

プッシュ型放送される関連データの能動的利用のための 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[†]

近年、有線および無線の通信技術の発展にともない、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。本論文では、クライアントが相関性を持つ複数のデータアイテムに対して、時間間隔を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次のアクセス要求を発行するまでのデータ利用時間を考慮して、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。さらに本論文では、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。

Caching Strategies for Active Use of Push-based Broadcast Data Considering Data Correlation and Think-time

WATARU UCHIDA,[†] TAKAHIRO HARA[†] and SHOJIRO NISHIO[†]

Recently, there has been an 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 in push-based information systems, we propose new caching strategies assuming an environment where clients consecutively issue access requests for multiple data items with think-time. The proposed strategies reduce the average response time by caching data items which have long expected response time according to each client's access characteristics such as correlations among data items and think-time between a data access and the next access request. Moreover, we verify effectiveness of the proposed strategies by simulation experiments.

1. はじめに

近年、有線・無線の通信環境の発展にともない、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、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。

一方、サーバの放送帯域をすべてのクライアントが共有するため、多種のデータを放送する必要があり、データアクセスに対するシステム全体の平均応答時間が大きくなる。そこで、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^(9),10)、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),19),20)、プッシュ型とオンデマンド型の融合戦略^(3),8)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⁽²³⁾、Suら⁽¹⁸⁾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

[†] 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
Department of Multimedia Engineering, Graduate
School of Information Science and Technology, Osaka
University

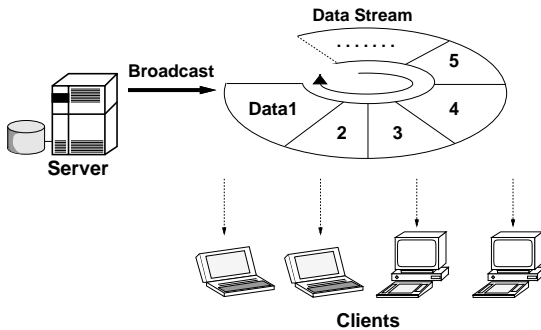


図1 プッシュ型情報システム

Fig.1 Push-based information system.

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方法を提案している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持つデータ集合に対して頻繁にアクセス要求する場合には、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できる。相関性を持つデータ集合に対するアクセス要求は、それらすべてが同時に発行される場合、すべてが時間間隔において発行される場合、両者が組み合わさって発行される場合の3通りが考えられる。

筆者らは、文献 21)、26)において、相関性を持つデータ集合に対して同時にすべてのアクセス要求が発行される環境を想定し、システム全体の応答時間を短縮するためのスケジューリング方式およびキャッシング方式を提案した。一方、すべての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 22)、25)において提案した。しかし、これまでにそのような環境におけるクライアント側のキャッシング方式は提案していない。文献 22)、25)のスケジューリング方式では、システム全体のグローバルなアクセス特性に基づいて放送内容を決定するため、アクセス特性がグローバルなものとは異なるクライアントに対しては、応答時間を短縮できない。そこで本論文では、クライアントが相関性を持つ複数のデータアイテムに対して、すべてのアクセス要求を時間間隔において発行する環境を想定し、個々のクライアントのアクセス特性を考慮したキャッシング方式を提案する。

本論文では、次のようなシステム環境を想定する。

- サーバは1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。
- 各データアイテムのサイズはすべて等しいものとし、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持つ。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、たとえば、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。
- クライアントは放送されているデータアイテムの中から必要なデータアイテムを受動的に受信するが、クライアントの利用者は必要なデータアイテムに対するアクセス要求をクライアントに対して能動的に指示する。つまり、配信形態としてはサーバ主導のプッシュ型放送をとり、クライアントアプリケーションの利用形態としては能動的なものを想定する。
- データの更新は発生しない。

本論文では、クライアント利用者が、クライアントに対してデータアクセスの指示を行い、クライアントがキャッシュへのアクセスもしくは放送帯域の監視を開始することを、クライアントがアクセス要求を発行すると表記する。

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論文の想定について述べる。3章で従来のキャッシング方式について述べ、4章でアイテム間の相関性およびアクセス要求発生の時間間隔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、6章で本研究の関連研究について述べる。最後に7章で本論文のまとめを行う。

