

# プッシュ型放送のための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

内田 渉<sup>†</sup> 原 隆浩<sup>††</sup> 西尾 章治郎<sup>††</sup>

本論文では、相関性を持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放送環境を想定し、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する。提案方式は、データ間の相関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、その値が最も大きいデータアイテムを放送する。

## A Scheduling Strategy Considering Simultaneous and Consecutive Accesses for Push-based Broadcast

WATARU UCHIDA,<sup>†</sup> TAKAHIRO HARA<sup>††</sup> and SHOJIRO NISHIO<sup>††</sup>

In this paper, to reduce the response time of data access in push-based broadcast, 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、サーバ側での様々なスケジューリング方式が提案されている<sup>1),6),7),10),12),16)</sup>。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持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、相関性を持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17)において提案している。また、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の連続発行を考慮したスケジューリング方式を文献13)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ることが一般的である。そこで本論文では、クライアントが相関性を持つデータアイテム

<sup>†</sup> 株式会社エヌ・ティ・ティ・ドコモ人事育成部

Personel Development Department, NTT DoCoMo Inc.

<sup>††</sup> 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

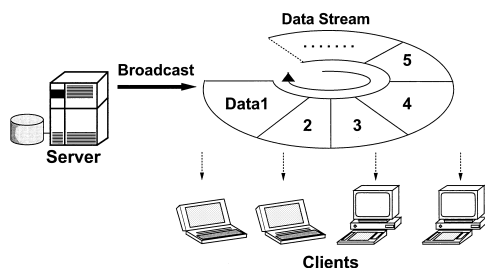


図 1 プッシュ型放送  
Fig.1 Push-based broadcast.

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する。また、提案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。

本論文では、次のようなシステム環境を想定する。

- 放送サーバは 1 つとし、プル型配達は行わない。
- 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。
- クライアント側でのキャッシングは考慮しない。

これらの前提条件には、実環境では必ずしも適当とは限らないものが存在する。これらの条件を想定しない場合における放送スケジューリングについては、7 章で議論する。なお本論文では、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 章で従来方式について説明し、4 章で本論文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。5 章で提案方式について説明する。6 章で提案方式の性能評価を行い、7 章で考察を行う。最後に 8 章で本論文のまとめを行う。

## 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一般的である。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 1 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、デジタル放送帯域などを利用して様々な音楽データを、各曲を 1 データアイテムとしてプッシュ型放送する場合を考える。つまり、ユーザは聴きたい曲を選び、要求を能動的に発行するが、クライアントはサーバにアクセス要求を発行せず、その曲のデータアイテムが放送されるのを待つ。通常、ユーザは同じアーティストや同じジャンル、同じアルバムの複数の曲目は同時にアクセス要求することが多く、それらに対するアクセス要求は同時に発生する。

次に、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放送する場合を考える。HTML ファイルや画像ファイルなどの Web 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、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、そのページを閲覧する時間において、リンク先の 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、互いに相関性を持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、順序性を持って、連続的にアクセス要求される場合がある。そこで本論文では、最も一般的な環境として、相関性を持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、すべ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 1 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要求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。

本論文では、クライアントが、1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持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初めに発行するクエリを先頭クエリと呼ぶ。

各クライアントにおけるクエリ間の相関性やクエリでアクセス要求するデータアイテム集合、データ利用時間の確率密度関数は、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. 従来のスケジューリング方式

#### 3.1 COP 方式

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。それらの中で最も代表的なものとして、Vaidya ら、Su らが提案した方式<sup>(12), (16)</sup> (以下、COP: Conventional OPtimal 方式と呼ぶ)がある。COP 方式では、データアイテムに対するアクセス確率に基づいて、各タイムスロットにおいて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放送することで平均応答時間を短縮している。

COP 方式では、クライアントの各データへのアクセス確率は時間的に一定と仮定しており、データ間の相関性について考慮していない。

#### 3.2 CBS 法

筆者らは文献 17) において、CBS (Correlation Based Scheduling) 法と呼ぶ方式を提案している。文献 17) では、クライアントは相関性を持つ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。

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。本論文の想定環境では、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。

#### 3.3 MTW 方式

筆者らは文献 13) において、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のすべてが、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、MTW (Maximum Total Waiting time) 方式を提案している。文献 13) では、同時にアクセス要求するデータアイテムは、1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。

