

プッシュ型情報システムにおけるデータ利用時間を 考慮したキャッシング方式

内田 渉[†] 原 隆浩[†] 西尾章治郎[†]

[†]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: †{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、放送通信技術を用いたプッシュ型の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、クライアントが相関性をもつ複数のデータアイテムに対して、時間間隔において連続的にアクセス要求を発行するようなプッシュ型情報システムを想定し、データアクセスの平均応答時間短縮のためのキャッシング方式を提案する。提案方式では、データ間の相関性およびデータ利用時間を考慮して、次回、および次々回のアクセス要求に対する応答時間の利得が大きいデータアイテムをあらかじめキャッシュする。さらに本稿では、シミュレーション実験によって、提案方式の有効性を検証する。

キーワード プッシュ型情報システム, 相関性, キャッシング方式, データ利用時間

Caching Strategies for Push-based Information Systems Considering Think-time of Data Accesses

Wataru UCHIDA[†], Takahiro HARA[†], and Shojiro NISHIO[†]

[†] Dept. of Multimedia Eng., Graduate School of Information Science and Technology, Osaka University

2-1 Yamadaoka, Suita, Osaka 565-0871, Japan

E-mail: †{wataru,hara,nishio}@ist.osaka-u.ac.jp

Abstract Recently, there has been increasing interest in research of push-based information systems that deliver data by broadcast in both wired and wireless environments. In this paper, to reduce the response time of data access, we propose new caching strategies assuming a push-based information system in which clients consecutively issue access requests for multiple data items with think time. The proposed strategies takes into account each client's access characteristics such as correlations among data items and think-time between a data access and the next access request and caches data items with long expected response times of next and after the next access requests. Moreover, we verify effectiveness of the proposed strategies by simulation experiments.

Key words push-based information system, correlation, caching strategy, think-time

1. はじめに

近年、有線・無線の通信環境の発展に伴い、サーバが放送型通信を用いてクライアントに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に送信せずにサーバの放送帯域を監視し、そのデータ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別にデータを配送する1対1の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化しない。また、放送内容をクライアントからサーバへ送信されたアクセス要求に基づいて決定する1対多のオンデマンド型情報システムと違い、アクセス要求がサーバへ送信されないため、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。

プッシュ型情報システムにおける平均応答時間を短縮する方法として、サーバ側でのスケジューリング戦略[5]~

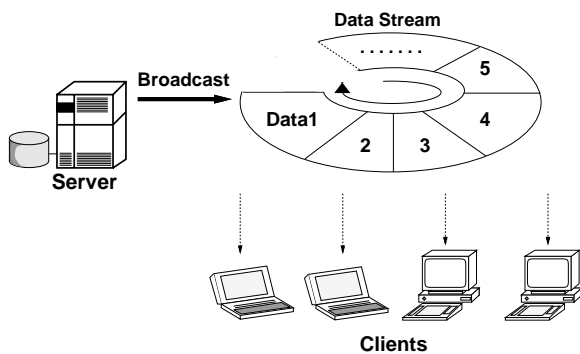


図1 プッシュ型情報システム

Fig. 1 A push-based information system.

[9], [14], [19], クライアント側でのデータアイテムのキャッシュ戦略 [2], [12], [13], プッシュ型とオンデマンド型の融合戦略 [3] などがこれまでに提案されている。

これまでに提案されたサーバ側でのスケジューリング戦略の中で最も代表的なものとして, Acharya らによる放送ディスク [1] や Vaidya ら [17], Su ら [11] が提案した方式がある。これらの方式では, クライアントが頻繁にアクセスするデータを頻繁に放送することによって, 平均応答時間を短縮している。また, キャッシュ戦略の中で代表的なものとしては, Acharya らが, 各クライアントにおけるデータのアクセス確率や放送予定から, 応答時間の利得が大きくなるように効果的なキャッシュの置き換えを行う PT 法を提案している [2]。

一方, クライアントはあるデータの集合にまとめてアクセスすることが多いといったように, データ間には相関性が存在することが一般的である。クライアントが相関性をもつデータ集合に対して頻繁にアクセス要求するような場合には, スケジューリングおよびキャッシングの際にこれらの相関性を考慮することによって, データアクセスの応答時間を短縮することが可能である。

実環境におけるプッシュ型放送の代表的なサービスであるテレビ朝日データの ADAMS [4] では, サーバは複数のカテゴリに分けられた様々な Web ページ形式のデータを地上派テレビ放送の隙間帯域を利用して放送する。このとき, サービスの利用者はあるデータを閲覧した後, そのデータと同じカテゴリに属するデータを連続してアクセスすることが多い。つまり, 同じカテゴリに属するデータは, 相関性を持ち, それらに対するアクセス要求は, ある時間間隔において連続的に発行されることが多い。

筆者らは, このようにアクセス要求が時間間隔において発生する環境におけるスケジューリング方式を, 文献 [15], [18] において提案した。また, 文献 [16] において, 後者の環境におけるクライアント側のキャッシング方式である RIB-PT (Request Interval Based PT) 法を提案した。RIB-PT 法はデータ間の相関性およびアクセス要求の発行間隔を考慮し, 次回に発行されるアクセス要求に対する応答時間の利得が大きくなるようにキャッシュを管理する。しかし, 次回に発行されるアクセス要求に対する応答時間のみを考慮するため, アクセス要求の発行