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

プッシュ型放送環境では、スケジューリングやキャッシングなどの管理コストを低減するために、各データ

の特性を考慮して、データ集合を1つのデータアイテムとして放送単位とすることが一般的である。ここで、あるデータ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、データ間には相関性が存在することが多い。そのため、データアイテムの構成によって、クライアントのデータアイテムに対するアクセス要求の発行形態が大きく変化する。

たとえば、サーバが静的な 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。ここで、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1つのデータアイテムとする場合、クライアントは、ある Web サイトのページを閲覧する際、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相関性を持ち、それらのデータアイテムに対するアクセス要求は、時間間隔において発行される。

また、HTML ファイルや画像ファイルを個別のデータアイテムとして配送する場合、クライアントは、Web ページを構成する複数のデータアイテムに対して同時にアクセス要求を発行する。クライアントは、1つの Web ページを構成するすべてのデータアイテムの取得を終了すると、そのページとリンクを持つ他の Web ページを構成する複数のデータアイテムに対する要求をデータ利用時間において連続的に発行する。

このように、相関性を持つデータアイテム集合に対するアクセス要求は、データアイテムの構成によって、それらすべてが同時に発行される場合、すべてが時間間隔において発行される場合、両者が組み合わさって発行される場合の3通りが考えられる。本論文では、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、アイテム1つ分の配信時間に比べて十分大きく、相関性を持つ個々のアイテムすべてが、時間間隔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、本論文では簡単のため、各クライアントが相関性を持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1つまでとする。

相関性の強さは、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基づく確率密度関数によって定義される。

実環境において、各クライアントにおけるアイテム間の相関性や、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 Web ページがデータアイテムとして放送される例のように、各ページ間のリンク関係や、コンテンツの内容などで解析的にアイテム間の相関性および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

サーバが様々な Web ページを放送している場合、クライアントはあるページを閲覧した後、そのページと相関性を持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、相関性を持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、1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、それと相関性を持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初め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とまなうことなく、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、プッシュ型通信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 法が提案されている。この方式では、データアイテム間の相関性は考慮せず、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。また、筆者らは文献 26) において、クライアントが相関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て、PT 法を拡張した CB-PT (Correlation-Based PT) 法を提案している。

以下では、PT 法および CB-PT 法の概要について説明する。

PT 法：

- (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 1 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相関性を考慮した場合, 各アイテムのアクセス確率は時間的に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 .

CB-PT 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, PT 値を相関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 1 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, 相関性を持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, 相関性を持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 .

4. アクセス要求発生時の時間間隔を考慮したキャッシング方式

本論文では, クライアントが相関性を持つアイテムに対して, データ利用時間をにおいて逐次的にアクセス要求を発行する環境を想定したキャッシング方式を提案する .

4.1 RIB-PT (Request Interval Based PT) 法

クライアントがアイテム i にアクセスした後にアイテム j にアクセス要求を発行する確率を c_{ij} とする ($\sum_{j=1}^M c_{ij} = 1$). アイテム i のアクセス後にアイテム j にアクセスするといった条件の下で, データ利用時間 t で j へのアクセス要求を発行する確率密度関数を $f_{ij}(t)$ と表す ($\int_0^{\infty} f_{ij}(t) dt = 1, f_{ij}(t) = 0 (t < 0)$).

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, 次回の i の放送時刻を $\zeta (\tau < \zeta)$ 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すでにアイテム 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, 1 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 の次回の放送時刻を $u_j(\tau)$ 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、次回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 は次の式で表されるものとする。

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

ここで、次に j をキャッシュする機会である次回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 の次回の放送時刻 $u_j(\tau)$ までとなる。

同様の式により、キャッシュ内のアイテムの RIB-PT 値を定義できる。この場合は、RIB-PT 値は時刻 τ にそのアイテムをキャッシュから削除することによって生じるアクセスの平均応答時間の増加分である。

提案する 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。

RIB-PT 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) 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 1 つのデータアイテムを選択する。

このように、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持つデータアイテムに対する次回の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。

4.2 RIB-PT 法の拡張

4.2.1 次々に発生するアクセス要求の応答時間
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、すでに 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回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。