MTW 方式は、各放送時において、各データアイテム  $d_i$  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、それが最も大きいデータアイテムを放送する。 $w_i(n)$  は、それまでの放送履歴および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  $p_i(n)$  から決定される。

MTW 方式は、すべ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1つまでと想定している。したがって、本論文の想定環境のように、複数のデータアイテムに対するクエリが時間間隔に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、待ち時間の期待値を正確に計算することができない。

### 4. モデル化と平均応答時間

#### 4.1 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  $C_{ij}$  ( $\sum_{j=1}^{M_q} C_{ij} = 1, C_{ii} = 0$ ) とする。また、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  $\Lambda$ 、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  $\theta_i$  とする。このとき、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  $\Lambda_i$  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  $\Psi$  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すべ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。

#### 4.2 システム全体の平均応答時間

$P_{i,S}(n+1)$  と  $W_{i,S}(n+1)$  は、次の3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従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実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実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  $n$  (時間区間  $[n, n+1)$ ) に新規発行されるクエリ  $i$  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。新規発行されるクエリ  $q_i$  は、 $Q_i$  中の全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  $g_{i,S}(n)$  は値を持ち、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クエリの、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クエリ  $q_i$  のうち、 $B(n)$  の放送によって  $S$  に含まれるどのアイテムへの要求も満たされなかったもの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  $T_2$  の間に満たされたすべ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。

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

## 5. 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

### 5.1 準備

最適な放送スケジュールを求めるには、 $(M_d)^{T_2}$  通りの候補から、式 (11) の値 ( 応答時間 ) を最短とするスケジュールを発見する必要がある。しかし、一般に候補数が非常に多くなるため、これを実時間で計算することは困難である。そこで本論文では、文献 13) 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、 $d_j$  を含むす

べ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、すべてのクエリ  $q_i$  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、すべ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。

### 5.2 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。

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

ここで、 $\gamma_i(n)$  は、次式で与えられる。

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

### 5.3 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．

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

$\sigma_i(n)$  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(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  $d_i$  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,  $Q_i$  に含まれるすべ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．

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

#### 5.4 MTW-Q 方式

MTW-Q 方式のアルゴリズムを次に示す．

- (1) すべてのクエリ  $i$ , 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) すべてのデータアイテム  $d_j$  のうち,  $\omega_j(n)$  が最も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 1 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, 1 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  $d_i$  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) すべての  $q_i$  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．

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

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

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

$\omega_{B(n)}(n+1) = 0$  とする．

(7)  $n = n+1, v_{B(n)} = n$  とし, 手順 (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．

## 6. 性能評価

### 6.1 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 ページを構成するファイル (HTML ファイルや画像ファイル) のそれぞれを 1 つの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 1 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  $H_1, H_2, \dots, H_5$ ) に分割した．また, 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%,  $q_i \in H_1 \wedge q_j \in H_k$ , ( $2 \leq k \leq 5$ ) 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% の確率で相関性を持つ．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  $q_i \in H_k \wedge q_j \in H_1$ , ( $2 \leq k \leq 5$ ) の場合は 80%、 $q_i \in H_k \wedge q_j \notin H_k \wedge q_j \notin H_1$  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 80%、その他のグループのクエリとは 20% の確率で相関性を持つ。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、すべての  $i$  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、 $\sum_{\{j|q_j \in H_k \vee q_j \in H_1\}} C_{ij} = 0.8$ 、 $\sum_{\{j|q_j \notin H_k \wedge q_j \notin H_1\}} C_{ij} = 0.2$  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くなる。

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  $h_1, h_2, \dots, h_5$ ) に分割した。また、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。

実環境において、あるデータに対してユーザがアクセスする確率は、そのデータの識別子のべき乗に反比例するという法則 (Zipf 則)<sup>18)</sup> に従うことが一般的に知られている。そこで次式に従って、データアイテム  $d_j$  の配置数  $a_j$  を仮に決定した。 $h_k$  はデータアイテム  $d_j$  が属するグループとする。

$$a_j = \left\lceil \frac{20 \cdot I \cdot (j \bmod 100)^{-\alpha}}{\sum_{\{l|l \in h_k\}} (l \bmod 100)^{-\alpha}} \right\rceil \quad (18)$$