間隔が短く, 次々回のアクセス要求がすぐに発行されるような場合は, 性能が劣化する。そこで本稿では, クライアントが相関性をもつ複数のデータアイテムに対して, 時間間隔において連続的にアクセス要求を発行するような環境を想定し, 次回および次々に発行されるアクセス要求の応答時間を考慮するように RIB-PT 法を拡張したキャッシング方式を提案する。

本稿では, 次のようなシステム環境を想定する。

- サーバは一つとする。
- サーバで放送されるデータはデータアイテムと呼ばれる単位で放送される。データアイテムは M 種類存在し, 1 から M の識別子を用いて区別する。
- 簡単のため, 各データアイテムのサイズは全て等しいものとし, 1 アイテムの放送にかかる時間を 1 タイムスロットとする。
- クライアントはキャッシュをもつ。キャッシュに保持しているアイテムに対するアクセス要求の応答時間は 0 とし, 保持していないアイテムに対する応答時間は, アクセス要求が発行されてから, そのアイテムの次回の放送時刻までの時間とする。
- クライアントは放送プログラムを知っている。これは, 例えば, サーバが周期的にプログラム情報を放送することによって実現することができる。
- 各クライアントにおけるデータ間の相関性などのアクセス特性は, 各クライアントで異なり, これらは各クライアントにおいて既知とする。
- オンデマンド型配信は行わない。
- データの更新は発生しない。

本稿では, ユーザがクライアントシステムにデータ要求を行い, クライアントシステムがデータアイテムに対するアクセスを試みることを, クライアントがアクセス要求を発行すると表記する。その際, クライアントは発行したアクセス要求をサーバへ送信せず, キャッシュされているデータへのアクセスもしくは放送帯域の監視を開始する。

以下, 2. 章で放送アイテム間の相関性およびアクセス要求の発生に関する本稿の想定について述べる。3. 章でこれまでに提案されたクライアント側のキャッシング方式について述べ, 4. 章で次回および次々に発行されるアクセス要求の応答時間を考慮したキャッシング方式の提案を行う。5. 章で提案方式の性能評価を行い, 最後に 6. 章で本稿のまとめを行う。

2. データアイテム間の相関性とアクセス要求

2.1 アイテム間の相関性

一般に, サーバから放送されている様々なデータアイテムは, あるデータアイテム集合がまとめてアクセスされることが多いといったように, 互いに相関性をもつことが多い。例えば, 1. 章における ADAMS の例のように, サーバが HTML ファイルや画像ファイルなどの Web ページを構成する要素をまとめて一つのデータアイテムとして, 様々な Web ページを, デジタル放送帯域などを利用して多数のクライアントにプッシュ型配信する場合を考える。つまり, ユーザからの Web ページの要求は能動的に発生するが, クライアントはサーバにアクセス要求を

送信せず、そのページが放送されるのを待つ。このとき、クライアントが、ある Web サイトのページを閲覧する場合、まずはトップページにアクセス要求を発行する。トップページを取得すると、クライアントはある時間、そのページを閲覧し、そのページからリンクされる別のページにアクセス要求を発行する。したがって、各ページからリンクされている他のページは、リンク元のページと関連性をもつ。

関連性をもつデータアイテム集合は、同時にアクセス要求される場合と、上記の例のように、時間間隔をにおいてアクセス要求される場合の二通りがある。本稿では、後者のように、クライアントが配信されたデータアイテムを利用する時間が、そのデータアイテムの配信に要する時間に比べて充分大きく、関連性をもつ個々のデータアイテムが時間間隔をにおいてアクセス要求される環境を想定する。以下では、この時間間隔をデータ利用時間と呼ぶ。また、本稿では簡単のため、各クライアントが関連性をもつデータアイテムに対して同時に発行するアクセス要求は一つまでとする。したがって、あるアイテムに対するアクセスが完了してから、そのアイテムと関連性をもつ別の単一のアイテムに対してアクセス要求を発行する。

関連性の強さは、クライアントがあるデータアイテムにアクセスした後の、他のデータアイテムへの連続したアクセス要求の発生確率として定義することができる。また、データ利用時間は、直前のデータアクセスからの経過時間によって定義される確率密度関数にしたがって決定する。

実環境において、各クライアントにおけるアイテム間の関連性や、データ利用時間を示す確率密度関数は、クライアントのアクセス履歴を調べることで決定できる。また、上記の例のように、Web ページがデータアイテムとして放送される場合など、データアイテムの内容によっては、解析的にアイテム間の関連性や確率密度関数を決定できる場合もある。

2.2 連鎖的なアクセス要求

前節の例のように、様々な Web ページが放送されている場合、クライアントはあるページを閲覧した後、そのページと関連性をもつ別のページにアクセスし、さらに別のページにアクセスするといったように、関連性をもつデータアイテムに連鎖的にアクセス要求を発行する。

本論文では、クライアントが、一つ目のデータアイテムに対して、ある確率で無記憶性のアクセス要求を発行した後、それと関連性をもつデータアイテムに、ある確率密度関数に従ったデータ利用時間をにおいて連鎖的にアクセスを複数回行う環境を想定する。その一連のアクセスをプロセスと呼び、各プロセスにおいて最初にアクセスされるアイテムを先頭アイテムと呼ぶ。

3. 従来のキャッシング方式

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、これまでに様々なキャッシング方式が提案されている。これらのうちで、プリフェッチ型キャッシング方式は、フィルタリングの手間以外のコストの増加を伴うことなく、重要なアイテムに対するヒット率を高め、平均応答時間を短縮できることから、プッシュ型情報システムに適したキャッシング方式と考え

