

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 関連データの放送スケジューリングについて

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

本稿では、関連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、時間間隔において複数回連続的に発行され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スケジューリング方式を提案する。提案方式は、データ間の関連性などのクライアントのアクセス特性および過去の放送履歴を用いて、各データアイテムに対するクライアントの待ち時間の総和を計算し、最もその値が大きいデータアイテムを放送する。

On Broadcast Scheduling of Correlated Data 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

In this paper, to reduce the response time of data access in push-based information systems, we propose a new scheduling strategy assuming an environment where simultaneous access requests for correlated data items are issued consecutively with think-time. The proposed strategy calculates the total waiting times for all data items according to the correlations among data items and the broadcast history and broadcasts the data item with the longest waiting 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プル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。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、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、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、さまざまなサーバ側でのスケジューリング戦略が提案されている[1, 3, 6]。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。

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に、

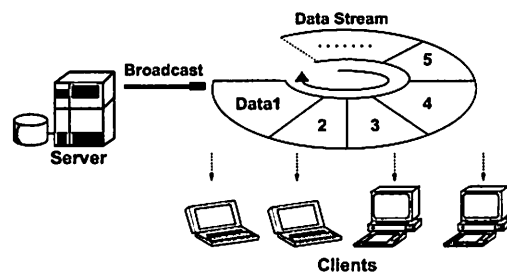


図1: プッシュ型情報システム

データ間に関連性が存在することが一般的である。クライアントが関連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これらの関連性を考慮することが有効である。

筆者らは、文献[4]において、関連性をも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を提案した。また、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[5, 7]において提案した。しかし、クライアントは、関連性をもつデータ集合に対して同時に複数のアクセス要求を発行し、それを時間間隔において連続的に繰り返すことがより一般的である。そこで本稿では、このような環境を想定し、データアクセスに対する平均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行わない。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章で本稿の想定環境のモデル化を行い、データアクセスの平均応答時間の計算方法を示す。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

2.1 アイテム間の相関性

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。

例えば、サーバが静的な Web ページを、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一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、しばらく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、一つの 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、そのページを閲覧する時間において、そのページからリンクされる他の Web ページを構成する複数のデータアイテムに対するアクセス要求を連続的に発行する。

このように、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三通りが考えられる。本稿では、相関性をもつデータアイテム集合に対して、同時に発行されるアクセス要求と、時間間隔において連続的に発行されるアクセス要求が組み合わさって発行されるような環境を想定する。

2.2 連鎖的なアクセス要求

本稿では、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 モデル化と平均応答時間

3.1 モデル化

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。

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。

あるクエリ q_i を発行しているクライアントは、 Q_i に含まれるデータアイテム d_j が放送されると、それに対するアクセス要求が満たされる。同じクエリ q_i を発行しているクライアントでも、クエリを発行した時刻に応じて、放送待ち状態にあるデータアイテムの集合は異なる。以下では、タイムスロット n ($n = 0, 1, \dots$) の先頭時刻 n における、アクセス要求がまだ満たされていないデータアイテム集合を S とするクエリ q_i の、システム全体での発行数の期待値を $P_{i,S}(n)$ とする。同様に、時刻 n における、 S を放送待ち状態のデータアイテム集合とするクエリ q_i を発行しているクライアントの、待ち時間の総和の期待値を $W_{i,S}(n)$ とする。

3.2 システム全体の平均応答時間

タイムスロット $n+1$ における、 S を放送待ち状態のデータアイテム集合とするクエリ q_i の発行数の期待値 $P_{i,S}(n+1)$ と待ち時間の期待値 $W_{i,S}(n+1)$ は、次の三つの要素を用いて計算可能である。

- タイムスロット n 内で、クエリ間の相関性にしたがって発行されたクエリや新規発生プロセスの先頭クエリとして発行されたクエリの発行数および待ち時間の期待値。
- タイムスロット n の先頭時刻において発生していた、 S を放送待ちデータアイテムの集合とするクエリの発行数および待ち時間の期待値。
- $B(n)$ の放送によって $B(n)$ に対するアクセス要求が満たされた結果、放送待ち状態のデータアイテム集合が S となるクエリ q_i の、タイムスロット n の先頭時刻における発行数および待ち時間の期待値。