つまり、各グループのデータアイテムは、識別子の降順で多くのクエリに配置されることになる。 $\alpha$  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、グループ内の各クエリで要求するデータアイテムに偏りが存在することを示す。また、 $I$  の値が大きいほど、1 つのクエリで多くのデータアイテムにアクセス要求を発行するため、不適切なスケジューリングを行った場合に応答時間が長

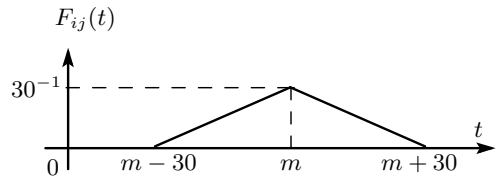


図 2  $F_{ij}(t)$   
Fig. 2  $F_{ij}(t)$ .

くなりやすい。

このように決定した仮の配置数のままでは、すべてのクエリで  $I$  個のデータアイテムをアクセス要求するように  $Q_i$  を決定できない場合がある。そこで、次の手順で各データアイテムの配置数を調整した。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  $d_j$  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。

プロセス発生時の先頭クエリがクエリ  $q_i$  である確率  $\Lambda_i$  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。タイムスロットごと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、要求発生の時間間隔 (データ利用時間) の確率密度関数  $F_{ij}(t)$  を、図 2 のように、 $(m - 30, 0)$ 、 $(m, 30^{-1})$ 、 $(m + 30, 0)$  を結び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  $(390, 410)$  の区間の一樣分布で与えた。

比較対象としては、MTW 方式、CBS 法、および COP 方式を用いた。CBS 法では、アイテム  $i, j$  間の相関性を、 $d_i, d_j \in Q_k$  となるクエリ  $q_k$  の数と  $\Lambda_i$  の積に比例するように与えた。MTW 方式では、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  $Q_i$  に含まれるすべ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。COP 方式では、MTW 方式と同様に  $Q_i$  を COP 方式におけるデータアイテムとし、それを単位として放送を行った。また、COP 方式に

表 1 各クエリで要求するアイテムに重なりがない場合の平均応答時間と平均第 1 応答時間 ( $I = 5, \alpha = 0$ )

Table 1 Average response time and average first response time when there exist no overlaps among data items accessed by different queries ( $I = 5, \alpha = 0$ ).

	MTW-Q	MTW	CBS	COP
平均応答時間 (タイムスロット)	262.1	192.7	307.4	199.2
平均第 1 応答時間 (タイムスロット)	116.2	185.0	295.4	191.1

おけるアクセス確率は、各クエリがプロセスの先頭で発行される確率  $\Lambda_i$  で与えた。

### 6.2 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  $\alpha$  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 (MRT(0, 1,000,000))、および  $Q_i$  中の 1 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 1 応答時間と表記する) を計算した。

Murray らは文献 11) において、インターネット上の Web ページが含む画像ファイル数は、平均 14.38、中央値 1 であると報告している。それに従い、 $I$  の値は、代表的なページが含むと思われるファイル数を参考にして 5, 10, 15 に設定した。 $I = 10$  の場合、 $\alpha = 0.7$  の時点で、最も多く配置されるデータアイテムの配置数が 20 になり、このデータアイテムはグループ内のすべてのクエリで要求される。そのため、 $\alpha$  は、クエリでアクセス要求されるデータアイテム集合にまったく偏りが無い状態である 0 から、 $(j \bmod 100)^{-\alpha}$  で与えられる配置数の比を保つ限界値 ( $I = 5$  では 0,  $I = 10$  では 0.7,  $I = 15$  では 0.5) の間の値を設定した。

#### 6.2.1 要求アイテムに重なりがない場合

$I = 5, \alpha = 0$  とした場合の平均応答時間、および平均第 1 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。この環境では、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い。したがって、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 1 つのデータアイテムと見なす場合、一度にアクセス要求されるデータアイテムは 1 つとする MTW 方式の想定環境と等価となる。そのため、MTW 方式が有効となる。

