時間の予測値などがある。また、各データアイテムの詳細情報（優先度や、ファイルをデータアイテムとする場合はファイル名、バイト数など）も管理する。

(3) 管理しているログを、要求に応じてモニタリング・チューニング部に渡す。

3.1.2 モニタリング・チューニング部

(1) 通信部より受け取った、システムの性能指標に関する情報をモニタリングし、視覚的に表示する。

(2) サーバ管理者の要求に応じて、スケジューリング方式を変更する。

(3) スケジューリング方式の動作をモニタリングし、視覚的に表示する。例えば、データアイテムごとのアクセス確率や放送頻度などをグラフ化する。

(4) サーバ管理者の要求に応じて、スケジューリング方式の動作をチューニングする。例えば、データアイテムごとの優先度を決定することにより、スケジューリングの微調整を可能とする。

(5) チューニング操作や方式の変更を通信部に通知する。

3.2 クライアント側ツールの仕様

3.2.1 通信部

(1) モニタリング・チューニング部で指定された方式に従い、サーバから放送されるデータアイテムをキャッシュする。

(2) アプリケーションからの要求に応じて、データアイテムをサーバから受信しアプリケーションに渡す。

(3) 現在の通信状況や指定されたキャッシング方式の動作、システムの性能指標に関するログを作成し記録・管理する。性能指標の例としては、データアイテムごとの放送間隔や、平均応答時間、ヒット率などが挙げられる。また、データアイテムごとの詳細情報（優先度や、ファイルをデータアイテムとする場合はファイル名、バイト数など）も管理する。

(4) 管理しているログを、要求に応じて、モニタリング・チューニング部に通知する。

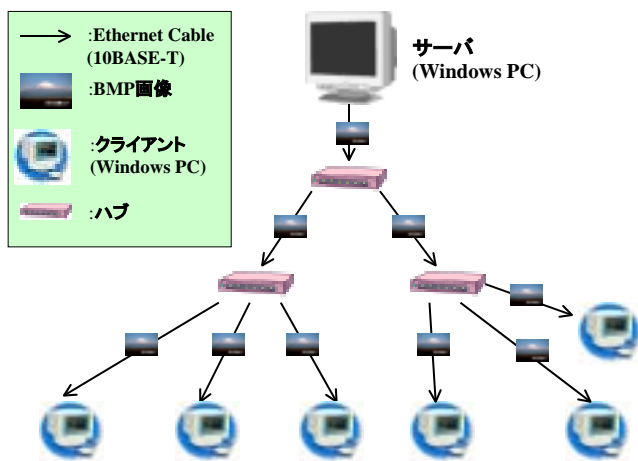


図3 システム環境
Fig. 3 System environment.

3.2.2 モニタリング・チューニング部

(1) 通信部より受け取った、システムの性能指標に関する情報をモニタリングし、視覚的に表示する。

(2) クライアント利用者の要求に応じて、キャッシング方式を変更する。

(3) キャッシング方式の動作をモニタリングし、視覚的に表示する。例えば、これから放送されるアイテムの順序や各アイテムのアクセス頻度、キャッシュ内容を視覚的に表示する。

(4) クライアント利用者の要求に応じて、キャッシング方式の動作をチューニングする。例えば、各データアイテムの優先度を決定することにより、キャッシングの微調整を可能とする。

(5) チューニング操作や方式の変更を通信部に通知する。

4. プロトタイプシステムの実装

本章では、前章の設計に基づいたツールのプロトタイプの実装について説明する。

4.1 システム環境

実装を行ったシステム環境の概要を図3に示す。データアイテムは、UDPを用いたPC間の通信により、1つのサーバから特定のサブネットマスクを持つLAN内の複数のクライアントに対し、10BASE-Tのイーサネットを介して同時にプッシュ型放送される。データアイテムには、ビットマップ画像ファイルを用いた。サーバ側およびクライアント側ツールは、それぞれWindows PC上にVisual C++を用いて実装した。

サーバは、指定されたスケジューリング方式に基づいてデータアイテムを選択し、放送を行う。また、スケジューリングに必要なシステム全体のアクセス特性は、サーバにおいて既知とした。実際の環境におけるそれらの取得方法としては、クライアントのアクセスログを、別の帯域を用いて定期的に収集し、統計的な処理を行うことなどが考えられる。

