

プッシュ型放送における同時アクセスと連続アクセスを考慮した スケジューリング方式の評価

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 大阪府吹田市山田丘 2-1

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

筆者らはこれまでに、相関性をもつ複数のデータアイテムに対する同時アクセス要求が、時間間隔をおいて複数回連続的に発行されるようなプッシュ型放送を行うシステムを想定し、データアクセスの平均応答時間短縮のためのスケジューリング方式を提案している。本稿では、相関性をもつデータアイテムに対する同時アクセス要求のみを考慮する従来方式、および連続アクセスのみを考慮する従来方式との性能比較を行い、提案方式の有効性を検証する。

Performance Evaluation of a Scheduling Strategy Considering Simultaneous and Consecutive Accesses in Push-based Broadcast

Wataru UCHIDA[†]

Takahiro HARA[†]

Shojiro NISHIO[†]

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

We proposed a scheduling strategy to reduce the response time of data access in push-based broadcast assuming an environment where simultaneous access requests for correlated data items are issued consecutively with think-time. In this paper, we verify the effectiveness of the proposed strategy by comparing with two conventional strategies; one of them considers only simultaneous access requests and the other considers only consecutive access requests.

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放送に関する研究の関心が高まっている。プッシュ型放送では、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プル型情報システムとは異なり、プッシュ型放送では、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。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。

プッシュ型放送では、多種のデータを放送する必要があるため、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、平均応答時間を短縮する方法として、サーバ側でのさまざまなスケジューリング戦略が提案されている[1, 2, 3, 4, 5, 6, 7, 10]。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。

一方、実環境では、クライアントはあるデータ集合にまとめてアクセスすることが多いといったよう

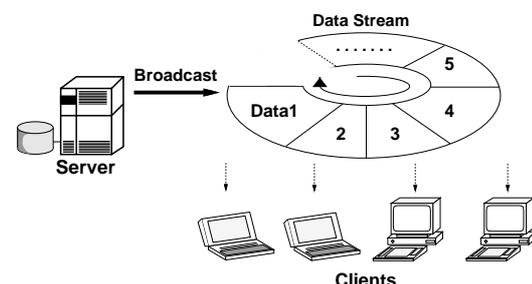


図1: プッシュ型放送を行うシステム

に、データ間に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場合、スケジューリングの際に相関性を考慮することが有効である。

筆者らの研究グループはこれまでに、相関性をもつデータアイテム集合に対するアクセス要求の同時発行を考慮し、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を文献[11]において提案している。また、相関性をもつデータアイテム集合に対するアクセス要求の連続発行を考慮したスケジューリング方式を文献[8]において提案している。

しかし、実環境では、アクセス要求の同時発行と連続発行が組み合わさった形となることが一般的である。そこで筆者らは、文献[9]において、クライアントが相関性をもつデータアイテム集合に対して、アクセス要求を同時に発行し、そのアクセス要求が満たされた後に別のデータアイテム集合に対してア

クセス要求を行う環境を想定して、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を提案した。本稿では、このスケジューリング方式と上記の従来方式を比較評価し、提案方式の有効性を検証する。

本稿では、次のようなシステム環境を想定する。

- サーバは1つとし、プル型配送は行わない。
- 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_d 種類存在し、 d_1, \dots, d_{M_d} 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。
- クライアント側でのキャッシングは考慮しない。この想定は、実環境では必ずしも適当ではないが、提案方式のスケジューリング方式としての性能を正当に評価するためのものである。

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章で提案方式の比較対象となる従来方式について説明し、4章で提案方式について説明する。5章で提案方式の性能評価を行い、6章で本稿のまとめを行う。

2 相関性とアクセス要求

実環境では、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。プッシュ型放送では、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、データ集合を一つのデータアイテムとして放送単位とすることが多い。データアイテムの内容によって、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。

例として、サーバが静的な WWW 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 WWW 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。HTML ファイルや画像ファイルなどの WWW ページを構成する個々の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、WWW 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、一つの WWW 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、そのページを閲覧する時間において、リンク先の WWW ページを構成する複数のデータアイテムに対するアクセス要求を、連続的に発行

する。

このように、互いに相関性をもつデータアイテムは、データアイテムの構成によって、同時にアクセス要求される場合と、順序性をもって、連続的にアクセス要求される場合がある。一般に、クライアントは、相関性をもつ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し、それらのアクセス要求が満たされた後、直前にアクセスしたデータアイテム集合と相関性をもつ別のデータアイテム集合に対してアクセス要求を発行する。したがって、両者が複合した形が一般的である。

そこで本稿では、上記のように相関性をもつデータアイテムに対して、アクセス要求の同時発行と連続発行が複合して行われる環境を想定する。複数のデータアイテムに対して同時に発行されるアクセス要求をクエリと呼び、クエリにおいて発行したデータアイテム集合に対するアクセス要求が、全て満たされることを、クエリが満たされると表現する。また、クライアントがクエリにおいて要求し、受信したデータアイテム集合を利用する時間は、アイテム一つ分の配信時間よりも大きいものとする。以下ではこのデータアイテム集合を利用する時間を、データ利用時間と呼ぶ。