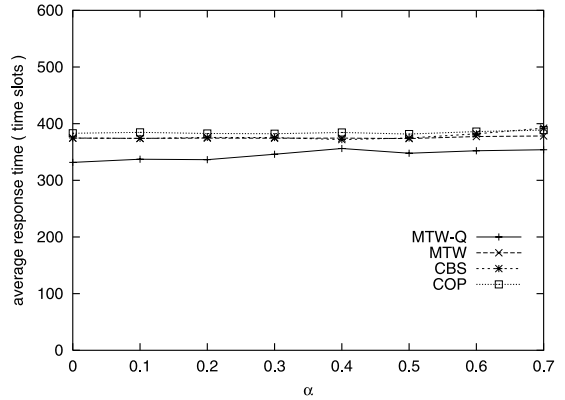


図 3 各クエリで要求するアイテム集合に重なりがある場合の平均応答時間 ( $I = 10$ )

Fig. 3 Average response time when there exist overlaps among data items accessed by different queries ( $I = 10$ ).

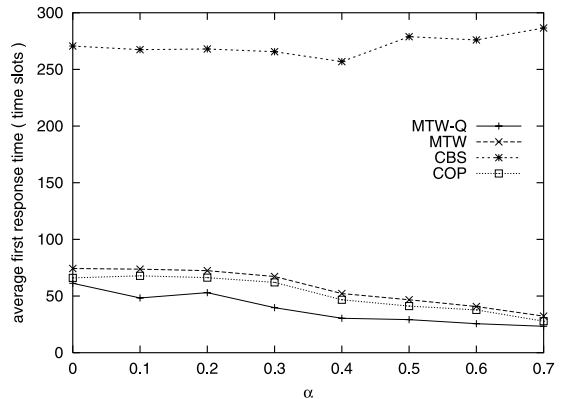


図 4 各クエリで要求するアイテム集合に重なりがある場合の平均第 1 応答時間 ( $I = 10$ )

Fig. 4 Average first response time when there exist overlaps among data items accessed by different queries ( $I = 10$ ).

またこの場合、クエリでアクセス要求したデータアイテム集合の間の積集合が空集合となるが、 $I > 1$  のため、MTW 方式と MTW-Q 方式は等価とはならない。MTW 方式と MTW-Q 方式が等価となるのは、 $I = 1$  でかつデータアイテム数  $M_d$  とクエリ数  $M_q$  が等しい場合に限られる。

#### 6.2.2 $\alpha$ の影響 ( $I = 10$ )

次に、 $I = 10$  とし、 $\alpha$  を 0 から 0.7 まで変化した場合の平均応答時間を図 3、平均第 1 応答時間を図 4 に示す。実際の Web ページでは、ロゴやバナー広告などの画像ファイルやインデクスとなる部分、スタイルシートが複数のページに共用される<sup>9)</sup>。このような状況は、図 3 および図 4 では、 $\alpha$  が 0.5 以上の場合が当てはまる (多くのアイテムの配置数が 1 で、



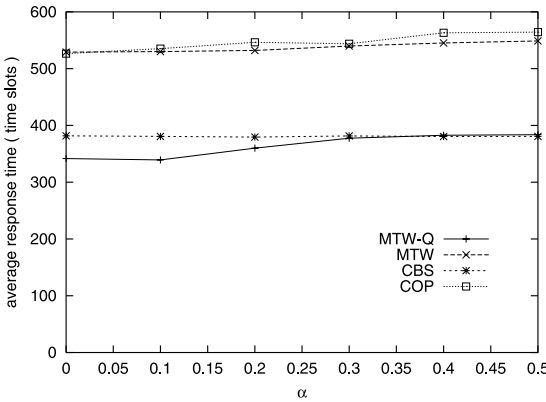


図 5 各クエリで要求するアイテムに重なりがあり、クエリで多くのアイテムを要求する場合の平均応答時間 ( $I = 15$ )

Fig. 5 Average response time when there exists overlaps among data items accessed by different queries and a query requests many items ( $I = 15$ ).