クライアントは、データアイテムが放送されるたびに指定されたキャッシング方式に基づいて、キャッシュの置き換えを行う。また、利用者のビットマップ画像に対する要求に従って、アクセスした画像を画面に表示する。さらに、現在までのアク

セス履歴を収集することによって、データアイテムごとのアクセス確率などの統計情報を得る。

4.2 サーバ側ツール

サーバ側ツールは、前章で示した通信部とモニタリング・チューニング部をサーバの画像配信アプリケーションの機能の一部として、アプリケーションに組み込む形式で実装した。

4.2.1 通信部

放送プログラムのスケジューリング方式としては次の三つを用いた。

- Vaidyaらのスケジューリング方式
- 放送ディスク
- 各データアイテムを定期的に等しい頻度で放送するスケジューリング方式

ここで、2.で述べた放送ディスクのあいまいさをなくすために、各データアイテム*i*の放送頻度比 w_i は式(4)のように決定することにした。このようにすることで、アクセス確率が時間に対して変化しない場合、平均応答時間が最短になることが文献[16]において明らかにされている。

$$w_i = \frac{\sqrt{P_i}}{\sum_{j=1}^M \sqrt{P_j}} \quad (4)$$

データアイテムのディスクへの割り振り方は、二重循環法[3]を参考にし、 $def = (w_{max} - w_{min})/N$ とすると、 w_i が $w_{max} - g \cdot def \leq w_i \leq w_{max} - (g-1) \cdot def$ のとき、アイテム*i*はディスク D_g に属することとした。ここで、各データアイテムの放送頻度比の中で、最大の値をとる w_i を w_{max} 、最小の値をとる w_i を w_{min} とする。また、全データアイテムの放送頻度比 w_i を w_{max} で割ることによって得られた値の小数第一位をそのデータアイテムの放送頻度の整数比とした。ここで、 w_{max} の値をとるデータアイテムの整数比は10とした。

サーバ側ツールは、放送が開始されると、指定されたスケジューリング方式に従って100回先までの放送プログラムを決定した後、それに従って各データアイテムをクライアントに放送する。通信プロトコルはUDP/IPを用いたが、下層のIPでは一度に送ることのできるデータグラム最大のサイズが65,535バイトに制限されている。したがって、本プロトタイプでは、一度に60,000バイトまでの送信を許し、データアイテムがこの最大送信バイト数より大きいときは分割して放送する。分割されたデータアイテムには、順にシーケンス番号を付与するものとした。通信部では、クライアントが放送されるデータの詳細を知ることができるように、ヘッダを付与する。ヘッダの構成を表1に示す。

データアイテムを一つ放送するごとに、前回までにスケジューリングしたアイテムの次に放送するアイテムを決定し、放送予定キューのアイテム数を一定に保つようにした。決定した放送予定はデータアイテムとは独立して、各データアイテムの放送終了後にクライアントに放送する。

また、放送中でもスケジューリング方式の変更などのチューニングを可能とした。チューニングや方式の変更の指示が通信部に送られると、現在放送中のデータアイテムの送信が終了次