連続して発行されるクエリにおいてアクセスされるアイテム集合間の相関性の強さ（クエリ間の相関性）は、クライアントがあるクエリを発行した後の、他のクエリを連続して発行する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、直前のクエリが満たされた時刻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ものとする。

本稿では、クライアントが、一つ目のデータアイテム集合に対して、ある確率で無記憶性のクエリを発行した後、その集合と相関性をもつデータアイテム集合に、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クエリを複数回発行する環境を想定する。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始めに発行するクエリを先頭クエリと呼ぶ。

実環境において、各クライアントにおけるクエリ間の相関性やクエリにおいてアクセス要求するデータアイテム集合、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 ページの例のように、ページ間のリンク関係やコンテンツの内容などから解析的に決定できる場合もある。

3 従来 of スケジューリング方式

3.1 CBS 法

プッシュ型放送の性能向上のための研究の一環として、これまでに様々なスケジューリング方式が提案されている。筆者らは文献 [11] において、CBS

(Correlation Based Scheduling) 法と呼ぶ方式を提案している．文献 [11] では，クライアントはデータアイテム集合に対して，無記憶性のアクセス要求を同時に発行する環境を想定している．

CBS 法は，放送スケジューリングの問題を巡回セールスマン問題に帰着し，相関性の強いデータアイテムを近くに配置する固定の放送スケジュールをヒューリスティックに作成する．これは，相関性をもつデータアイテム集合に対して同時にアクセス要求が発行される場合，互いに相関性をもつデータアイテムを放送プログラム上で隣接させることにより，最も応答時間を短縮するという事実に基づいている．

CBS 法の概要を次に示す．

CBS 法:

1. 全データアイテムを接点とする完全グラフを作成する．完全グラフの節点 k, l 間の各辺の重みは， $V_{kl} = 1 - s_{kl}$ とする．ここで， s_{kl} は，クライアントがデータアイテム d_k と d_l に対して同時にアクセス要求を発行する確率（データアイテム d_k, d_l 間の相関性）とする．
2. 全ての節点を 1 度ずつ通り，かつ，その道に含まれる辺の重みの総和が最小となる閉路を見つける．
3. 閉路上のアイテムを順に放送プログラム上に配置する．

上記の手順 (2) は，巡回セールスマン問題と等価である．巡回セールスマン問題は NP 完全であるため，アイテム数が非常に多い場合，実時間で解くことができない．しかし，巡回セールスマン問題の近似解を得るための高速なアルゴリズムがこれまでに多く提案されている．したがって，アイテム数が非常に多い場合は，これらを利用して近似的に巡回セールスマン問題を解くものとする．

CBS 法では，放送プログラム内で隣接するアイテム間の相関性のみを考慮しているため，作成される放送順序は必ずしも最適とはならない．しかし，相関性の強いアイテムを隣接させて放送することにより，アクセス要求の同時発行に対する応答時間の短縮が期待できる．

また，CBS 法は，アクセス要求の同時発行のみを考慮し，集合単位のアクセス要求を発行する確率は時間に対して一定であると想定している．本稿の想定環境では，クエリが満たされたクライアントは，相関性に従って，他のデータアイテム集合に対してクエリを連続発行するため，各時点でクエリの発行確率が変化する．

3.2 MTW 方式

筆者らは文献 [8] において，相関性をもつデータアイテム集合に対するアクセス要求の全てが，時間間隔において，連続的に発行されるような環境を想

定して，MTW (Maximum Total Waiting time) 方式を提案している．文献 [8] では，同時にアクセス要求するデータアイテムは，一つまでと想定し，連続して発行される一連のアクセス要求をプロセスと呼ぶ．この想定環境は，本稿において，一つのクエリでアクセス要求するデータアイテム集合をまとめて，一つのデータアイテムとした場合に相当する．

以下では，1 章の想定環境に基づいて，一つのデータアイテムを放送するために必要な時間を 1 タイムスロットとし，タイムスロット n ($n = 0, 1, \dots$) は時区間 $[n, n + 1)$ を表すものとする．

データアイテム d_i をアクセス要求したあと， d_j を連続してアクセスする確率を c_{ij} ($\sum_{i=1}^{M_d} c_{ij} = 1, c_{ii} = 0$) とし，その場合の， d_i へのアクセスが終了した後， d_j に対するアクセスが発行されるまでの時間間隔（データ利用時間） t は確率密度関数 $f_{ij}(t)$ ($\int_0^\infty f_{ij}(t) dt = 1, f_{ij}(t) = 0 (t < 0)$) で表されるものとする．また，単位時間当りのプロセス発生頻度を λ ，プロセス発生の際にデータアイテム d_i が始めにアクセス要求される確率を ρ_i ，あるデータアクセスが終了した際にプロセスを終了する確率を ψ とする．