られている。代表的なプリフェッチ型キャッシング方式として、文献 [2] において PT 法と呼ばれる方式が提案されている。この方式では、データアイテム間の関連性は考慮せず、すべてのアクセス要求が無記憶に発生する環境を想定している。そこで、筆者らは PT 法を拡張した方式として、クライアントが関連性のあるアイテム集合に対して同時にアクセスを要求することを想定した CB-PT (Correlation-Based PT) 法 [19]、データ利用時間をもってアクセス要求が発行される環境を想定した RIB-PT 法 [16] を提案した。

以下では、PT 法、CB-PT 法、および RIB-PT 法の概要について説明する。

3.1 PT 法

PT 法では、次のような手順でキャッシュの置き換えを行う。

(1) 各アイテムの放送開始時に、キャッシュ内のアイテムおよび放送されるアイテムに PT 値と呼ばれる値を与える。アイテム i に与える PT 値 L_i は次式で表される。

$$L_i = p_i \cdot (u_i(t) - t) \quad (1)$$

ただし、 p_i はアイテム i に対するそのクライアントのアクセス確率、 t は現在時刻、 $u_i(t)$ は時刻 t におけるアイテム i の次回の放送時刻とする。

(2) 放送されるアイテム i の PT 値 L_i が、キャッシュ内で PT 値が最小となるアイテム j の PT 値 L_j より大きい場合、アイテム i と j を置き換える。キャッシュ内で PT 値が最小となる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、あるアイテムに対してアクセス要求を発行したとき、その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を示している。すなわち、PT 法では、各アイテムの放送開始時に、各々の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ようにキャッシュを管理する。ここで、PT 法におけるデータアイテムの置き換えは長期的に見た場合に必ずしも最適とはならない。また、データアイテム間の関連性を考慮した場合、各アイテムのアクセス確率は時間に対して一定とはならないため、正確に応答時間の利得を計算することができない。

3.2 CB-PT 法

CB-PT 法では、次のような手順でキャッシュの置き換えを行う。

(1) 各アイテムの放送開始時に、キャッシュ内および放送中のアイテムに対して、PT 値を関連性を考慮するように拡張した CB-PT 値を与える。アイテム i に与える CB-PT 値 G_i は、次式で表される。

$$G_i = (u_i(t) - t) \sum_{k \in C} p_{ik} + \sum_{k \in Q_i} (u_i(t) - u_k(t)) p_{ik} \quad (2)$$

ここで、 p_{ik} はクライアントがアイテム i と k に一括してアクセス要求を起こす確率とする。また、 C は放送データのうちキャッシュ内にあるアイテムの集合とし、 Q_i は現時点からアイテム i の次回の放送時間までに放送され、かつ、 C に含まれないアイテムの集合とする。

(2) 放送されるアイテム i の CB-PT 値 G_i がキャッシュ内で CB-PT 値が最小となるアイテム j の CB-PT 値 G_j より大きい場合、アイテム i と j を置き換える。キャッシュ内で CB-PT 値が最小となる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。

式 (2) は、アイテム i と放送プログラム内の任意のアイテムの 2 つに同時にアクセス要求を起こしたとき、アイテム i をキャッシュに格納していなかった場合に増加する待ち時間を示す。したがって、式 (2) の値が大きなアイテムをキャッシュに残す置き換えは、相関性をもつアイテムに対して同時にアクセス要求を発行する環境において、その瞬間に生じるアクセス要求の平均応答時間を最短にする。しかし、CB-PT 法における置き換えも、PT 法と同様に、長期的に見た場合に必ずしも最適とはならない。また、相関性をもつアイテムに対するアクセス要求が時間間隔において発生する場合は、正確に応答時間の利得を計算できない。

3.3 RIB-PT 法

現在時刻 τ において、クライアントが最近にアクセス要求を発行したアイテムを i とし、アクセス要求の発行時点でキャッシュ内にアイテム i が存在せず、クライアントが次の i の放送を待っているものとする。このとき、次の i の放送時刻を ζ ($\tau < \zeta$) とすると、アイテム i のアクセス後にアイテム j へのアクセス要求が発行されるといった条件の下で、時刻 t に j に対するアクセス要求が発行される確率密度関数は、 $f_{ij}(t - \zeta)$ となる。

次に、最近のアクセス要求が既にアイテム i の放送もしくはキャッシュアクセスによって時刻 ζ に満たされており ($\zeta \leq \tau$)、 i へのアクセス後、一つのアクセス要求も発生していないものと仮定する。このとき、アイテム i のアクセス後にアイテム j のアクセス要求が発生するといった条件の下で、時刻 t に j に対するアクセス要求が発生する確率密度関数は次式のように表される。

$$\frac{1}{1 - \int_{\zeta}^{\tau} f_{ij}(t - \zeta) dt} f_{ij}(t - \zeta) \quad (3)$$