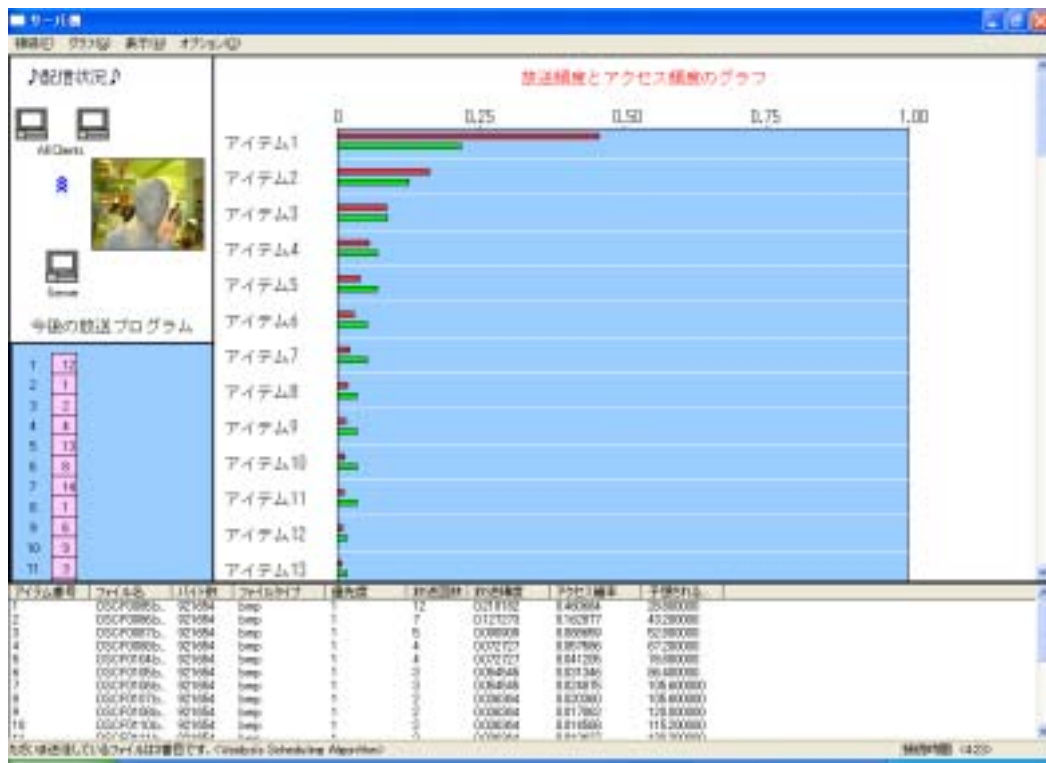


図 4 サーバ側ツールの動作画面

Fig. 4 A screen shot of the server tool.

第, 今までの放送予定を消去し再びスケジューリングを行う. スケジューリングが完了次第, 放送を再開する.

本プロトタイプでは, 次の性能指標の情報を管理する.

- 各データアイテムの放送回数, 放送頻度
- 各データアイテムのシステム全体のアクセス確率
- 予想されるシステム全体の平均応答時間

4.2.2 モニタリング・チューニング部

モニタリング機能としては, 次のものを実現した.

- (1) 放送プログラムの図示
- (2) 現在放送中のアイテムのサムネイル表示
- (3) 現在使われているスケジューリング方式の動作の詳細表示
- (4) 放送を行っていた経過時間の表示
- (5) 4.2.1 に挙げた通信部が管理する性能指標およびデータアイテムに関する詳細情報 (ファイル名, バイト数, ファイルタイプ, 優先度など) のリストビューによる一覧表示
- (6) 性能指標のグラフによる表示

サーバ管理者は図 4 に示すメイン画面を用いてモニタリング

を行う. (1) と (2) はメイン画面の左上のペインに, (3) と (4) は最下部のステータスウィンドウに, (5) は下のペインに, (6) は右上のペインに表示される. 図 4 において, 右上のペインに表示されているのは, アイテムごとの放送頻度とアクセス確率のグラフである.

チューニング機能としては, 次のものを実現した.

- 特定のアイテムの優先度を指定し, 放送頻度を調整する機能. 具体的には, Vaidya らのスケジューリング方式を用いる場合は, 各データアイテムに優先度を評価関数 $G(j)$ に掛け合わせる. 各データアイテムを定期的に等しい頻度で放送するスケジューリング方式では, 優先度を一周期に放送される頻度とした.

- スケジューリングアルゴリズムのパラメータ調整機能. 例として, Vaidya らのスケジューリング方式を用いるときに, スケジューリング時に用いる評価関数 $G(j)$ のパラメータとなるシステム全体のアクセス確率 p_j や, 各データアイテムのサイズ l_j , 各データアイテムが前回放送されてから経過した時間 $(Q - R(j))$ への重み付けを変更できるようにした. 例えば, $G(j) = (Q - R(j))^4 \cdot p_j / l_j$ のように変更することにより, アクセス確率やデータアイテムのサイズよりも, 前回放送されてからの経過時間を重視するように, スケジューリング方式の動作を変更することができる. 放送ディスクでは, 分割するディスク数 N を指定することができる.

4.3 クライアント側ツール

クライアント側ツールは, 前章で示した通信部とモニタリング・チューニング部を, クライアントの画像配信・閲覧アプリ

表 1 ヘッダの要素

Table 1 Elements of the header.