MTW 方式は，各放送時において，それまでの放送履歴およびこれらのクライアントのアクセス特性を用いて，タイムスロット n におけるデータアイテム $B(n)$ の放送によって満たされたアクセス要求数の期待値 $\mu(n)$ ，タイムスロット n における各データアイテム d_i のアクセス要求発生数の期待値 $p_i(n)$ ，およびそれをアクセス要求するクライアントの待ち時間の総和の期待値 $w_i(n)$ を求め， $w_i(n)$ が最も大きいデータアイテムを放送する．

MTW 方式の概要を次に示す．

MTW 方式:

1. タイムスロット n を 0 に初期化する．また，各データアイテム d_i に対して， $w_i(0) = p_i(0)$ に初期化する．
2. 全データアイテムのうちで， $w_i(n)$ が最大となるアイテムを探し，タイムスロット n における放送データアイテム $B(n)$ とする．そのような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アイテムを選択する．
3. $\mu(n) = p_{B(n)}(n)$ とし， $p_{B(n)}(n)$ ， $w_{B(n)}(n)$ を 0 にリセットする．
4. 次のタイムスロット $n + 1$ の先頭時点におけるアクセス要求の発生数 $p_i(n + 1)$ を，次式のように計算する．

$$p_i(n + 1) = p_i(n) + \lambda \cdot \rho_i + \sum_{m=0}^n \{ \mu(m) \cdot (1 - \psi) \cdot c_{B(m)i} \cdot f_{B(m)i}(n - m) \} \quad (1)$$

5. $w_i(n+1) = w_i(n) + p_i(n+1)$, $n = n+1$ とし, 手順 2 に戻る.

MTW 方式は, クライアントの待ち時間の総和が最も大きいデータアイテムを各タイムスロットで放送する. したがって, 各アイテムの放送時には最適なアイテムを選択することができる. ただし, この選択は, 長期的には最適な選択とは限らない.

また, MTW 方式は, 全てのアクセス要求が連続的に発行され, 同時にアクセス要求されるデータアイテムは一つまでと想定している. したがって, 本稿における想定環境のように, 複数のデータアイテムに対するクエリが, 時間間隔をおいて連続的に発行される環境では, クエリ単位のアクセス要求を考慮することができず, 正確に待ち時間の期待値を計算することができない.

4 同時アクセスと連続アクセスを考慮したスケジューリング方式

4.1 準備

筆者らが文献 [9] において提案した MTW-Q (Maximum Total Waiting time considering Queries) 方式は, 文献 [8] の MTW 方式を, クエリ単位のデータアクセスを考慮するように拡張したものである. MTW-Q 方式は, これまでの放送スケジュールおよびクエリとしてアクセス要求されるデータアイテム集合, データ間の相関性などのクライアントのアクセス特性を用いて各タイムスロットにおけるクエリの発行数の期待値を計算する. それを用いて各データアイテムの待ち時間の総和を計算し, その値が最も大きいアイテムを放送することでヒューリスティックに平均応答時間の短縮を図る.

以下では, クライアントがクエリとして同時にアクセス要求を発行するデータアイテム集合は M_q 種類存在するものとし (Q_1, Q_2, \dots, Q_{M_q}), それぞれのクエリを q_1 から q_{M_q} までの識別子を用いて区別する. また, クライアントが, あるクエリ q_i が満たされたあと, 次にクエリ q_j を発行する確率 (クエリ q_i - q_j 間の相関性) を C_{ij} ($\sum_{j=1}^{M_q} C_{ij} = 1, C_{ii} = 0$) とし, その場合のデータ利用時間 t は, 確率密度関数 $F_{ij}(t)$ ($\int_0^\infty F_{ij}(t)dt = 1, F_{ij}(t) = 0 (t < 0)$) で決定されるものと想定する.

単位時間あたりのプロセス発生確率を Λ , クエリ q_i がプロセスの先頭において発行される確率を θ_i とする. この時, 各時点においてクエリ q_i を先頭とするプロセスが発行される確率 Λ_i は, $\Lambda \cdot \theta_i$ となる. また, あるクエリが満たされたあと, プロセスが終了する確率を Ψ とする.

クライアントがあるクエリ q_i を発行した場合, Q_i に含まれるデータアイテムが全て放送された時点でクエリ q_i が満たされる. 以下では, クエリの応答

時間は, クエリが発行されてから, 満たされるまでに必要な時間とする.