そこで本節では、現時点を基準として次々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 1 応答時間、次々に発行するアクセス要求の応答時間を第 2 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、今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 の次回の放送時刻 $u_j(\tau)$ 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 がつねに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．

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

一方、データアイテム x がつねに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 のすべ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、 $v_x(\tau)$ を τ 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．

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

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．

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

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、すべ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 2 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タイムスロットごとに変化するため、 x の要求時点のキャッシュ状態を予測することは非常に困難である．また、 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、各タイムスロットごとに実時間で計算を終了することは困難である．そこで本論文では、計算量の削減のために、第 2 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 2 つの方式を提案する．

4.2.2 TR-NC 法

TR-NC 法では、第 2 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 はつねにキャッシュに存在しないものと仮定する． E_{jx} の計算にともな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とおりである．

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すべてキャッシュに存在しないものと仮定した第 2 応答時間の期待値を、RIB-PT 値に加え、TR-NC 値として次式のように求める．なお、次式はアイテム i の TR-NC 値 W_i を表している．

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

- (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 1 つの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 1 応答時間と第 2 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 2 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 2 応答時間の期待値 N_{jx} よりも大きくなる．TR-NC 法ではアイテム x はつね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 2 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くなり、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うる．しかし、実際よりも第 2 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 1 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ふことがないため、キャッシュのサイズが大きい場合で

も、第1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。

4.2.3 TR-NSI法

TR-NSI法では、第2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2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るうえ、次々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2応答時間の期待値を、RIB-PT値 R_i に加え、TR-NSI値として次式のように求める。なお、次式はアイテム i のTR-NSI値 V_i を表している。

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

- (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1つのデータアイテムを選択する。

TR-NSI値の計算では、次回回の 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1応答時間と第2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。

一方、データ利用時間が長い場合には、アクセス要求が実際よりも早い時刻で発行されることを仮定しているため、実際のものよりも第2応答時間を大きく計算してしまう。したがって、本来不要なデータアイテムをキャッシュしてしまうため、性能が低下する可能性がある。

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、サーバが1つのWebページを1データアイテムとして、様々なWebページを放送する環境を想定する。サーバが放送する内容は、主に5つのサイトからなり、各Webページは、同じサイトのWebページにリンクしており、クライアントは同じサイトの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サイト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、各サイトの規模は等しいが、クライアントが頻繁にアクセスする1つのサイトのみ、プロ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[13]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。

まずアイテム数を500とし、全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。 c_{ij} (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、各アイテムは同じグループのアイテムとはそれぞれ10%、別のグループのアイテムとはそれぞれ40%の確率で相関性を持つ。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、 $\sum_{i, j \in G_k} c_{ij} = 0.8$ 、 $\sum_{i \in G_k \cap j \in G_l \cap k \neq l} c_{ij} = 0.2$ 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個のアイテムの中の1つに対してアクセス要

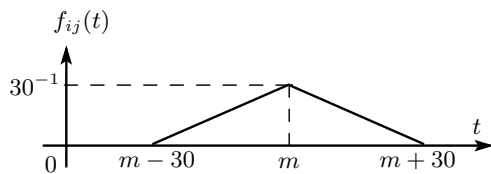


図2 $f_{ij}(t)$
Fig. 2 $f_{ij}(t)$.

求を発行し、0.2の確率で、異なったグループの約160個のアイテムの中の1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アイテム i である確率 q_i は、グループ1に属するアイテムの高いものとし、他のグループに属するアイテムの16倍とした。タイムスロットごと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、要求発生の時間間隔(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のように、 $(m-30, 0)$, $(m, 30^{-1})$, $(m+30, 0)$ 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、すべてのアイテムを1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法およびCB-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、シミュレーション実験の結果を示し、考察を行う。

5.1 データ利用時間の平均値の影響

キャッシュに格納可能なアイテム数(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まで変化させた場合の平均応答時間を図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、TR-NC法、TR-NSI