時刻 τ におけるアイテム j の次の放送時刻を $u_j(\tau)$ とすると、 t ($\tau \leq t \leq u_j(\tau)$) に発生する j へのアクセス要求の応答時間は、 j がキャッシュにない場合、 $u_j(\tau) - t$ となる。したがって、クライアントの最近のアクセス要求がアイテム i に対するもので、時刻 $u_j(\tau)$ にアイテム j が放送される場合、 j をキャッシュに格納しないことで生じる、次のアクセス要求に対する平均応答時間の増加分 (期待値) R_j は次式のように表される。この値を、アイテム j の RIB-PT 値と呼ぶ。

$$R_j = S_j \cdot \int_{\tau}^{u_j(\tau)} f_{ij}(t - \zeta) \cdot (u_j(\tau) - t) dt \quad (4)$$

ただし、 S_j は次式で表される。

$$S_j = \begin{cases} c_{ij} & (\tau < \zeta) \\ \frac{c_{ij}}{1 - \int_{\zeta}^{\tau} f_{ij}(t - \zeta) dt} & (\zeta \leq \tau) \end{cases} \quad (5)$$

ここで、次に j をキャッシュする機会である次の放送時刻以前を考慮するため、積分の範囲は現在時刻 τ から、 j の次の放送時刻 $u_j(\tau)$ までとなる。

RIB-PT 法では、次のような手順でキャッシュの置き換えを行う。

(1) 各アイテムの放送開始時に、キャッシュ内および放送中のアイテムに対して、式 (4) で表される RIB-PT 値を計算する。

(2) 放送されるアイテム i の RIB-PT 値 R_i がキャッシュ内で RIB-PT 値が最小となるアイテム j の RIB-PT 値 R_j より大きい場合、アイテム i と j を置き換える。キャッシュ内で RIB-PT 値が最小となる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。

RIB-PT 法では、最近にアクセス要求を発行したアイテムのアクセス (予定) 時刻に基づいて、そのアイテムと相関性をもつデータアイテムに対して次回に発行されるアクセス要求の応答時間を最短にするようにキャッシュの置き換えを行う。

4. RIB-PT 法の拡張

式 (4) から、データアイテム j をキャッシュしないことによる、次回に j に対して発生するアクセス要求の応答時間の増加分が得られる。ここで、データアイテム a をアクセスした後は b をアクセスすることが多く、 b をアクセスした後は c をアクセスすることが多いクライアントが、既に a をアクセスし終えている場合を考える。RIB-PT 法では、次回にアクセスする確率の高い、すなわち a と相関性の強いデータアイテムのみをキャッシュするため、クライアントが b に対してアクセスを要求した時点では、アイテム c はキャッシュに存在しない確率が高い。ここでアイテム b の利用時間が短い場合は、短時間でさらに次のアクセス要求が発行されるため、 b へのアクセス要求時刻と次のアクセス要求時刻の間に放送されるデータアイテムが少なく、 b と相関性の強いデータアイテム c などをキャッシュしておく機会が少ない。つまり、データの利用時間が短い場合は、現時点から考えて、次々回以降の各データアイテムに対するアクセス要求まで考慮して、キャッシュ内のデータアイテムを管理しなければ、キャッシュのヒット率が低下してしまう。

そこで本稿では、現時点を基準として次々回に発行するアクセス要求に対する応答時間も考慮するように、RIB-PT 法を拡張する。以下では、現時点を基準として、次回に発行するアクセス要求に対する応答時間を第一応答時間、次々回に発行するアクセス要求の応答時間を第二応答時間と呼ぶ。

ここで、直前にアクセス要求を発行したデータアイテムを i 、そのアクセス (予定) 時刻を ζ とし、次回はデータアイテム x ($x \neq i, j$) にアクセスするものと仮定する。このとき、 x のアクセス時刻を r とすると、次々回のアクセス要求が j に対するものであるという条件の下で、時刻 t に j に対するアクセス要求が発生する確率密度関数は、 $f_{xj}(t - r)$ となる。したがって、 j の次の放送時刻 $u_j(\tau)$ 以前に x をアクセスしたクライアントが、 j に対して発行するアクセス要求の応答時間の期待値は、次式で求められる。

$$\int_r^{u_j(\tau)} c_{xj} \cdot f_{xj}(t-r) \cdot (u_j(\tau) - t) dt \quad (6)$$

ここで、 x がキャッシュに存在する場合、 x のアクセス時刻 r は x に対するアクセス要求が発行された時刻と等しい。時刻 τ の時点で、次のアクセス要求がキャッシュ内のアイテム x に対するものである場合、 x のアクセス時刻 r の確率密度関数は次式で与えられる。

$$S_x \cdot f_{ix}(r - \zeta) \quad (7)$$

したがって、次回にキャッシュ内に存在するアイテム x をアクセスし、さらにアイテム j をアクセス要求する場合の応答時間の期待値は、式 (6) と式 (7) を用いて次式で与えられる。

$$E_{jx} = S_x \cdot \int_r^{u_j(\tau)} f_{ix}(r - \zeta) \cdot \int_r^{u_j(\tau)} c_{xj} \cdot f_{xj}(t-r) \cdot (u_j(\tau) - t) dt dr \quad (8)$$