各時刻におけるデータアイテム d_j をアクセス要求するクライアントの待ち時間の総和は, d_j を含む全てのデータアイテム集合 S に対して, それをアクセス要求するクエリの待ち時間 $W_{i,S}(n)$ の総和をとったものとなる. しかし, $W_{i,S}(n)$ を計算するためには, 全てのクエリ q_i に対し, Q_i の各部分集合 S について, それを放送待ちデータアイテム集合とするクエリの発行数および待ち時間の総和 $W_{i,S}(n)$ を記憶し, 毎タイムスロットにおいて, 全ての値を更新する必要があるため, 多くの計算時間および記憶領域を必要とする. そこで, MTW-Q 方式では, タイムスロット n におけるクエリ q_i の発行数 $\Pi_i(n)$, クエリ q_i が満たされた数 $\sigma_i(n)$, データアイテム d_j の前回の放送タイムスロット v_j , タイムスロット n において新規に発行されたクエリ q_i の数 $\gamma_i(n)$ を管理し, 簡略化された計算方法によって, 各データアイテムをアクセス要求するクエリの待ち時間の総和の期待値を近似的に求める.

4.2 データアイテムをアクセス要求するクエリの待ち時間

各タイムスロットの先頭時刻 n において, d_j をアクセス要求するクエリの発行数の期待値を $\pi_j(n)$ とする. MTW-Q 方式では, d_j をアクセス要求するクエリの待ち時間の総和 $\omega_j(n)$ を, 各タイムスロットにおける $\pi_j(n)$ を足し合わせたものとして近似的に計算する. 新規のクエリが発行されるため, d_j をアクセス要求するクエリ数はタイムスロット n 内 (時区間 $[n, n+1)$) で変化するが, 簡単化のため, それらは時刻 n における値から変化しないものと仮定する.

時刻 n において発行されているクエリ q_i のうち, d_j に対するアクセス要求が満たされているのは, d_j の前回の放送タイムスロットの先頭時刻 v_j の時点で発行されていたクエリのみとなる. d_j をアクセス要求しているのは, それ以外の, 時刻 v_j より後に発行されたクエリである. そのため, $\pi_j(n)$ は, 次式で与えられる.

$$\pi_j(n) = \sum_{\{i|d_j \in Q_i\}} \sum_{m=v_j}^{n-1} \gamma_i(m) \quad (2)$$

$\gamma_i(n)$ は, 次式で与えられる.

$$\gamma_i(n) = \sum_{m=0}^n \sum_{j=1}^{M_q} \int_n^{m+1} \sigma_j(m) \cdot (1 - \Psi) \cdot C_{ji} \cdot F_{ji}(u - m) du + \Lambda_i \quad (3)$$

4.3 クエリの発生数

タイムスロット $n+1$ の先頭時刻におけるクエリ q_i の発生数 $\Pi_i(n+1)$ は, タイムスロット n におけ

る発生数 $\Pi_i(n)$ と $\sigma_i(n)$, タイムスロット n 内で新規に発行されたクエリ q_i の数 $\gamma_i(n)$ を用いて, 次式で与えられる .

$$\Pi_i(n+1) = \Pi_i(n) - \sigma_i(n) + \gamma_i(n) \quad (4)$$

$\sigma_i(n)$ は, $v_j, \sigma_i(m), \Pi_i(m)$ ($m = 0, 1, \dots, n-1$) を用いて次のように計算することができる .

$B(n)$ が Q_i に含まれる場合, その放送によって, 発行されているクエリ q_i の一部が満たされる . $B(n)$ 以外の Q_i に含まれるデータアイテムのうち, 最も v_j が小さい, すなわち最後に放送されてから最も時間が経っているデータアイテムを d_l とする . v_l の時点で発行されており, 時刻 n においてまだ満たされていないクエリ q_i の数は $\Pi_i(v_l) - \sum_{m=v_l}^{n-1} \sigma_i(m)$ となる . それらのクエリは, タイムスロット n における $B(n)$ の放送によって, Q_i に含まれる全てのデータアイテムに対するアクセス要求が満たされる . つまり, これらのクエリは, $B(n)$ の放送によって満たされることとなり, $\sigma_i(n)$ は次式で表すことができる .

$$\sigma_i(n) = \Pi_i(v_l) - \sum_{m=v_l}^{n-1} \sigma_i(m) \quad (5)$$

4.4 MTW-Q 方式

MTW-Q 方式のアルゴリズムを次に示す .

1. 全てのクエリ i , データアイテム d_j に対して, $\Pi_i(0) = 0, \omega_j(0) = 0$ とし, $n = 0$ とする .
2. 全てのデータアイテム d_j のうち, 最も $\omega_j(n)$ が大きくなるようなデータアイテムを調べ, 放送データアイテム $B(n)$ とする . そのようなデータアイテムが複数存在する場合は, 無作為に一つを選択する .
3. $B(n)$ を Q_i に含むクエリ q_i に対して, 次の操作を行う .
 Q_i に含まれる $B(n)$ 以外のデータアイテムのうち, 一つでもまだ放送されていないデータアイテムが存在する場合は, $\sigma_i(n) = 0$ とする . それ以外の場合は, Q_i に含まれる $B(n)$ 以外のデータアイテム d_j のうち, $v_j(n)$ が最も小さい d_j を探して d_l とし, $\sigma_i(n)$ を式 (5) を用いて求める .
4. $B(n)$ を Q_i に含まないクエリ q_i に対しては, $\sigma_i(n) = 0$ とする .
5. 全ての q_i に対して, $\Pi_i(n+1)$ を式 (4) を用いて求める .
6. サーバの放送開始以降, まだ放送されたことのないデータアイテム d_j に対しては, $\omega_j(n+1)$ を, 次式を用いて求める .