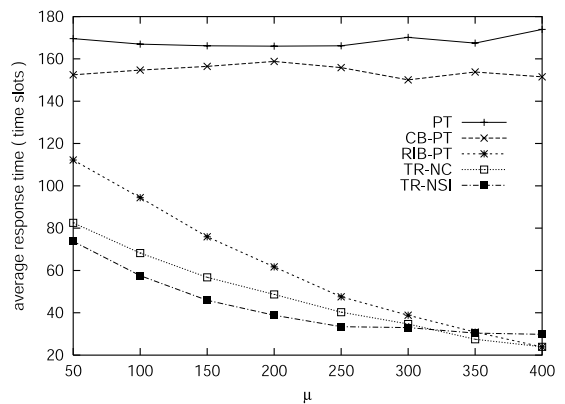


図3 μ と平均応答時間 ($|C| = 100$)
Fig. 3 μ vs. average response time ($|C| = 100$).

法の3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分かる。これは、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アイテム数が多く、効果的な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2応答時間を考慮するTR-NC法およびTR-NSI法の2方式が、RIB-PT法よりも良い性能を示している。これは、次回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ため、第1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かったためである。第2応答時間を考慮する2方式の中では、TR-NSI法が良い性能を示している。これは次回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うえ、実際のデータ利用時間が0に近い場合、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、次回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2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2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次回の放送時刻以降は第2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、第2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1応答時間が大きい本来必要なデータアイテムをキャッシュから追い出

してしまう。

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に示す。前節の結果と同様に、RIB-PT法、TR-NC法、TR-NSI法の3方式の性能は、データ利用時間の平均値が大きいほど良くなっている。データ利用時間が短い図4(a)、(b)においても、第2応答時間まで考慮するTR-NC法、TR-NSI法の2方式が、RIB-PT法に比べて良い性能を示している。

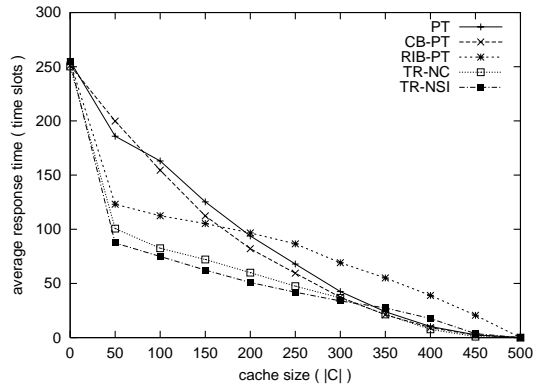
TR-NC法とTR-NSI法を比較すると、 $|C|$ が小さい場合はTR-NSI法が良い性能を示しているが、 $|C|$ が大きい場合は性能が逆転している。TR-NSI法では、次回にアクセスするデータ x の利用時間を0と仮定するため、実際よりも第2応答時間を大きく計算する。その誤差は、 x がキャッシュに存在しない場合よりも、キャッシュに存在する場合の方が相対的に大きくなる。 $|C|$ が大きいとき、 x はキャッシュに存在するケースが多く、この誤差の影響が大きくなるため、本来不要なデータアイテムをキャッシュしてしまう。

提案したRIB-PT法、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。たとえば、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時間が不十分な場合は、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2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。一方、タイムスロットが十分大きく、キャッシュ置き換えのための計算時間が十分にあるときは、データ利用時間が非常に長い場合を除いて、TR-NC法およびTR-NSI法を用いるべきである。このとき、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。

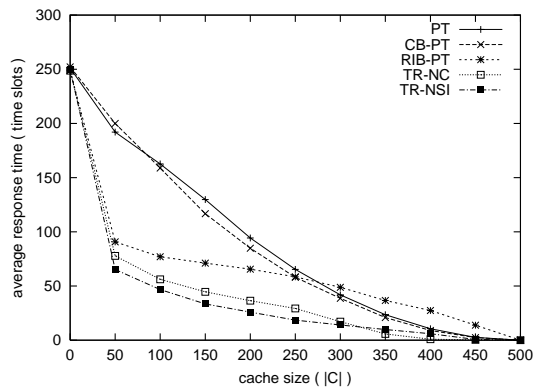
6. 関連研究

本研究で想定するプッシュ型放送を行う環境に限らず、演算装置に関する研究分野やデータベース分野、Webなどのネットワーク上の情報と二次記憶および一次記憶を含めた階層型の記憶装置における研究分野において、プリフェッチに関する研究がこれまでに多く行われている^{(4), (5), (7), (11), (16), (17), (24)}。