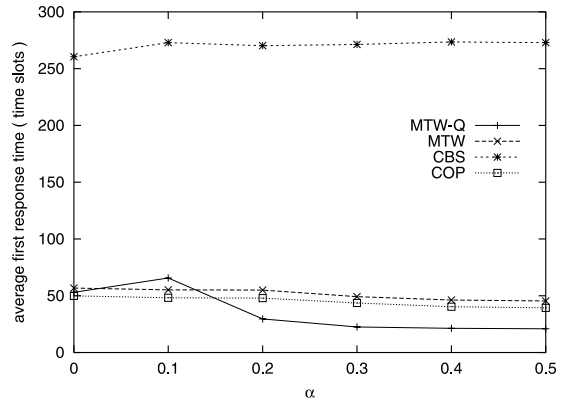


図 6 各クエリで要求するアイテムに重なりがあり、クエリで多くのアイテムを要求する場合の平均第 1 応答時間 ( $I = 15$ )

Fig. 6 Average first response time when there exists overlaps among data items accessed by different queries and a query requests many items ( $I = 15$ ).

一部のアイテムの配置数が大きい).  $I = 10$  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 1 応答時間の双方においてつね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 方式や COP 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. そのため、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、すべ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どうしは近くに配置されるために平均第 1 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 1 応答時間ともに他の 3 方式に比べて長くなっている.

$\alpha$  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 方式や COP 方式、MTW-Q 方式が  $\alpha$  が大きいほど平均第 1 応答時間を短くしているのは、これらの 3 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. ただ

し、それらを過剰に放送してしまった場合、頻繁にアクセスされないアイテムの放送頻度が小さくなる. クエリの応答時間は最も最後に放送されるデータアイテムの応答時間に依存するため、この場合、クエリの応答時間は長くなる.  $\alpha$  が大きい場合に平均応答時間が長くなってしまふのはこのためである.

### 6.2.3 $\alpha$ の影響 ( $I = 15$ )

$I = 15$  とし、 $\alpha$  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 1 応答時間をそれぞれ図 5、図 6 に示す.  $I = 10$  とした場合と比較して、5 つ多いデータアイテムを取得するために、クエリ単位のアクセスを考慮することができない MTW 方式や COP 方式の応答時間が 30% 以上増加している. それに対し、MTW-Q 方式の性能は変化せず、ほとんどの場合において最も短い平均応答時間、平均第 1 応答時間を示している. しかし、 $\alpha$  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い. たとえば、MTW-Q 方式の平均応答時間は、 $\alpha = 0.5$  の場合で  $\alpha = 0$  の場合に比べて約 15% 増加している. その結果、 $\alpha = 0.4, 0.5$  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.  $\alpha$  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、1 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.

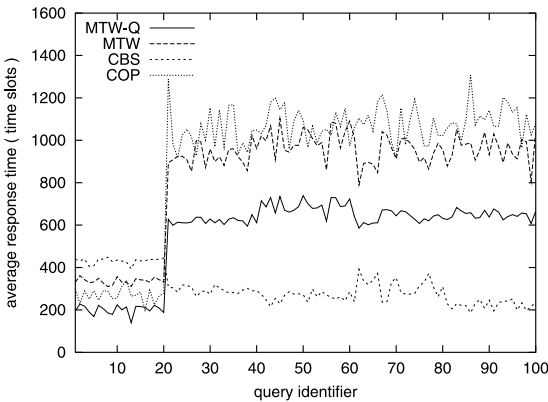


図7 クエリに要求するアイテムが多く、要求するアイテムに偏りが無い場合の  $q_i$  に対する平均応答時間 ( $I = 15, \alpha = 0.0$ )  
 Fig. 7 Average response time of  $q_i$  when many items are requested in a query and all items are equally requested by all queries ( $I = 15, \alpha = 0.0$ ).

#### 6.2.4 アイテム間の公平性

MTW-Q 方式は頻繁に発行されやすい  $H_1$  のクエリで要求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している。この場合、 $H_2$  から  $H_5$  のクエリで要求されるデータアイテムの放送頻度は低くなり、それらを発行した場合の応答時間が長くなるものと思われる。この影響を調べるために、個別のクエリに対する平均応答時間を評価した。評価では、 $I = 15, \alpha = 0$  とした。

評価結果を図7に示す。図7より、クエリ単位のアクセスを考慮しない MTW 方式や COP 方式と比較した場合、すべてのクエリに対して平均応答時間を短縮できている。すべてのデータアイテムを等しい頻度で放送する CBS 法と比較すると、アクセスが集中する  $H_1$  のクエリに対する平均応答時間を短縮しており、その結果、 $H_2$  から  $H_5$  までのクエリに対する平均応答時間は長くなっている。本論文では  $\alpha = 0$  の場合の結果を示したが、 $\alpha = 0.5$  とした場合も同様の傾向が見られた。この結果から、提案方式は人気のないクエリの応答時間を長くするが、その度合いは MTW 方式や COP 方式などの従来方式と比較して、特に問題となるレベルではないことが分かる。