$$\omega_j(n+1) = \omega_j(n) + \sum_{\{i|d_j \in Q_i\}} \Pi_i(n) \quad (6)$$

既に一度以上放送されているデータアイテムのうち, $d_j \neq B(n)$ となる全ての d_j に対しては, $\omega_j(n+1)$ を次式を用いて求める .

$$\omega_j(n+1) = \omega_j(n) + \pi_j(n) \quad (7)$$

$\omega_{B(n)}(n+1) = 0$ とする .

7. $n = n+1, v_{B(n)} = n$ とし, ステップ 2 へ戻る .

このように MTW-Q 方式では, アクセス要求を発行するクエリの待ち時間の総和が最も大きいデータアイテムを放送する . ただし, あるデータアイテム d_j の放送は, d_j を含むクエリ q_i を必ずしも満たすものではないため, データアイテム d_j の放送がシステム内に存在するクエリ q_i の応答時間を直接短縮するとは限らない . したがって, この選択は, 長期的には最適とは限らない .

5 性能評価

5.1 評価環境

本章では, 提案方式の性能評価を行う . 評価では, サーバが, Web ページを構成する, それぞれの HTML ファイルや画像ファイルを 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した . さらに, 各 Web ページは複数のファイルからなり, 同じサイトに属するページは一部のファイルを共有するものとした . 各サイトの規模は等しいが, ある一つのサイトのみ, クライアントが頻繁にアクセスするものとした . 具体的には, 想定環境を表現するパラメータ設定を以下のように行った .

クエリ数 M_q を 100 とし, 全クエリをクエリ数 20 の 5 つのグループ (グループ H_1, H_2, \dots, H_5) に分割し, クエリ q_i は $H_{\lceil i/20 \rceil}$ に属するものとした . グループ H_1 に属するクエリに対して, C_{ij} を次のようにして与えた . $q_i, q_j \in H_1$ の場合は 80%, $q_i \in H_1 \cap q_j \in H_k, (2 \leq k \leq 5)$ の場合は 20% の確率で $C_{ij} > 0$ とした . つまり, グループ H_1 のクエリは同じグループ H_1 のクエリとそれぞれ 80%, 他のグループのクエリとそれぞれ 20% の確率で相関性をもつ . また, グループ H_1 内のクエリ q_i に対して, 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, $\sum_{\{j|q_j \in H_k\}} C_{ij} = 0.8, \sum_{\{j|q_j \notin H_k\}} C_{ij} = 0.2$ とした .

グループ H_2 から H_5 に属するクエリについては, 次のように C_{ij} を与えた . $q_i, q_j \in H_k, (2 \leq k \leq 5)$ もしくは $q_i \in H_k \cap q_j \in H_1, (2 \leq k \leq 5)$ の場合は 80%, $q_i \in H_k \cap q_j \notin H_k \cap q_j \notin H_1$ の場合は 20% の確率で $C_{ij} > 0$ とした . つまり, グ

ループ H_2 から H_5 の各クエリは同じグループのクエリ, およびグループ H_1 のクエリとはそれぞれ 80%, その他のグループのクエリとはそれぞれ 20% の確率で相関性をもつ. また, グループ H_2 から H_5 のクエリに対して, C_{ij} の値は, 同じグループとグループ H_1 のクエリ, その他のグループのクエリに対してそれぞれ等しいものとし, 全ての i に対して, q_i が属するグループを H_k , ($2 \leq k \leq 5$) とすると, $\sum_{\{j|q_j \in H_k \cup q_j \in H_1\}} C_{ij} = 0.8$, $\sum_{\{j|q_j \notin H_k \cap q_j \notin H_1\}} C_{ij} = 0.2$ とした.

このようにクエリ間の相関性を決定することで, クライアントがグループ 1 のクエリを発行した場合は続けて同じグループ 1 のクエリを発行しやすく, 他のグループのクエリを発行した場合は, 同じグループ, もしくはグループ 1 のクエリを発行しやすい環境となる.

データアイテム数 M_d は 500 とし, これらもデータアイテム数 100 の 5 つのグループ (グループ h_1, h_2, \dots, h_5) に分割し, アイテム d_j はグループ $h_{\lfloor j/100 \rfloor}$ に属するものとした. グループ H_k , ($1 \leq k \leq 5$) に属するクエリは, グループ h_k に属するデータアイテムのうち, I 個のデータアイテムをアクセス要求するものとし, クエリ i でアクセス要求するアイテム集合 Q_i は, 次のように決定した.