一方、データアイテム x がキャッシュに存在しない場合、 x のアクセス時刻 r は、アクセス要求発生後の最初の x の放送時刻となる。ここで、時刻 $\max(\tau, \zeta)$ と $u_j(\tau)$ の区間における、 x の全ての放送時刻の集合を A とする。ただし、 $\max(\tau, \zeta)$ は、 τ と ζ のうちで大きい方の値を表している。このとき、 $v_x(t)$ を時刻 t 以前の前の x の放送時刻とすると、クライアントが区間内の各放送時刻 r ($r \in A$) において x をアクセスする確率は、次式で与えられる。

$$S_x \cdot \int_{\max(\tau, \zeta, v_x(r))}^r f_{ix}(t - \zeta) dt \quad (9)$$

したがって、データアイテム i を最近にアクセス要求したクライアントが、次にキャッシュに存在しないデータアイテム x をアクセスし、続いて j をアクセス要求する場合、時刻 τ の時点での応答時間の増加分（期待値） N_{jx} は式 (6) と式 (9) を用いて、次式で与えられる。

$$N_{jx} = S_x \cdot \sum_{r \in A} \left\{ \int_{\max(\tau, \zeta, v_x(r))}^r f_{ix}(t - \zeta) dt \cdot \int_r^{u_j(\tau)} c_{xj} \cdot f_{xj}(t-r) \cdot (u_j(\tau) - t) dt \right\} \quad (10)$$

以上より、データアイテム x がアクセス要求される時点でのキャッシュ状態が既知であれば、全てのデータアイテム x に対して E_{jx} もしくは N_{jx} を計算し、総和をとることによって、 j に対する第二応答時間の期待値を求めることができる。ところがキャッシュの状態は現在からデータアイテム x の要求時刻までのキャッシング方式の動作に依存して、各タイムスロット毎に変化するため、 x の要求時点のキャッシュ状態を予測することは困難である。また、たとえキャッシュの状態を予測することができたとしても、 R_j や N_{jx} の積分計算は、あらかじめ不定積分の形で求めておけば定数時間での計算が可能であるが、 E_{jx} を計算するためには、二重積分が必要となり、各タイムスロット毎に実時間で計算を終了することは困難である。そこで本稿では、計算量の削減のために、第二応答時間の期待値を近似的に求める

ヒューリスティックなアプローチをとる。このようなアプローチに基づいて、TR-NC (Two step RIB-PT, Neglect of Cache) 法および TR-NSI (Two step RIB-PT, Neglect of Second Interval) 法と呼ぶ二つの方式を提案する。

4.1 TR-NC 法

TR-NC 法では、第二応答時間の計算の際に、次回アクセスするデータアイテム x は常にキャッシュに存在しないものと仮定する。 E_{jx} の計算に伴う二重積分が省かれるため、計算量を削減することができる。具体的な手順は次の通りである。

TR-NC 法：

(1) 各アイテムの放送開始時に、キャッシュ内および放送中のアイテムに対して、式 (4) で表される RIB-PT 値を計算する。

(2) 上記のアイテムに対して、次回にアクセスするデータアイテムが全てキャッシュに存在しないものと仮定した第二応答時間の期待値を RIB-PT 値に加えたものを、アイテム i の TR-NC 値 W_i として次式のように求める。

$$W_i = R_i + \sum_{x=1}^M N_{ix} \quad (11)$$

(3) 放送されるアイテム i の TR-NC 値 W_i がキャッシュ内で TR-NC 値が最小となるアイテム j の TR-NC 値 W_j より大きい場合、アイテム i と j を置き換える。キャッシュ内で TR-NC 値が最小となる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。

ここで、キャッシュのサイズが小さい場合、データアイテム x はキャッシュに存在する可能性が少ないため、この手法により、ほぼ正確に第一応答時間と第二応答時間の期待値の和を計算することができる。

また、次回アクセスするデータアイテム x がキャッシュに存在する場合、 x に対するアクセスが即座に終了するため、次々回のアクセス要求は x がキャッシュに存在しない場合よりも早く発行される。したがって、 x がキャッシュに存在する場合の第二応答時間の期待値 E_{jx} は、 x がキャッシュに存在しない場合の第二応答時間の期待値 N_{jx} よりも大きくなる。TR-NC 法ではデータアイテム x は常にキャッシュに存在しないものと仮定するため、実際よりもデータアイテム j の第二応答時間を小さく計算する。そのため、次々回のアクセスの応答時間の期待値が大きい、本来必要なデータアイテムをキャッシュしない場合があり得る。しかし、実際よりも第二応答時間を大きく計算して、不適当なデータアイテムをキャッシュし、第一応答時間の大きいデータアイテムをキャッシュから追い出してしまうことがないため、キャッシュのサイズが大きい場合でも、第一応答時間のみを考慮する RIB-PT 法より性能が劣ることは少ない。

4.2 TR-NSI 法

TR-NSI 法では、第二応答時間の計算時に、現在から x の放送までキャッシュの状態が保存されるものと仮定する。さらに、そのままでは E_{jx} の計算が困難なため、 x をアクセスしたクライアントは、即座に j に対してアクセス要求を発行する（データ利用時間が 0 である）と仮定する。

この場合、時刻 r において x をアクセスしたクライアントが、

次に j にアクセス要求を発行する場合の応答時間の期待値は、式 (6) ではなく、 $c_{xj}(u_j(\tau) - r)$ で与えられる。したがって、 x がキャッシュされている場合、およびキャッシュされていない場合の、第二応答時間の期待値 E'_{jx} 、 N'_{jx} は、それぞれ次のように与えられる。

$$E'_{jx} = S_x \cdot \int_{\tau}^{u_j(\tau)} f_{ix}(r - \zeta) \cdot c_{xj} \cdot (u_j(\tau) - r) dr \quad (12)$$

$$N'_{jx} = S_x \cdot \sum_{r \in A} \left\{ \int_{\max(\tau, \zeta, v_x(r))}^r f_{ix}(t - \zeta) dt \cdot c_{xj} \cdot (u_j(\tau) - r) \right\} \quad (13)$$