ここで、プッシュ型放送において、平均応答時間を短縮するための放送スケジュールを作成するには、一部の人気のあるデータアイテムを優先して放送するというアプローチが一般的である。プッシュ型放送では放送帯域をすべてのアイテムで共用するため、一部のアイテムに対して多くの帯域を割り当てた場合、他のアイテムの平均応答時間が長くなってしまいうことは避けられない。COP 方式や MTW 方式を含め、これま

で提案された方式のほとんどがそのようなアプローチを用いて平均応答時間を短縮している。

また、プッシュ型放送における平均応答時間短縮のためのもう1つのアプローチとして、クライアント側におけるデータアイテムのキャッシングがあげられる。実環境ではクライアントは有限容量のキャッシュを持つ場合が一般的なため、人気のあるコンテンツを頻繁に放送することにより、平均応答時間が大きくなるコンテンツが存在する場合でも、各クライアントがそれらをキャッシュすることで、それらに対する平均応答時間を短縮することが可能である。筆者らの研究グループでは、クエリ単位のアクセス要求の連続発行を考慮したキャッシング方式を文献15)において提案している。

#### 6.3 評価結果のまとめ

以上のように、MTW-Q 方式は他の3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1応答時間ともに良い結果を示す。平均応答時間については、最大で CBS 法から約13%、MTW 方式から約35%、COP 方式から約37%の短縮を実現する。また、平均第1応答時間は、最大で CBS 法から約91%、MTW 方式から約50%、COP 方式から約48%の短縮を実現する。特に平均第1応答時間を短縮するため、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放送頻度が低くなり、結果として、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1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。

#### 7. 実環境への適用

提案方式は、筆者らがこれまでに提案した CBS 法や MTW 方式よりも、より一般的なクライアントの

アクセス形態を考慮してスケジューリングを行い、平均応答時間を短縮する。しかし、提案方式をプッシュ型放送に適用し、それを実運用する場合、さらに考慮すべき事項が残されている。本章ではそれらの要求事項を明らかにし、対応方法について議論する。

一般にプッシュ型放送を行うシステムを実環境で実現するためには、次のような要件が求められる。

- (1) 平均応答時間の短縮
- (2) 放送内容の更新への対応
- (3) 通信エラーへの対応
- (4) クライアントのアクセス制御
- (5) クライアントの消費電力の削減(特にクライアントがモバイル端末の場合)
- (6) クライアントのアクセス特性の取得

本論文の提案方式は、要件(1)を満たすためのものである。その他の要件のうち、要件(2)については文献3)、要件(3)については文献16)、要件(4)については文献4)、要件(5)については文献5)、8)においてそれぞれ有効な方式が提案されている。これらの方式は提案方式とは独立しているか、もしくは提案方式に容易に応用可能である。そのため、これらの方式を利用することで、要件(2)から要件(5)は対応可能である。したがって、新たに対応方法を考えなければならないのは、要件(6)のクライアントのアクセス特性の取得となる。

さらに、1章で述べた想定環境は、実環境では必ずしも正当ではないものが含まれる。以下ではそれらの事項および上述の要件(6)への対応方法について議論する。

### 7.1 クライアント側のキャッシング方式

本論文の想定環境では、クライアント側におけるデータアイテムのキャッシングについては考慮していない。この想定は、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に行ったものであり、実環境では必ずしも適当ではない。

クライアント側で効果的なデータアイテムをキャッシングするために、これまでに様々な方式が提案されている。それらの中で代表的なものに、文献2)で提案されているPT法がある。また、筆者らの研究グループではこれまでに、相関性を持つデータアイテム集合に対するクライアントの同時アクセスのみを考慮したCB-PT(Correlation Based PT)法<sup>17)</sup>や、連続アクセスのみを考慮したRIB-PT(Request Interval Based PT)法<sup>14)</sup>を提案している。さらに、本論文と同様に、同時アクセスと連続アクセスが組み合わさった形で発行される環境を想定したキャッシング方式に