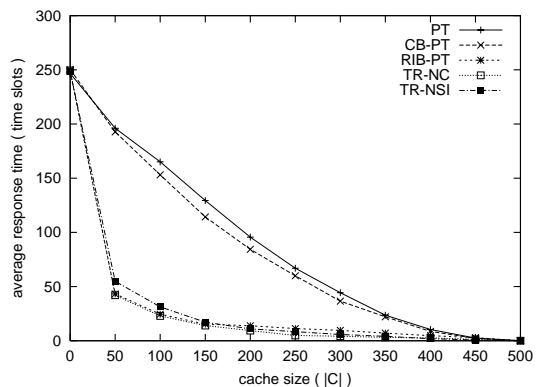
これらの従来研究では、プリフェッチを行う側がアクセス要求を行い、それに応じてデータが配送される1対1のプル型の環境を想定している。プッシュ型放送を行う環境におけるプリフェッチは、プル型環境におけるプリフェッチと比較して、次の特徴を持つ。



(a) $\mu = 50$.



(b) $\mu = 150$.



(c) $\mu = 400$.

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
Fig.4 Cache size $|C|$ vs. average response time.

- 配送形態がサーバ主導のプッシュ型放送をとるため、サーバがあらかじめ放送スケジュールを決定することで、クライアントがデータアイテムの将来の配送時刻を正確に知ることができる。
- 各データアイテムの次回の放送までの時間は、時間的に変化するため、各アイテムをキャッシュすることの利得は動的に変化する。

- アクセス要求をサーバへ送信することがないため、プリフェッチに関するコストはフィルタリングのコストと受信コストのみとなる。
- 各タイムスロットにおいてプリフェッチできるデータアイテムは放送されているデータアイテム1つのみである。
- データアイテムの単位が画像や動画など、放送帯域幅に対してサイズが比較的大きな場合には、キャッシングの動作を決定するための計算に十分な時間があるため、確率密度関数を用いるような複雑な方法で応答時間を計算することも十分実用的となる。

このように、1対1のプル型の配信形態をとる環境とは異なる点が多く、本論文の提案方式はこれらの特徴を考慮しているため、プッシュ型放送を行う環境に特有の方式といえる。以下では、より具体的に、提案方式と従来研究の相違点について議論する。

提案方式は、クライアントのアクセス特性に基づいて、次回および次々回に発行するアクセス要求に対する応答時間の期待値を計算しているが、クライアントの連続的なアクセスを予測してデータのプリフェッチを行う手法は、これまでいくつか提案されている。それらのうちでも、本研究と最も共通点が多いものとして、文献 11) があげられる。文献 11) の提案方式では、マルコフ連鎖モデルを応用することによって、プリフェッチ候補となるデータに対して、一定時間先までに発行されるアクセス要求発生数の期待値を求める。アクセス要求発生数の期待値は、本論文と同様にデータ間の相関性と、データ利用時間を考慮して求めている。文献 11) は、プリフェッチを扱ったその他の多くの論文のように静的なアクセス確率を用いるのではなく、各時刻におけるクライアントのアクセス状況からそれらを求めている点で、本論文と共通点を持つ。

文献 11) では、各データのキャッシュの際に用いられる評価値はあくまでアクセス要求の発生数の期待値であり、それを用いることによってキャッシュのヒット率を向上し、結果として平均応答時間の短縮を図っている。一方、プッシュ型放送では、データの配送がサーバ主導で行われるため、各データがどの時刻に放送されるのかをクライアントが正確に知ることができる。そのため、文献 11) のようにプリフェッチの利得をアクセス要求発生数として計算するのではなく、各時点での各データアイテムに対するアクセス要求発行の確率密度と各アイテムの次回の放送までの時間を用いて、そのアイテムをキャッシュすることによる応答時間の利得を直接計算できる。このように、本研究