TR-NSI 法の具体的な手順を次に示す。

TR-NSI 法：

- (1) 各アイテムの放送開始時に、キャッシュ内および放送中のアイテムに対して、式 (4) で表される RIB-PT 値を計算する。
- (2) 現在のキャッシュ状況が保存された上、次々回のアクセス要求がデータ利用時間 0 で発行されると仮定した第二応答時間の期待値を、RIB-PT 値 R_i に加え、アイテム i の TR-NSI 値 V_i として次式のように求める。

$$V_i = R_i + \sum_{x \in C} E'_{ix} + \sum_{x \notin C, 1 \leq x \leq M} N'_{ix} \quad (14)$$

- (3) 放送されるアイテム i の TR-NSI 値 V_i がキャッシュ内で TR-NSI 値が最小となるアイテム j の TR-NSI 値 V_j より大きい場合、アイテム i と j を置き換える。キャッシュ内で TR-NSI 値が最小となる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。

TR-NSI 値の計算では、次々回の j に対するアクセス要求は x のアクセスと同時に発行されるものと仮定しているため、 x のデータ利用時間が短く、さらにキャッシュの状況が現在から x のアクセス要求までに大きく変化しない場合には、第一応答時間と第二応答時間の期待値の和をほぼ正確に計算することができる。

5. 性能評価

本章では、提案方式のシミュレーション実験による性能評価を行う。評価では、1. 章に挙げた ADAMS のように、様々なカテゴリに分けられた Web ページ形式のデータを、プッシュ型配送するシステムを想定した。サーバは、一つの Web ページを 1 データアイテムとしてプッシュ型配送し、クライアントは PDA などの携帯端末を用いて Web ページを取得する。サーバが放送する内容は、主に 5 つのカテゴリからなり、各 Web ページは、同じカテゴリの Web ページにリンクしており、クライアントは同じカテゴリの Web ページを続けて要求する可能性が高いものとする。また、クライアントは低い確率で他カテゴリのページを要求することもあるが、どのページを要求するかを予想するのは困難とする。さらに各カテゴリの規模は等しいが、クライアントが頻繁にアクセスする一つのカテゴリのみ、プロ

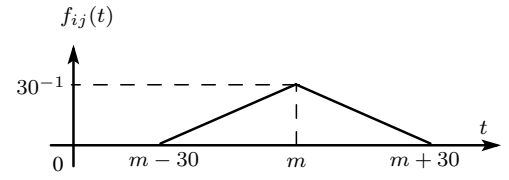


図2 $f_{ij}(t)$.
Fig.2 $f_{ij}(t)$.

セスの先頭においてアクセスされる確率が高いものとする。

実環境では、個々のクライアントのデータアクセスのうち 80% が、放送されている全アイテムの 20% という限られたアイテムに対するものであるという事実が文献 [10] において報告されている。そこで、このような事実に基づいて、想定環境を表現するパラメータ設定を以下に行った。

まずデータアイテム数を 500 とし、全データアイテムをアイテム数 100 の 5 つのグループ (グループ G_1, G_2, \dots, G_5) に分割した。各グループは、上記のカテゴリを示している。 c_{ij} (データアイテム間の相関性の強さ) は、乱数を用いて与え、 $i, j \in G_k, (1 \leq k \leq 5)$ の場合は 10%、 $i \in G_k, j \in G_l, (1 \leq k, l \leq 5, k \neq l)$ の場合は 40% の確率で $c_{ij} > 0$ とした。つまり、各データアイテムは同じグループのデータアイテムとはそれぞれ 10%、別のグループのデータアイテムとはそれぞれ 40% の確率で相関性をもつ。ここで、それぞれの c_{ij} の値は、グループ内、グループ間でそれぞれ等しいものとし、 $\sum_{i,j \in G_k} c_{ij} = 0.8$ 、 $\sum_{i \in G_k, j \in G_l, k \neq l} c_{ij} = 0.2$ とした。このように c_{ij} を与えることによって、クライアントはあるデータアイテムにアクセスを行った後、0.8 の確率で、同じグループの約 10 個のデータアイテムの中の一つに対してアクセス要求を発行し、0.2 の確率で、異なったグループの約 160 個のデータアイテムの中の一つに対してアクセス要求を発行する環境を表している。

さらに、プロセス発生時の先頭アイテムがデータアイテム i である確率 q_i は、グループ 1 に属するデータアイテムのみ高いものとし、他のグループに属するデータアイテムの 16 倍とした。タイムスロット毎のプロセス発生確率は 0.01、あるアクセスを終了した時点でプロセスを終了する確率も 0.01 とした。

各アイテム間 $i-j$ において、要求発生の間隔 (データ利用時間) の確率密度関数を、次式で与えた。

$$f_{ij}(t) = \begin{cases} 30^{-2}(t - m) + 30^{-1} & (m - 30 \leq t < m) \\ -30^{-2}(t - m) + 30^{-1} & (m \leq t \leq m + 30) \\ 0 & (t < m - 30, t > m + 30) \end{cases} \quad (15)$$

これは、 $t-f_{ij}(t)$ 平面上では、図 2 に示すように、 $(m - 30, 0)$ 、 $(m, 30^{-1})$ 、 $(m + 30, 0)$ を結ぶ突起型となる。評価では、 $i-j$ 間のデータ利用時間の平均値を表す m を、 $(\mu - 10, \mu + 10)$ の区間の一様分布で与えた。