まず, 次の式に従って, データアイテム d_j の配置数 a_j を仮に決定した. ここで, h_k はデータアイテム d_j が属するグループとする.

$$a_j = \left\lceil \frac{20 \cdot I \cdot (j \bmod 100)^{-\alpha}}{\sum_{\{l|l \in h_k\}} (l \bmod 100)^{-\alpha}} \right\rceil \quad (8)$$

グループごとのデータアイテム配置数の合計を $T_k = \sum_{\{j|d_j \in h_k\}} a_j$ とする. もし, $T_k > I \cdot 20$ となるグループ h_k が存在する場合, $T_k = I \cdot 20$ となるまでグループ内で j の値が小さいデータアイテム $d_{100 \cdot (k-1)}$ から順に配置数 a_j を 1 つずつ減らし, 各データアイテムの配置数を変更した. ただし, $a_j - 1 = 0$ となる場合は, 再び $d_{100 \cdot (k-1)}$ から順に配置数を減少するものとした.

最後に, グループ h_k のデータアイテム q_j が, H_k のクエリのうち, 無作為に a_j 個のクエリにアクセス要求されるように Q_i を決定した. ここで, α の値が大きいくほど, より少数のデータアイテムがより多くのクエリでアクセス要求されることを示す.

さらに, プロセス発生時の先頭クエリがクエリ q_i である確率 Λ_i は, グループ H_1 に属するクエリのみ高いものとし, 他のグループに属するクエリの 16 倍とした. タイムスロット毎のプロセス発生確率は 0.05, あるクエリが満たされた時点でプロセスを終了する確率も 0.05 とした.

各クエリ $q_i - q_j$ 間において, 要求発生の間隔 (データ利用時間) の確率密度関数 $F_{ij}(t)$ を, 図 2

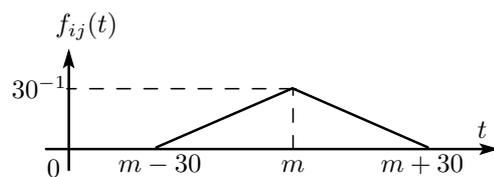


図 2: $F_{ij}(t)$.

表 1: 平均応答時間と平均第一応答時間 ($I = 5$, $\alpha = 0$).

	MTW-Q	MTW	CBS
平均応答時間	262.10	192.70	307.41
平均第一応答時間	116.20	185.00	295.43

のように, $(m-30, 0)$, $(m, 30^{-1})$, $(m+30, 0)$ を結ぶ突起型として与えた. m は, $q_i, q_j \in H_1$ の場合は $(30, 50)$, それ以外は $(390, 410)$ の区間の一様分布で与えた.

比較対象としては, CBS 法, および MTW 方式を用いた. CBS 法の評価の際は, s_{ij} は $d_i, d_j \in Q_k$ となるクエリ Q_k の数と Λ_i の積に比例するように与えた. MTW 方式の評価の際は, クエリでアクセス要求するデータアイテム集合 Q_i を放送の単位 (MTW 方式におけるデータアイテム) とし, 集合 Q_i に含まれる全てのデータアイテムを放送した後, 次に放送するアイテム集合を MTW 方式で決定するものとした. つまり, MTW 方式はあるデータアイテムが複数のクエリにアクセス要求される場合, それらを異なるデータとして扱い, アクセス要求発生数を独立して計算することとなる.

5.2 評価結果

以上のような環境において, 各クエリでアクセス要求するデータアイテム数 I , および α の値を変化させた場合のタイムスロット 0 から 1,000,000 までの平均応答時間, および Q_i 中の一つ目のデータアイテムを取得するまでの平均応答時間 (以下では平均第一応答時間と表記する) を計算した. 平均応答時間は, 文献 [9] に示す方法で計算した.

$I = 5$, $\alpha = 0$ とした場合の平均応答時間, および平均第一応答時間を表 1 に示す. このパラメータ設定は, 各データアイテムが 1 つのクエリのみにおいてアクセス要求され, 複数のクエリにアクセス要求されるデータアイテムが存在しない状況を示す. 表 1 において, MTW 方式が最も平均応答時間を短縮している. これは各クエリでアクセス要求するデータアイテム集合に重なりがなく, 各クエリでアクセス要求するデータアイテム集合をまとめて一つのデータアイテムとして考えた場合, 一度にアクセス要求されるデータアイテムは一つとする MTW 方

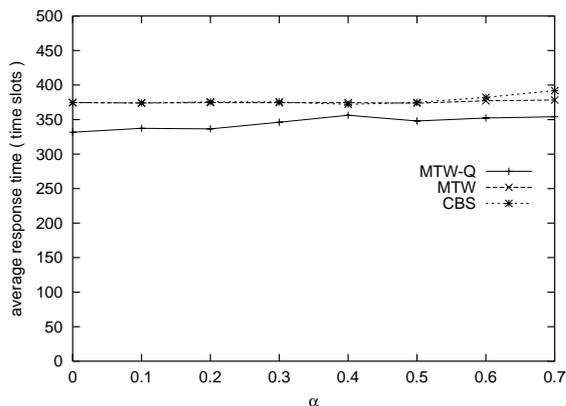


図 3: α と平均応答時間 ($I = 10$) .