要素	説明
Flag	データアイテム本体か放送プログラムかを識別するフラグ
ID	データアイテムの識別番号
Sequence	分割されて放送されたときのシーケンス番号
Name	データアイテムの名前 (ファイル名)
Size	データアイテムのサイズ (バイト数)

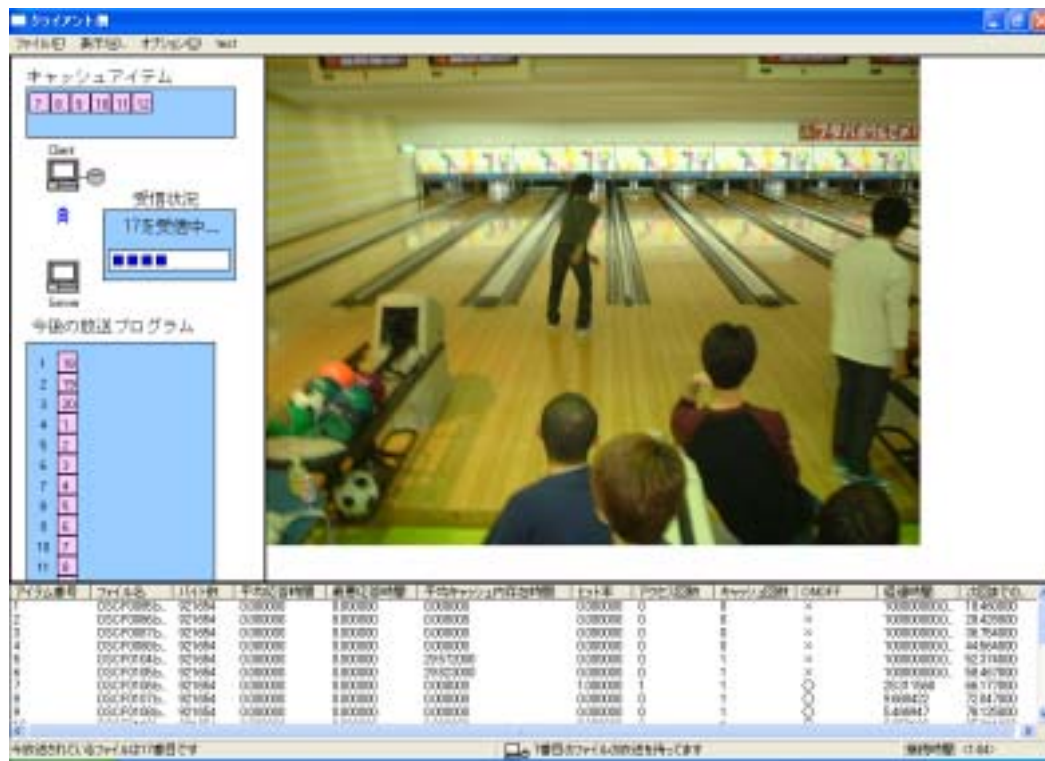


図 5 クライアント側ツールの動作画面
Fig. 5 A screen shot of the client tool.

ケーションの一部として、アプリケーションに組み込む形で実装した。このアプリケーションにおいて、通信部とモニタリング・チューニング部以外の、ビットマップ画像の要求および表示を行う部分を要求・表示部と呼ぶ。

4.3.1 通信部

キャッシング方式としては次の三つを用いた。

- PT 法
- PIX 法
- LRU(Least Recently Used) 法

クライアント側ツールでは、クライアント利用者によりデータの受信開始が指示されると、放送帯域の監視を始める。各データの放送時に、放送データのヘッダの Flag を確認し、それがデータアイテム本体か放送プログラムかを識別する。放送データがデータアイテムのときは、ヘッダの Size に示されているサイズを参照し、受信が正常に行われたことを確認する。受信が正常に行われたときは、指定されたキャッシング方式に従って、キャッシュの置き換えを行う。何らかの理由により受信が正常に行われなかった場合は、そのデータアイテムの受信をあきらめ、次のデータが放送されるのを待つ。送られたデータが放送プログラムのときは、それを保存する。この放送プログラムを参照することにより、クライアント利用者がデータアイテムにアクセス要求したときの応答時間の予測値を表示したり、PT 法のパラメータの値を決定できる。

また、要求・表示部よりデータアイテムへのアクセス要求が発生したことが通知されると、そのデータアイテムがキャッシュに存在するならば、その画像を要求・表示部に送る。キャッシュ