ここで、 $g_{i,S}(n)$ 、 $G_{i,S}(n)$ を、タイムスロット n (時区間 $[n, n+1)$) に新規に発行されるクエリ i の発行数およびそれらの待ち時間の期待値とする。各タイムスロット m において満たされた、クエリ q_j の数の期待値を $\sigma_j(m)$ とする。新規に発行されるクエリ q_i は、 Q_i に含まれる全てのデータアイテムをアクセス要求しているため、 $S = Q_i$ となる場合のみ $g_{i,S}(n)$ は値をもち、その値は次式で与えられる。

$$g_{i,S}(n) = \begin{cases} \sum_{m=0}^n \sum_{j=1}^{M_q} \int_n^{n+1} \sigma_j(m) \cdot (1 - \Psi) \cdot C_{ji} \cdot F_{ji}(u - m) du + \Lambda_i & (S = Q_i) \\ 0 & (S \neq Q_i) \end{cases} \quad (1)$$

タイムスロット n 内のある時刻 u ($n \leq u < n+1$) において発行されたクエリの待ち時間は、時刻 $n+1$ において、 $(n+1-u)$ となる。したがって、タイムスロット n 内において新たに発行される、 S をアクセス要求するクエリ q_i の、時刻 $n+1$ における待ち時間の期待値 $G_{i,S}(n)$ は、次式で与えられる。

$$G_{i,S}(n) = \begin{cases} \sum_{m=0}^n \sum_{j=1}^{M_q} \int_n^{n+1} \sigma_j(m) \cdot (1 - \Psi) \cdot C_{ji} \cdot F_{ji}(u - m) \cdot (n+1-u) du + \frac{1}{2} \cdot \Lambda_i & (S = Q_i) \\ 0 & (S \neq Q_i) \end{cases} \quad (2)$$

ここで、 $r_{i,S}(n)$ 、 $R_{i,S}(n)$ を、タイムスロット n の先頭時刻において発行されている、 S をアクセス要求するクエリ q_i のうち、タイムスロット $n+1$ の先頭時刻においてなお S をアクセス要求しているクエリ q_i の発生数および待ち時間の期待値とする。 S が $B(n)$ を含む場合は、放送によってそれらのアクセス要求するデータアイテム集合は変化するため、 $r_{i,S}(n) = R_{i,S}(n) = 0$ となる。

$$r_{i,S}(n) = \begin{cases} P_{i,S}(n) & (B(n) \notin S) \\ 0 & (B(n) \in S) \end{cases} \quad (3)$$

$$R_{i,S}(n) = \begin{cases} W_{i,S}(n) + P_{i,S}(n) & (B(n) \notin S) \\ 0 & (B(n) \in S) \end{cases} \quad (4)$$

$x_{i,S}(n)$ 、 $X_{i,S}(n)$ を、 $B(n)$ に対するアクセス要求がその放送によって満たされた結果、放送待ち状態のデータアイテム集合が S となるようなクエリ q_i の発生数および待ち時間の期待値とする。そのようなクエリは、タイムスロット n の先頭時刻において、 S に $B(n)$ を加えたデータアイテム集合 $\{d | d \in S \vee d = B(n)\}$ をアクセス要求している。 Q_i に $B(n)$ が含まれない場合、もしくは $S = Q_i$ の場合はそのようなクエリ q_i は存在しないため、 $x_{i,S}(n) = X_{i,S}(n) = 0$ となる。

$$x_{i,S}(n) = \begin{cases} P_{i,\{d | d \in S \vee d = B(n)\}}(n) & (B(n) \in Q_i \wedge S \neq Q_i) \\ 0 & (B(n) \notin Q_i \vee S = Q_i) \end{cases} \quad (5)$$

$$X_{i,S}(n) = \begin{cases} P_{i,\{d | d \in S \vee d = B(n)\}}(n) + W_{i,\{d | d \in S \vee d = B(n)\}}(n) & (B(n) \in Q_i \wedge S \neq Q_i) \\ 0 & (B(n) \notin Q_i \vee S = Q_i) \end{cases} \quad (6)$$