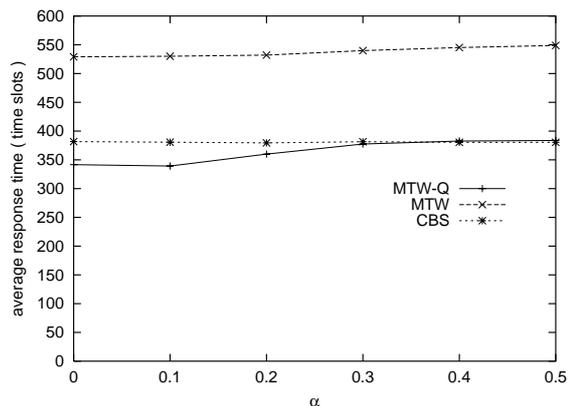


図 5: α と平均応答時間 ($I = 15$) .

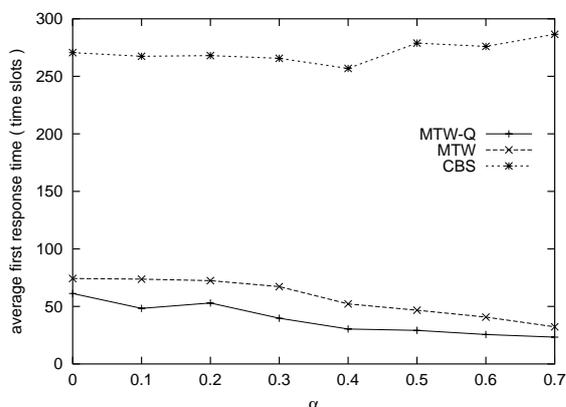


図 4: α と平均第一応答時間 ($I = 10$) .

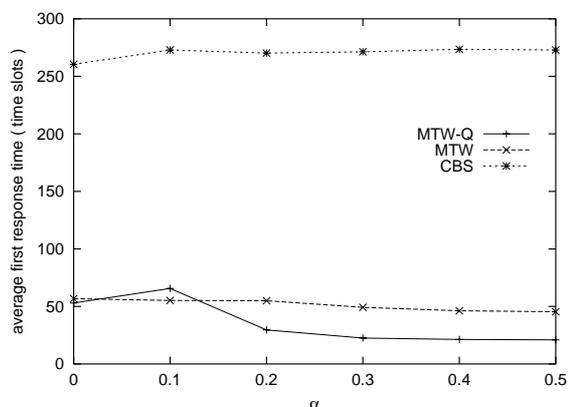


図 6: α と平均第一応答時間 ($I = 15$) .

式の想定環境と等価となるためである。

$I = 10$ とし, α を 0 から 0.7 まで変化させた場合の平均応答時間を図 3, 平均第一応答時間を図 4 に示す. $I = 10$ としているため, 多くのデータアイテムが複数のクエリにアクセス要求される. MTW-Q 方式が平均応答時間, 平均第一応答時間の双方において常に最も良い結果を示している. これは, MTW-Q 方式がクエリ単位で発行されるアクセス要求確率の変化を考慮して, 高い精度で各データアイテムに対する待ち時間の総和を計算することができるためである. 一方, MTW 方式は, 別のクエリでアクセス要求される同一のデータアイテムをそれぞれ別のデータとして, アクセス要求の発行数を計算するため, 最近放送が行われており, アクセス要求がほとんど発生していないデータアイテムを再び放送してしまうなど, 各データアイテムに対する待ち時間の総和を正確に計算することができない. また, CBS 法は, 全てのデータアイテムを等しい頻度で放送するため, クライアントのアクセス要求確率の偏りの変化を考慮することができない. したがって, 同じクエリでアクセス要求されるデータアイテム同士は近くに配置されるために平均第一応答時間

と平均応答時間の差は小さくなっているが, 平均応答時間, 平均第一応答時間ともに他の 2 方式に比べて長くなっている.