がない場合はそのデータアイテムが次に放送されるのを待ち、受信を完了するとそれを要求・表示部に送る。

さらに、受信中でもキャッシング方式の変更などのチューニングを可能とした。モニタリング・チューニング部からチューニングや方式の変更が通知されると、現在受信中のデータアイテムの受信が終了次第、キャッシング方式の変更やチューニングを行う。

本プロトタイプでは、次の性能指標の情報を管理する。

- 各データアイテムの平均応答時間、最悪応答時間、クライアントのアクセス確率、アクセス回数、ヒット率、キャッシュ内存在時間、キャッシュ回数。

- 全データアイテムの平均応答時間、全データアイテムのヒット率

4.3.2 モニタリング・チューニング部

モニタリング機能としては、次のものを実現した。

- (1) 放送プログラムの図示
- (2) アクセス要求したデータアイテムの応答時間の予測値の表示
- (3) 現在使われているキャッシング方式の動作の詳細表示
- (4) 4.3.1 節に挙げた、通信部で管理する性能指標、およびデータアイテムに関する詳細情報（ファイル名、バイト数、キャッシュの有無など）のリストビューによる一覧表示
- (5) 性能指標のグラフによる表示
- (6) 現在のキャッシュ内容の図示

クライアント利用者は図 5 で示すメイン画面を用いてモニタリングを行う。(1) と (6) はメイン画面の左上のペインに、(2),

(3) は最下部のステータスウィンドウに、(4) は下のペインに、(5) は右上のペインに表示される。なお右上のペインは要求・表示部が画像を表示する場合にも使用される。

チューニング機能としては、次のものを実現した。

- 特定のアイテムの優先度を指定し、キャッシュの頻度を調整する機能。

- キャッシングアルゴリズムのパラメータ調整機能。例として、PT 法を用いるときは、キャッシングの際に必要な PT 値のパラメータとなる、 p_j や $(u_j(Q) - Q)$ への重み付けを変更できるようにした。例えば、 $p_j^4 \cdot (u_j(Q) - Q)$ のように変更することにより、次回の放送までの時間よりも、データアイテムへのアクセス確率を重視するように、キャッシング方式の動作を変更することができる。

5. おわりに

本稿では、プッシュ型情報システムにおいてスケジューリング方式やキャッシング方式のモニタリングやチューニングが可能なサーバツールおよびクライアントツールを提案した。このツールでは、ユーザがスケジューリングおよびキャッシング方式の動作や、性能指標に関する情報を視覚的にモニタリングし、必要に応じて方式の変更やチューニングを行うことができる。これにより、システム環境や個々のユーザの要求に応じた性能を得ることが可能となる。さらに本稿では、設計を行った提案ツールのプロトタイプの実装を行った。

本稿では、サーバがスケジューリングに必要なシステム全体のアクセス特性を得る方法については、議論していない。したがって、そのような機能の実現は今後の課題である。

また、実装したプロトタイプシステムは、通信部、モニタリング・チューニング部を一つのアプリケーションに組み込む形で実現した。今後は、外部アプリケーション、通信部、およびモニタリング・チューニング部を分離することによって、データアイテムの内容に依存せず、必要な場合にのみモニタリング・チューニングを実行することを可能とするように実装を拡張する予定である。

また、既存のプッシュ型放送システムと比較して、提案ツールがユーザの要求や環境に応じて動作していることを確認するための評価を行う必要がある。ただし、ユーザの要求は多分に感覚的なものである上、それらは時間的に変化するため、ツールのモニタリング・チューニング機能に対する評価を正当に行うことは非常に困難であると考えられる。したがって、評価方法についても十分な検討を行う必要がある。

謝辞

本研究は、文部科学省 21 世紀 COE プログラム（研究拠点形成費補助）、文部科学省特定領域研究（14019063）および文部科学省科学技術振興調整費「モバイル環境向 P2P 型情報共有基盤の確立」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] S. Acharya, R. Alonso, M. Franklin, and S. Zdonik, “Broadcast Disks: Data Management for Asymmetric Communica-

tion Environments,” Proc. ACM SIGMOD’95, pp. 199–210, 1995.