ついて、文献15)において検討している。

### 7.2 異なるデータアイテムのサイズへの対応

本論文では議論の簡単化のため、データアイテムのサイズはすべて等しいものと想定したが、この想定は実環境では適当とは限らない。

データアイテムのサイズが等しくない場合、各データアイテムの放送にかかる時間が異なるため、提案方式におけるタイムスロット単位のスケジューリング手順を変更する必要がある。データアイテムのサイズを考慮した放送スケジューリングに関する研究としては、文献16)があげられる。文献16)のアプローチを応用し、提案方式が放送データアイテムを決定する際に用いる評価値(待ち時間の総和)を、データアイテムのサイズで割るように変更することで、データアイテムのサイズが異なる場合に対応可能である。

具体的には、放送データアイテムをタイムスロット  $n$  ごとではなく、各データアイテムの放送が終了するごとに決定するように変更する。それにともない、5章の各式で用いられるタイムスロット  $n$  (時間区間  $[n, n+1)$ ) を、 $n$  個目のデータアイテムの放送が行われる区間に変更する。ここで、 $n$  個目のデータアイテムの放送開始時刻を  $\beta(n)$  ( $n = 1, 2, \dots$ ) で表す。たとえば、式(13)の積分範囲は、 $\beta(n)$  から  $\beta(n+1)$  となる。

さらに、MTW-Q方式における手順(2)の「 $\omega_j(n)$  が最も大きくなるデータアイテムを調べ」という部分を「 $\omega_j(n)/L_j$  が最も大きくなるデータアイテムを調べ」に変更する。ここで、 $L_j$  はアイテム  $d_j$  のサイズである。

### 7.3 複数帯域における協調スケジューリング

本論文では、放送サーバは1つと想定した。この想定はより正確には、1つの放送ステーションの1チャンネルにおけるスケジューリングに着目することを意味する。異なった放送ステーションやチャンネルでは通常、まったく異なったデータを放送し、そのスケジュールはそれぞれがサービス対象とするクライアントのアクセス特性に基づいて独立に作成するものと考えられる。したがって、本論文の前提条件は多くの場合で適当である。

この前提条件があてはまらない環境としては、共通部分を持つ複数のデータ集合を、複数の放送ステーションやチャンネルを用いて、共通したクライアントに対して放送する場合が考えられる。この場合、他の帯域における放送によってアクセス要求が満たされることを考慮してスケジューリングを行う必要があるが、提案方式のスケジューリング手順の大部分を応用する

ことが可能である。

まず、複数の放送ステーションやチャンネルでまったく同じデータ集合をサービスしている場合の順序について述べる。ある放送帯域でデータアイテムの放送が終了するごとに、MTW-Q方式と同様に、要求するクエリの待ち時間が最も大きいデータアイテムを調べ、それを放送アイテムとする。そして、次にアイテムの放送が終了する帯域での放送終了時刻までに、新たに発行されるクエリ数や各データアイテムを要求するクエリの待ち時間を計算する。

一方、一部の共通部分を持つデータ集合を異なった放送帯域で放送している場合は、各放送帯域で MTW-Q方式に従ってスケジューリングを行う。そして、自帯域で放送しているデータアイテムが他帯域で放送された場合、それらのデータアイテムに対する待ち時間のリセットや、クエリの充足数の計算を行うことで対応可能である。

ただし、実環境におけるほとんどのクライアントは、同時に複数の帯域を受信することは不可能である。そのため、同じクエリで発行される複数のアクセス要求は同時に満たされないということを、クエリの発行数や各データアイテムに対する待ち時間の計算の際に考慮する必要がある。

#### 7.4 アクセス特性の取得

プッシュ型放送では、クライアントからのリアルタイムなアクセス要求はサーバへ送信されないため、スケジューリングの際はサーバがクライアントのアクセス特性に基づいて実際に発行されているアクセス要求を予測する必要がある。このアクセス特性は2章で述べたように、クライアント全体または一部からアクセスログを収集し、統計処理を行うことや、放送データアイテムの内容から解析的に決定することができる。アクセス特性を得るための具体的な方法として、どのようなものが有効かは応用システムに依存して決定する。

#### 8. おわりに

本論文では、プッシュ型放送において、クライアントが相関性を持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。提案方式はクエリ単位のデータアクセスを考慮し、クエリ間の相関性などのアクセス特性、および予測した過去のアクセス状況などから各データアイテムを要求するクエリの待ち時間の総和を求め、その値が最も大きいデータアイ