以上より、 $P_{i,s}(n+1)$ 、 $W_{i,s}(n+1)$ は次の式で求められる。

$$P_{i,s}(n+1) = g_{i,s}(n) + r_{i,s}(n) + x_{i,s}(n) \quad (7)$$

$$W_{i,s}(n+1) = G_{i,s}(n) + R_{i,s}(n) + X_{i,s}(n) \quad (8)$$

また、タイムスロット n において満たされるクエリ q_i の数 $\sigma_i(n)$ は、タイムスロット n の先頭時刻において、 $B(n)$ のみをアクセス要求しているクエリの数に等しい。

$$\sigma_i(n) = \begin{cases} P_{i,\{B(n)\}}(n) & (B(n) \in Q_i) \\ 0 & (B(n) \notin Q_i) \end{cases} \quad (9)$$

タイムスロット n において満たされるクエリ q_i の待ち時間の総和 $\Omega_i(n)$ は同様に、時刻 n において $B(n)$ のみをアクセス要求しているクエリの待ち時間に等しい。

$$\Omega_i(n) = \begin{cases} W_{i,\{B(n)\}}(n) & (B(n) \in Q_i) \\ 0 & (B(n) \notin Q_i) \end{cases} \quad (10)$$

以上の議論より、タイムスロット T_1 から T_2 までのシステム全体の平均応答時間 $MRT(T_1, T_2)$ は、タイムスロット T_1 から T_2 の間に満たされた全てのクエリに対する待ち時間の総和を、クエリの総数で割ったものとなる。

$$MRT(T_1, T_2) = \frac{\sum_{n=T_1}^{T_2} \sum_{i=1}^{M_q} \Omega_i(n)}{\sum_{n=T_1}^{T_2} \sum_{i=1}^{M_q} \sigma_i(n)} \quad (11)$$

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

本稿の想定環境において、放送スケジューリングの問題は、式 (11) の値を最短とする放送スケジュール \mathbf{B} を求める問題となる。しかし、候補となるスケジュール \mathbf{B} は、 $(M_d)^{T_2}$ 通り存在するため、その中で最も応答時間を短縮するスケジュールを、実時間で計算することは困難である。そこで本稿では、文献 [5] 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、MTW-Q (MTW considering Queries) 方式を提案する。MTW-Q 方式は、これまでの放送スケジュールおよび $Q_i, \theta_i, C_{ij}, F_{ij}(t), \Lambda, \Psi$ 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。

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、前章でのべたとおり、全てのクエリ q_i 、 S について、 $P_{i,s}(n)$ および $W_{i,s}(n)$ 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。

4.1 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。新規のクエリが常に発行されるため、 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。

$$\pi_j(n) = \sum_{\{i|d_j \in Q_i\}} \sum_{m=v_j}^{n-1} \gamma_i(m) \quad (12)$$

$\gamma_i(n)$ は、式 (1) と同様に、次式で与えられる。

$$\gamma_i(n) = \sum_{m=0}^n \sum_{j=1}^{M_q} \int_n^{n+1} \sigma_j(m) \cdot (1 - \Psi) \cdot C_{ji} \cdot F_{ji}(u - m) du + \Lambda_i \quad (13)$$

4.2 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。

$$\Pi_i(n+1) = \Pi_i(n) - \sigma_i(n) + \gamma_i(n) \quad (14)$$

$\sigma_i(n)$ 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 $\Pi_i(v_i) - \sum_{m=v_i}^{n-1} \sigma_i(m)$ 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。

$$\sigma_i(n) = \Pi_i(v_i) - \sum_{m=v_i}^{n-1} \sigma_i(m) \quad (15)$$

4.3 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 (15) 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 (14) 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。

$$\omega_j(n+1) = \omega_j(n) + \sum_{\{i|d_j \in Q_i\}} \Pi_i(n) \quad (16)$$

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。

$$\omega_j(n+1) = \omega_j(n) + \pi_j(n) \quad (17)$$

$W_{B(n)}(n+1) = 0$ 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、長期的には最適な選択とは限らない。