では、プリフェッチのための評価値の計算において、プッシュ型放送に特化した方法を用いている点が、文献 11) とは異なる。

共有メモリ型並列計算機において、スレッドやプロセス間のデータの共有の際、バリア同期に先立ってデータの同期が確認されているものと仮定し、メモリアクセスをともなう計算をあらかじめ行うことにより、同期の遅延を削減もしくは隠蔽する、投機的実行 (Speculative Processing) に関する研究^{(6), (12), (14), (15)}がさかんに行われている。これらの研究は、計算の経路を先読みしてメモリ (データ) アクセスを実行するという点で、本研究と共通点を持つ。しかし、実行経路はプログラムによりあらかじめ決定されるため、確率的な計算を行う必要がない点で、本研究とは大きく異なる。さらに、投機的な実行の際のミス・ペナルティが、単純なメモリアccessのオーバーヘッドではなく、並行して実行されるプログラム間の関係によって引き起こされるロールバックのコストとなる点でも、本研究とは異なる。

7. おわりに

本論文では、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、データ利用時間を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、最も利得が大きいようにキャッシュを置き換える。

さらに本論文では、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、データ利用時間が長い場合は、データ利用時間を考慮しない従来方式と比較して、応答時間を大幅に短縮することを確認した。提案した3方式は、キャッシュサイズやデータ利用時間などのシステム特性に応じて、異なる性能を示すため、実環境ではそれらを考慮して最適な方式を選択する必要がある。なお本論文では、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、様々な環境を想定した評価を行う予定である。

また、本論文の想定環境では、データ間の相関性およびデータ利用時間の確率密度関数などのクライアント側のアクセス特性は既知のものであるとしている。しかし実環境では、各クライアントがアクセス特性を独自に調べることが困難な場合がある。そこで今後は、このような場合にも、クライアントが自身のアクセス

特性をある程度の精度で得られるような方法を検討する必要がある。たとえば、データ利用時間を確率密度関数の形で保持するのではなく、過去のアクセスにおけるデータ利用時間の平均値のみを記憶しておくアプローチなどが考えられる。

本論文では、相関性を持つデータアイテム集合に対して、すべてのアクセス要求がデータ利用時間をもって連続的に発行される場合を想定した。しかし、実環境では、相関性を持つデータアイテム集合に対するアクセス要求が同時に発行される場合と、データ利用時間において発行される場合が組み合わさることが一般的である。そこで、今後はそのような環境を想定したキャッシング方式やスケジューリング方式について検討する必要がある。

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（15017262）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。