α が大きい場合は一部のデータアイテムが多くのクエリでアクセス要求されるためにクライアントのアクセス要求に偏りが生じ, それらのクエリに対する待ち時間の総和が頻繁に高まる. MTW 方式, MTW-Q 方式ともに, α が大きいほど平均第一応答時間を短くしているのは, これらの 2 方式が頻繁にアクセス要求されるアイテムに対する待ち時間の総和を考慮し, それらを頻繁に放送しているためである. 一方, 平均応答時間は, わずかに長くなっている. α が大きい場合は, 多くのクエリで要求されるデータアイテムを放送する効果が大きく, 頻繁に要求されるグループ H_1 のクエリの応答時間を短縮することができる. しかし, グループ H_1 のクエリを発行したクライアントは, 再び H_1 のクエリを発行する確率が高く, H_1 のクエリでアクセスされる h_1 のデータアイテムを放送すればするほど, h_1 に対する要求が高まるため, MTW-Q 方式はそれらを頻繁に放送することになる. このとき, h_2 から h_5 のデータアイテムに対する応答時間は逆に大きくなっ

てしまう。そのため、 α が大きい場合に平均応答時間が長くなってしまふ。

$I = 15$ とし、 α を 0 から 0.7 まで変化させた場合の平均応答時間、平均第一応答時間をそれぞれ図 5、図 6 に示す。ほとんどの場合において、MTW-Q 方式が最も短い平均応答時間、平均第一応答時間を示している。しかし、 α の値が大きい場合には、 $I = 10$ とした図 3、図 4 よりも平均応答時間の増加分が大きく、特に $\alpha = 0.4, 0.5$ の場合の平均応答時間は CBS 法よりもわずかに長くなっている。ある一回のクエリに注目した場合、その応答時間は、クエリの発行時刻と、次回の放送時刻が最も後になるデータアイテムの放送時刻との差となる。 α が大きくなるにつれて、MTW-Q 方式における各データアイテムごとの放送頻度の差は大きくなる。 I が大きく、一つのクエリで多くのデータアイテムをアクセス要求する場合、次回の放送時刻までの時間が非常に長いデータアイテムを要求する確率が高くなり、 I が小さい場合よりも平均応答時間が長くなる。

以上のように、MTW-Q 方式は他の 2 方式と比較して、ほとんどの場合において平均応答時間、平均第一応答時間ともに良い結果を示し、特に平均第一応答時間を大幅に短縮する。2 章で述べた、Web ページを放送する例などでは、取得済みのデータアイテムから順にアプリケーションで利用できるため、平均応答時間が同じ場合でも、他の方式と比べて感覚的な応答時間を短縮することができる。

ただし、一部のデータアイテムを頻繁に放送した場合、そのデータアイテムを含むグループの他のアイテムに対する要求が高まりやすくなり、結果として、他のグループのデータアイテムに対する応答時間が長くなる。MTW-Q 方式はそのような、あるデータアイテムを放送した場合の、他のデータアイテムの応答時間に対する影響を考慮することができない。そのため、データアイテムごとにアクセス要求されるクエリの数が大きく異なるような場合は、各データアイテムの待ち時間を考慮して、各時点で放送アイテムを決定するアプローチよりも、CBS 法のように、クエリでまとめてアクセス要求しやすいデータアイテムを近づけ、各データアイテムを一つずつ配置した固定のスケジュールに従って放送するアプローチの方が良い性能を示す。

6 おわりに

本稿では、プッシュ型放送において、クライアントが相関性をもつデータアイテム集合に対するクエリを、データ利用時間において複数回連続的に行う環境を想定した、データアクセスの平均応答時間短縮のためのスケジューリング方式の性能評価を行った。性能評価の結果、既存の方式と比較して、多くの場合で提案方式が平均応答時間、平均第一応答時

間ともに良い結果を示すことを確認した。

今後は、プロセスを単位として、連続してクエリが発行される環境を想定した、クライアント側でのキャッシング方式についても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「ネットワーク共生環境を築く情報技術の創出」、特定領域研究 (15017262) および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] Erçetin, Ö. and Tassiulas, L.: Push-based Information Delivery in Two Stage Satellite-terrestrial Wireless Systems, IEEE Transactions on Computers, Vol. 50, No. 5, pp. 506–518 (2001).
- [3] Franklin, M. and Zdonik, S.: “Data in Your Face”: Push Technology in Perspective, Proc. ACM SIGMOD'98, pp. 516–519 (1998).
- [4] 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).
- [5] Hsu, C. H., Lee, G., and Chen, A. L. P.: A Near Optimal Algorithm for Generating Broadcast Programs on Multiple Channels, Proc. Int'l Conf. on Information and Knowledge Management (ACM CIKM 2001), pp. 303–309 (2001).
- [6] Jiang, S. and Vaidya, N. H.: Response Time in Data Broadcast Systems: Mean, Variance and Tradeoff, Mobile Networks and Applications, Vol. 7, No. 1, pp. 37–47 (2002).
- [7] 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).
- [8] 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).
- [9] 内田 渉, 原隆浩, 西尾章治郎: プッシュ型放送における同時アクセスと連続アクセスを考慮したスケジューリング方式について, 情報処理学会マルチメディア通信と分散処理ワークショップ論文集, Vol. 2003, No. 19, pp. 115–120 (2003).
- [10] 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).
- [11] 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).