- [2] S. Acharya, M. Franklin, and S. Zdonik, “Prefetching from a Broadcast Disk,” Proc. Int’l Conf. on Database Engineering, pp. 276–285, 1996.
- [3] 青野正宏, 田窪昭夫, 渡辺尚, 水野忠則, “データ放送におけるスケジューリング決定法「二重循環法」の提案と評価,” 情報処理学会論文誌, Vol. 40, No. 3, pp. 1267–1275, 1999.
- [4] Ö. Erçetin and L. Tassiulas, “Push-Based Information Delivery in Two Stage Satellite-Terrestrial Wireless Systems,” IEEE Transactions on Computers, Vol. 50, No. 5, pp. 506–518, 2001.
- [5] S. Hameed and N. H. Vaidya, “Efficient Algorithms for Scheduling Data Broadcast,” ACM-Baltzer Journal of Wireless Networks, Vol. 5, No. 3, pp. 183–193 (1999).
- [6] 石川裕治, 田辺雅則, 箱守聡, 井上潮, “HTML 文書間のデータ共有を考慮した放送型情報提供方式,” 情報処理学会論文誌, Vol. 40, No. 7, pp. 3051–3062, 1999.
- [7] カン ギョウビ, 浅田一繁, 飯沢 篤志, 古瀬 一隆, “2 次元的な放送モデルにおける配信間隔と配信スケジューリング,” 情報処理学会論文誌: データベース, Vol. 42, No. SIG 10(TOD 11), pp. 54–63, 2001.
- [8] W. Li, W. Zhang, V. Liberatore, V. Penkrot, J. Beaver, M. A. Sharaf, S. Roychowdhury, P. k. Chrysanthis, and K. Pruhs: “An optimized multicast-based data dissemination middleware: a demonstration,” in Proc. ICDE2003, to appear (demo session) (2003).
- [9] L. Lin and Z. Xingming, “Heuristic MultiDisk Scheduling for Data Broadcasting,” Proc. Int’l Workshop on Satellite-Based Information Services (WOSBIS’97), pp. 1–5, 1997.
- [10] C. J. Su, L. Tassiulas, and V. J. Tsotras, “Broadcast Scheduling for Information Distribution,” ACM-Baltzer Journal of Wireless Networks, Vol. 5, No. 2, pp. 137–147, 1999.
- [11] C. J. Su and L. Tassiulas, “Joint Broadcast Scheduling and User’s Cache Management for Efficient Information Delivery,” ACM-Baltzer Journal of Wireless Networks, Vol. 6, No. 4, pp. 279–288, 2000.
- [12] L. Tassiulas and C. J. Su, “Optimal Memory Management Strategies For a Mobile User in a Broadcast Data Delivery System,” IEEE Journal on Selected Areas in Communications, Vol. 15, No. 7, pp. 1226–1238, 1997.
- [13] 内田涉, 原隆浩, 塚本昌彦, 矢島悦子, 西尾章治郎, “データ間の相関性とアクセス頻度を考慮した放送スケジューリング,” 情報処理学会論文誌: データベース, Vol. 43, No. SIG 2 (TOD 13), pp. 146–157, 2002.
- [14] 内田涉, 原隆浩, 西尾章治郎, “アクセス要求発生頻度の時間的変化を考慮した相関データの放送スケジューリング,” 情報処理学会論文誌: データベース, Vol. 43, No. SIG 9 (TOD 15), pp. 28–38, 2002.
- [15] 内田涉, 原隆浩, 西尾章治郎, “プッシュ型放送のためのデータ利用時間と相関性を考慮したキャッシング方式,” 情報処理学会第 129 回データベースシステム研究報告 (DBS-129/BCC-4), to appear, 2003.
- [16] N. H. Vaidya and S. Hameed, “Scheduling Data Broadcast in Asymmetric Communication Environments,” ACM-Baltzer Journal of Wireless Networks, Vol. 5, No. 3, pp. 171–182, 1999.
- [17] 矢島悦子, 原隆浩, 塚本昌彦, 西尾章治郎, “相関性をもつデータ間の放送時間間隔について,” 情報処理学会論文誌, Vol. 40, No. 1, pp. 188–196, 1999.
- [18] 矢島悦子, 原隆浩, 塚本昌彦, 西尾章治郎, “データ間の相関性を考慮した放送データのスケジューリング法およびキャッシング法,” 情報処理学会論文誌, Vol. 40, No. 9, pp. 3577–3585, 1999.