サーバは、全てのアイテムを一度ずつランダムに配置したプログラムを、周期的に放送するものとした。評価では、PT 法、CB-PT 法および RIB-PT 法を比較対象とした。PT 法におけるアクセス確率 p_i は、 i が先頭アイテムである確率 q_i で与えた。

また、CB-PT 法における相関性 p_{ij} は、 $p_{ij} = c_{ij}/M$ として与えた。

以上のようなシステム環境において、3,000,000 タイムスロットのシミュレーション実験によって、提案方式の性能評価を行った。以下では、各シミュレーション評価の結果を示し、考察を行う。

5.1 データ利用時間の平均値の影響

クライアント機器として想定する PDA のユーザ記憶領域の容量は、代表的なものでは 11MB 程度である。この領域は他のアプリケーションデータと共有するため、クライアントがキャッシュとして利用することのできる容量は 2MB 程度と考えられる。したがって、一つの Web ページのサイズを 20KB 程度と考えると、クライアントは約 100 個のデータアイテムをキャッシュすることができる。また、Web ページを閲覧する時間は、平均 3 分から 15 分程度を想定する。64K bps の PHS 程度の帯域幅で受信するものとする、データ利用時間の平均値は 72 から 360 タイムスロットとなる。

このような想定に基づいて、キャッシュに格納可能なデータアイテム数（キャッシュサイズ、以下では $|C|$ と表記する）を 100 とし、平均データ利用時間 m の平均値 μ を 50 から 400 まで変化した場合の平均応答時間を評価した。評価結果を図 3 に示す。図 3 より、相関性およびデータ利用時間を考慮してキャッシュの置き換えを行う RIB-PT 法および、TR-NC 法、TR-NSI 法の二つの提案方式が、 μ が大きくなるほど平均応答時間を短縮していることがわかる。これは、平均のデータ利用時間が長い方が、次回に発行されるアクセス要求までに放送されるデータアイテムが多く、効果的なデータアイテムをキャッシュする機会が多くなるためである。

また、平均データ利用時間が短い場合は、第二応答時間を考慮する TR-NC 法および TR-NSI 法の二つの提案方式が、RIB-PT 法よりも良い性能を示している。これは、回目のデータアクセスが行われてから、次々回のアクセス要求が発行されるまでの間隔が短い、第一応答時間しか考慮しない RIB-PT 法ではキャッシュの入れ換えが十分行われなためである。二つの提案方式のうちでは、TR-NSI 法が良い性能を示している。これは回目のアクセス要求の発行時刻までにあまりキャッシュの置き換えが起きない上、実際のデータ利用時間が 0 に近い、TR-NSI 法によってほぼ正確に平均応答時間の期待値を計算できるからである。

一方、平均データ利用時間が非常に長い場合、TR-NC 法および RIB-PT 法が、TR-NSI 法よりも良い性能を示している。データ利用時間が非常に長い場合、回目のデータアクセスが行われた後、次々回のアクセス要求が発行されるまでに、第二平均応答時間の計算対象となるデータアイテムが再び放送される確率が高い。そのため、第二応答時間を考慮する効果は小さく、RIB-PT 法でも十分な性能を示している。TR-NC 法は、計算対象のデータアイテムの回目の放送時刻以降は第二応答時間を計算しないため、RIB-PT 法とほぼ同じ性能を示す。一方、TR-NSI 法は、データ利用時間を 0 と仮定することの誤差により、第二応答時間を実際の値よりも大きく見積もるため、本来

不要なデータアイテムをキャッシュし、第一応答時間が大きく必要なデータアイテムをキャッシュから追い出してしまふ。

5.2 キャッシュサイズの影響

$\mu = 50, 150, 400$ のそれぞれの場合について、キャッシュサイズ $|C|$ を変化させたときの平均応答時間を図 4, 図 5, 図 6 に示す。前節の結果と同様に、RIB-PT 法、TR-NC 法、TR-NSI 法の三方式の性能は、データ利用時間の平均値が大きいくほど良くなっている。データ利用時間が短い図 4, 図 5 においても、第二応答時間まで考慮する TR-NC 法、TR-NSI 法の二つの提案方式が、RIB-PT 法に比べて良い性能を示している。

TR-NC 法と TR-NSI 法を比較すると、 $|C|$ が小さい場合は TR-NSI 法の方が良い性能を示しているが、 $|C|$ が大きい場合は性能が逆転している。TR-NSI 法では、次回にアクセスするデータ x の利用時間が 0 であると仮定するため、実際よりも第二応答時間を大きく計算する。その誤差は、 x がキャッシュに存在しない場合よりも、キャッシュに存在する場合の方が相対的に大きくなる。 $|C|$ が大きいとき、 x はキャッシュに存在する機会が多く、この誤差の影響が大きくなるため、本来不要なデータアイテムをキャッシュしてしまう。

提案した TR-NC 法、TR-NSI 法は、システム特性や要求される計算時間、応答時間に基づいて選択する必要がある。キャッシュサイズが小さい場合は TR-NSI 法、大きい場合は TR-NC 法が有効である。しかし、二つ提案方式は RIB-PT 法と比較して計算量が大きい、タイムスロットが小さく、キャッシュ置き換えのための計算に十分な時間が無い場合は、計算量の小さい RIB-PT 法を用いるべきである。RIB-PT 法は、多くの場合で他の二方式より応答時間が長くなるが、データ利用時間が十分大きい場合は良い性能を示す。