参 考 文 献

- 1) Acharya, S., Alonso, R., Franklin, M. and Zdonik, S.: Broadcast Disks: Data Management for Asymmetric Communication Environments, *Proc. ACM SIGMOD'95*, pp.199-210 (1995).
- 2) Acharya, S., Franklin, M. and Zdonik, S.: Prefetching from a Broadcast Disk, *Proc. Int'l Conf. on Database Engineering*, pp.276-285 (1996).
- 3) Acharya, S., Franklin, M. and Zdonik, S.: Balancing Push and Pull for Data Broadcast, *Proc. ACM SIGMOD'97*, pp.183-194 (1997).
- 4) Cao, P., Felten, E.W., Karlin, A.R. and Li, K.: A Study of Integrated Prefetching and Caching Strategies, *Proc. 1995 ACM SIGMETRICS*, pp.171-182 (1995).
- 5) Curewitz, K.M., Krishnan, P. and Vitter, J.S.: Practical Prefetching via Data Compression, *Proc. 1993 ACM-SIGMOD Conference on Management of Data*, pp.257-266 (1993).
- 6) Gopal, S., Vijaykumar, T., Smith, J.E. and Sohi, G.S.: Speculative Versioning Cache, *Proc. Int'l Symposium on High-Performance Computer Architecture*, pp.195-205 (1998).
- 7) Griffieon, J. and Appleton, R.: Reducing File System Latency Using a Predictive Approach, *Proc. USENIC Summer Tech. Conf.*, pp.197-207 (1994).
- 8) 箱守 聡, 田辺雅則, 石川裕治, 井上 潮: 放送型通信/オンデマンド型通信を統合した情報提供システム, 情報処理学会研究報告, Vol.34, No.8, pp.55-60 (1997).
- 9) Hameed, S. and Vaidya, N.H.: Efficient Algorithms for Scheduling Data Broadcast, *ACM-Baltzer Journal of Wireless Networks*, Vol.5, No.3, pp.183-193 (1999).
- 10) 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).
- 11) Kraiss, A. and Weikum, G.: Integrated Document Caching and Prefetching in Storage Hierarchies based on Markov-Chain Predictions, *VLDB Journal: The Very Large Data Bases*, Vol.7, No.3, pp.141-162 (1998).
- 12) Lai, A.C. and Falsafi, B.: Memory Sharing Predictor: The Key to a Speculative Coherent DSM, *Proc. 26th Int'l Symposium on Computer Architecture*, pp.172-183 (1999).
- 13) Lin, L. and Xingming, Z.: Heuristic MultiDisk Scheduling for Data Broadcasting, *Proc. Int'l Workshop on Satellite-Based Information Services (WOSBIS'97)*, pp.1-5 (1997).
- 14) Martinez, J.F. and Torrellas, J.: Speculative Synchronization: Applying Thread-Level Speculation to Explicitly Parallel Applications, *Proc. Int'l Conf. on Architectural Support for Programming Languages and Operating Systems*, pp.18-29 (2002).
- 15) Mukherjee, S.S. and Hill, M.D.: Using Prediction to Accelerate Coherence Protocols, *Proc. 25th Int'l Symposium on Computer Architecture*, pp.179-190 (1998).
- 16) Oly, J. and Reed, D.A.: Markov Model Prediction of I/O Requests for Scientific Applications, *Proc. Int'l Conf. on Supercomputing (ICS'02)*, pp.147-155 (2002).
- 17) Palmer, M. and Zdonik, S.B.: Fido: A Cache That Learns to Fetch, *Proc. 17th Int'l Conf. on Very Large Data Bases*, pp.255-264 (1991).
- 18) Su, C.J., Tassiulas, L. and Tsotras, V.J.: Broadcast Scheduling for Information Distribution, *ACM-Baltzer Journal of Wireless Networks*, Vol.5, No.2, pp.137-147 (1999).
- 19) Su, C.J. and Tassiulas, L.: 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).
- 20) Tassiulas, L. and Su, C.J.: 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).

- 21) 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).
- 22) 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).
- 23) Vaidya, N.H. and Hameed, S.: Scheduling Data Broadcast in Asymmetric Communication Environments, *ACM-Baltzer Journal of Wireless Networks*, Vol.5, No.3, pp.171–182 (1999).
- 24) Vellanki, V. and Chervenak, A.L.: A Cost-Benefit Scheme for High Performance Predictive Prefetching, *Proc. 1999 ACM/IEEE conference on Supercomputing (CDROM)* (1999).
- 25) 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).
- 26) 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).

(平成 15 年 1 月 7 日受付)

(平成 15 年 4 月 1 日採録)

(担当編集委員 清水 康)



内田 渉

2001 年大阪大学工学部情報システム工学教室卒業。2002 年同大学院工学研究科博士前期課程修了。同年同大学院情報科学研究科博士後期課程入学, 現在に至る。データベ

ースシステムおよび放送型通信に興味を持つ。日本データベース学会会員。



原 隆浩 (正会員)

1995 年大阪大学工学部情報システム工学科卒業。1997 年同大学院工学研究科博士前期課程修了。同年同大学院工学研究科博士後期課程中退後, 同大学院工学研究科情報システム工学専攻助手。2002 年同大学院情報科学研究科マルチメディア工学専攻助手となり, 現在に至る。工学博士。1996 年本学会山下記念研究賞受賞。2000 年電気通信普及財団テレコムシステム技術賞受賞。データベースシステム, モバイルコンピューティング等の研究に従事。IEEE, 電子情報通信学会, 日本データベース学会の各会員。



西尾章治郎 (正会員)

1975 年京都大学工学部数理工学科卒業。1980 年同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。この間, カナダ・ウォータールー大学, ビクトリア大学客員。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, Data & Knowledge Engineering, Data Mining and Knowledge Discovery, The VLDB Journal 等の論文誌編集委員。本学会フェロー含め, ACM, IEEE 等 9 学会の会員。