テムを放送する。

さらに本論文では、解析評価により、提案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、ほとんどの場合で提案方式が平均応答時間、平均第1応答時間ともに良い結果を示すことを確認した。提案方式は特に第1応答時間を短縮するため、Webページを放送する場合など、取得したデータアイテムから順に利用できる環境に適している。

本論文の提案方式を用いることによって、システム全体の平均応答時間を短縮することが可能であるが、システム全体とアクセス特性が異なるクライアントは、平均応答時間を短縮することができない。そのような場合、個々のクライアントが自身のアクセス特性を考慮してデータアイテムをキャッシュすることが有効である。そこで今後は、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検討する予定である。

謝辞 本研究は、文部科学省 21世紀 COE プログラム「ネットワーク共生環境を築く情報技術の創出」および特定領域研究(16016260)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. IEEE ICDE'96*, pp.276-285 (1996).
- 3) Acharya, S., Franklin, M. and Zdonik, S.: Disseminating Updates on Broadcast Disks, *Proc. VLDB'96*, pp.354-365 (1996).
- 4) Celik, A. and Datta, A.: DG: A Scalable Approach for Broadcasting Data Securely in Wireless Networks, *Wireless Networks*, Vol.9, No.3, pp.249-259 (2003).
- 5) Datta, A., Debra, E.V., Celik, A. and Kumar, V.: Broadcast Protocols to Support Efficient Retrieval from Databases by Mobile Users, *ACM Trans. Database Syst.*, Vol.24, No.1, pp.1-79 (1999).
- 6) Erçetin, Ö. and Tassiulas, L.: Push-based Information Delivery in Two Stage Satellite-terrestrial Wireless Systems, *IEEE Trans. Comput.*, Vol.50, No.5, pp.506-518 (2001).
- 7) Hameed, S. and Vaidya, N.H.: Efficient Algorithms for Scheduling Data Broadcast, *Wireless Networks*, Vol.5, No.3, pp.183-193 (1999).

- 8) Imielinski, T., Viswanathan, S. and Badrinath, B.R.: Energy Efficient Indexing on Air, *Proc. ACM SIGMOD'94*, pp.25–36 (1994).
- 9) 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).
- 10) 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).
- 11) Muray, B.H. and Moore, A.: Sizing the Internet, White paper, Cyveillance (2000).
- 12) Su, C.J., Tassiulas, L. and Tsotras, V.J.: Broadcast Scheduling for Information Distribution, *Wireless Networks*, Vol.5, No.2, pp.137–147 (1999).
- 13) 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.SIG9(TOD15), pp.28–38 (2002).
- 14) 内田 渉, 原 隆浩, 西尾章治郎: 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, 情報処理学会論文誌: データベース, Vol.44, No.SIG8(TOD18), pp.114–125 (2003).
- 15) 内田 渉, 原 隆浩, 西尾章治郎: プッシュ型放送のための同時アクセスと連続アクセスを考慮したキャッシング方式について, 電子情報通信学会データ工学ワークショップ (DEWS 2004) 論文集 (2004).
- 16) Vaidya, N.H. and Hameed, S.: Scheduling Data Broadcast in Asymmetric Communication Environments, *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: データ間の関連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).
- 18) Zipf, G.K.: *Human Behavior and the Principle of Least Effort*, Addison-Wesley (1949).

(平成 15 年 12 月 20 日受付)

(平成 16 年 4 月 12 日採録)

(担当編集委員 波多野 賢治)



内田 渉

2001 年大阪大学工学部情報システム工学科卒業。2004 年同大学院情報科学研究科博士後期課程修了。同年(株)エヌ・ティ・ティ・ドコモ入社。現在に至る。工学博士。データベースシステムおよび放送型通信に興味を持つ。



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, 2000 年電気通信普及財団テレコムシステム技術賞, 2004 年本学会研究開発奨励賞を各受賞。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(フェロー)

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, 2003 年より大阪大学大学院情報科学研究科長を併任。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, Data & Knowledge Engineering, Data Mining and Knowledge Discovery 等の論文誌編集委員。ACM, IEEE 等 9 学会の会員。