6. おわりに

本稿では、クライアントが相関性のあるデータアイテムのアクセス要求を、データ利用時間において連続的に発行するプッシュ型情報システムを想定して、応答時間短縮のためのキャッシング方式を提案した。提案方式では、データ利用時間およびデータアイテム間の相関性に基づいて、次回および次々回のアクセス要求に対する応答時間の利得を計算し、最も利得が大きいうちにキャッシュを置き換える。

さらに本稿では、シミュレーション実験により、提案方式の性能評価を行った。性能評価の結果、データ利用時間を考慮しない従来方式や第一応答時間のみを考慮する方式と比較して、応答時間を大幅に短縮することを確認した。提案した二方式は、キャッシュサイズに応じて、異なる性能を示すため、実環境ではシステム環境を考慮して適切な方式を選択する必要がある。なお本稿では、特定のアクセス特性を想定して評価を行ったが、今後はデータ利用時間や相関性などについて、さまざまな環境を想定した評価を行う予定である。

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 Communication 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] S. Acharya, M. Franklin, and S. Zdonik, "Balancing Push and Pull for Data Broadcast," Proc. ACM SIGMOD'97, pp. 183–194, 1997 .
- [4] テレビ朝日データ [ADAMS], <URL:http://www.tv-asahidata.com/>.
- [5] 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 .
- [6] 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 .
- [7] 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) .
- [8] 石川裕治, 田辺雅則, 箱守聰, 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 .
- [9] 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 .
- [10] 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 .
- [11] 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 .
- [12] 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 .
- [13] 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 .
- [14] 内田 渉, 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 .
- [15] 内田 渉, 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関連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 .
- [16] 内田 渉, 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), pp. 67–74, 2003 .
- [17] 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 .
- [18] 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 .
- [19] 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 .

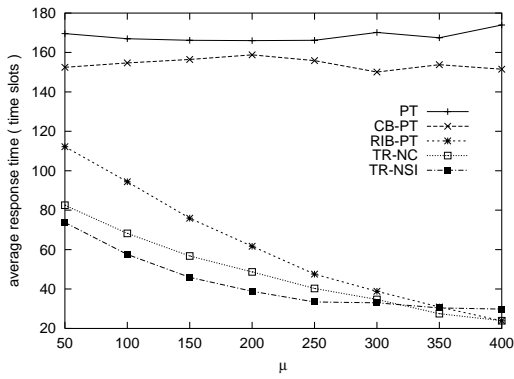


図 3 μ と平均応答時間 ($|C| = 100$) .

Fig. 3 μ vs. average response time ($|C| = 100$).

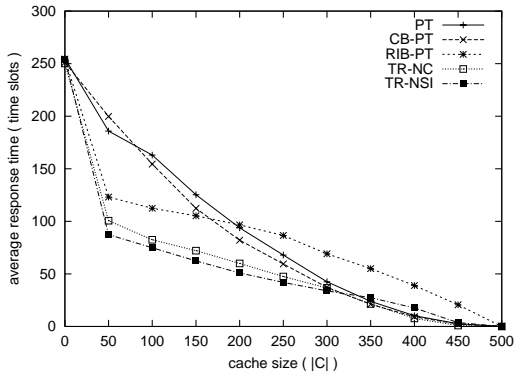


図 4 キャッシュ可能データアイテム数 $|C|$ と平均応答時間 ($\mu = 50$) .

Fig. 4 Cache size $|C|$ vs. average response time ($\mu = 50$).

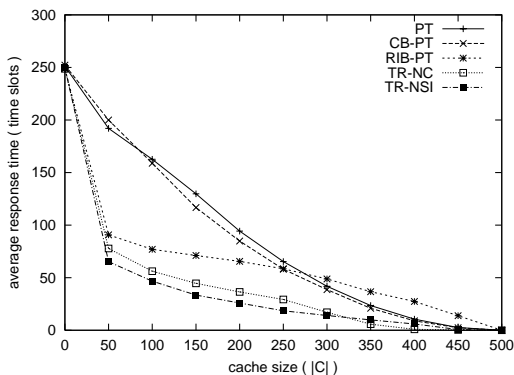


図 5 キャッシュ可能データアイテム数 $|C|$ と平均応答時間 ($\mu = 150$) .

Fig. 5 Cache size $|C|$ vs. average response time ($\mu = 150$).

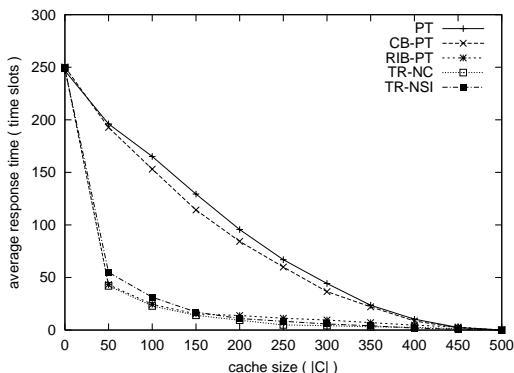


図 6 キャッシュ可能データアイテム数 $|C|$ と平均応答時間 ($\mu = 400$) .

Fig. 6 Cache size $|C|$ vs. average response time ($\mu = 400$).