5 性能評価

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する。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する。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一つのサイトのみ、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、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% に対するものであるという事実が文献 [2] において報告されている。そこで、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) に分割した。 C_{ij} は、乱数を用いて与え、 $q_i, q_j \in H_k, (1 \leq k \leq 5)$ の場合は 80%、 $q_i \in H_k, q_j \in H_l, (1 \leq k, l \leq 5, k \neq l)$ の場合は 20% の確率で $C_{ij} > 0$ とした。つまり、各クエリは同じグループのクエリとはそれぞれ 80%、別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。 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、全ての i に対して、 q_i が属するグループを H_k とすると、 $\sum_{\{j|q_j \in H_k\}} C_{ij} = 0.8$ 、 $\sum_{\{j|q_j \notin H_k\}} C_{ij} = 0.2$ とした。

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、5 つ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、 $z_j = 1/\alpha^{j \bmod 100}$ を、全てのデータアイテム d_j に与える。クエリ q_i がグループ H_k に属する場合は、グループ h_k に属するそれぞれのデータアイテムに乱数を与え、その乱数と z_j との積を評価値とし、評価値が最も大きい 5 つのデータアイテムを Q_i とした。 α の値が大きいほど、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

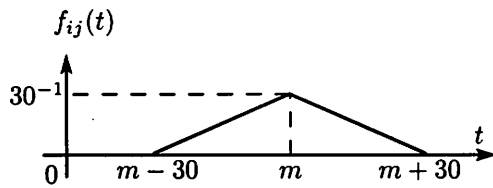


図 2: $F_{ij}(t)$.

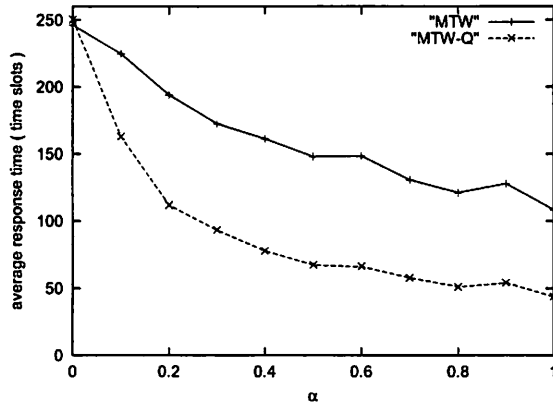


図 3: α と平均応答時間.

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 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, $(90, 110)$ の区間の一様分布で与えた.

比較対象としては, MTW 方式を用いた. 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.

以上のようなシステム環境において, α を 0 から 1.0 まで変化させた場合の, タイムスロット 0 から 1,000,000 までの平均応答時間 (MRT(0, 1,000,000)) を計算した. その結果を図 3 に示す.

図 3 より, ほとんどの場合において, クエリ単位のデータアクセスを考慮する MTW-Q 方式が良い結果を示している. また, α の値が大きいほど, つまり一部のデータアイテムが多くのクエリにおいてアクセス要求されるほど, MTW-Q 方式と MTW 方式の性能の差が大きくなっている. MTW 方式はクエリでアクセス要求されるデータアイテム集合をまとめて放送するため, クエリでアクセス要求するデータアイテムに重複がある場合でも, それらを考慮することができない. 一方, MTW-Q 方式は, 複数のクエリでアクセス要求されるデータアイテムが存在する場合でも, それぞれの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を正確に計

算することができるため, α の値が大きいほど, その効果が大きくなる.

反対に, $\alpha = 0$ の場合, 複数の Q_i に含まれるデータアイテムがほとんど存在しないため, データアイテムごとに待ち時間の期待値を計算し, 放送する MTW-Q 方式よりも, クエリ単位で待ち時間の期待値を計算し, まとめて放送する MTW 方式の方が良い結果を示している.

6 おわりに

本稿では, プッシュ型情報システムにおいて, 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, 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良い性能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 (研究拠点形成費補助), 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] 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).
- [3] 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).
- [4] 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).
- [5] 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).
- [6] 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).
- [7] 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).