

## 放送型データベースシステムにおける適応的問合せ処理方式

加下雅一\* 寺田 努† 原 隆浩‡ 塚本昌彦‡ 西尾章治郎‡

近年、無線通信技術の発展にともない、サーバが携帯端末や PDA などの移動型クライアントにデータベースの内容を定期的に放送する放送型データベースシステムが注目されている。放送型データベースシステムにおける問合せ処理方式としては、オンデマンド型、クライアント型および筆者らが提案した協調型方式などがある。問合せ発生間隔や問合せ結果のサイズ等の環境に応じて最適な手法は異なるが、システムの状況は常に変化しつづけるため、静的に最適な方式を決定できない。そこで本稿では、問合せ処理時点で最も性能が良い方式を適応的に選択する適応的選択型方式を提案する。提案方式ではサーバへの問合せ到着間隔に着目し、平均応答時間が小さく、問合せ成功率が大きくなるように方式の選択を行う。提案方式を用いることで、3 方式を単独で用いる場合と比べて平均応答時間および問合せ成功率を改善できる。

### An Adaptive Query Processing Method for Database Broadcasting Systems

Masakazu KASHITA\* Tsutomu TERADA† Takahiro HARA‡  
Masahiko TSUKAMOTO‡ Shojiro NISHIO‡

In the recent evolution of wireless communication technology, there has been an increasing interest in the broadcast database system where a server periodically broadcasts contents of a database to mobile clients such as portable computers and PDAs. There are three query processing methods in the database broadcasting system; the on-demand method, the client method, and the collaborative method which we proposed. Though the performance of each method changes according to the interval of query generation and the size of the query result, it is difficult to choose the optimal method among them statically. In this paper, we propose an adaptive method which chooses the optimal method at the moment of query processing. This method takes into account the interval of query generation and not only reduces the response time but also increases the success rate of query processing.

## 1 はじめに

近年、無線通信技術の発展にともない、サーバがクライアントへの広い帯域幅を利用して多種のデータを周期的に放送し、クライアントは必要なデータのみを選択して取得する放送型情報システムが注目されている。放送型情報システムでは、クライアント数が増加してもデータ配信のコストがほとんど変わらないため、クライアント数が多い場合に通信品質を落さず情報配信ができ、データアクセスのスループット向上が期待できる。

これまでに、放送型情報システムの性能向上を目的とし、放送データのスケジューリング戦略 [1, 4, 9, 10, 11, 7], クライアント側のキャッシュ戦略 [1], データ更新の反映 [2], プッシュ型とプル型の融合戦略 [3, 6], 放送を用いたプル型通信におけるアイ

テムのプリフェッチ戦略 [5] など多くの研究が行われている。

これらの研究では、放送データを単なるデータアイテムとして扱っており、具体的な放送内容やデータ形式に基づいてシステムの効率化を行っているものは少ない。しかし、放送型情報システムでは、アプリケーションに依存してハイパーリンク形式やリレーショナルデータモデル形式など、様々なデータ形式が存在するため、放送するデータの内容や形式に則したデータ処理機構が性能向上の重要な要因となる。そこで本稿では、サーバがリレーショナルデータベースの内容を繰り返し放送し、ユーザが放送されるデータベースに対して問合せを発行する環境を想定する。このようなシステムを放送型データベースシステムと呼ぶ。

放送型データベースシステムにおける問合せ処理方式としては、オンデマンド型、クライアント型、および筆者らが文献 [8] で提案した協調型方式などがある。問合せ発生間隔や問合せ結果のサイズといった環境の変化に応じて、各方式の平均応答時間や問合せ成功率といったシステム性能は変化するが、システムの状況は常に変化し続けるため、静的に最適な方式を決定できない。

\* 大阪大学大学院工学研究科情報システム工学専攻  
Dept. of Information Systems Eng., Graduate School  
of Engineering, Osaka University

† 大阪大学サイバーメディアセンターサイバーコミュニティ  
研究部門  
Cybercommunity Division, Cybermedia Center, Osaka  
University

‡ 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  
Dept. of Multimedia Eng., Graduate School of  
Information System and Tech., Osaka University

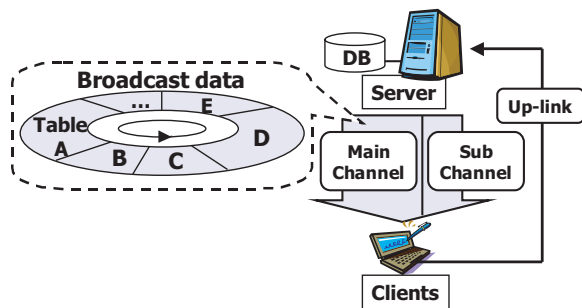


図 1: 放送型データベースシステム

そこで本稿では、問合せ処理時点で最も性能が良い方式を適応的に選択する適応的問合せ処理方式を提案する。提案方式ではサーバへの問合せ到着間隔に着目し、応答時間が小さく、また問合せ成功率が大きくなるように方式の選択を行う。

以下、2章では放送型データベースシステムおよび放送型データベースシステムにおける問合せ処理方式について説明する。3章で提案する適応的撰択型方式について述べ、4章で提案方式の性能評価を行う。5章で本研究の考察を行い、最後に6章で本稿のまとめと今後の課題について述べる。

## 2 放送型データベースシステム

本研究では、図1に示す放送型情報システムにおいてリレーショナルデータベースの内容を放送し、ユーザ(クライアント)が問合せを行う放送型データベースシステムを想定する。放送型データベースシステムは次に示す特徴をもつ。

**放送する内容:**サーバは、リレーショナルデータベースの内容を周期的に放送する。

**クライアント:**放送を受信するクライアントは、記憶領域、電力資源、処理能力の乏しい携帯端末である。

**ダウンリンク:**サーバからクライアントへの放送帯域は、2つのチャンネルに分割されているものとする。図1に示すように、広帯域のメイン放送帯域を用いて、データベースの内容を繰り返し放送し、狭帯域のサブ放送帯域を用いてそれ以外のデータを放送する。

**アップリンク:**クライアントからサーバへの狭帯域の通信チャンネルが存在する。クライアントは、アップリンクを用いて問合せをサーバに送信する。

放送型データベースシステムにおいて、クライアントがデータベースに問合せを行う場合の処理方式

として、オンデマンド型、クライアント型、協調型の3方式が考えられる。オンデマンド型方式ではサーバが問合せを処理して結果だけを返送し、クライアント型方式ではクライアントのみで問合せ処理を行う。協調型方式ではサーバとクライアントが協調して問合せを処理する。以下、各方式の処理手順と特徴を示す。

### 2.1 オンデマンド型方式

オンデマンド型方式における問合せ処理手順は以下のようになる。

1. クライアント上で、放送されるデータベースに対して、SQLによる問合せが発生する。問合せはアップリンクを経由してサーバに送信される。
2. サーバは問合せ処理を行い、問合せ結果をサブ放送帯域を利用して放送する。
3. クライアントは、自分宛に放送される問合せ結果を受信する。

オンデマンド型方式では、問合せ処理のすべてをサーバが実行し、クライアントは放送される結果を受け取るだけでよい。問合せを処理するためのディスク領域や計算を必要としない。また、発生する問合せ数が少ない場合、問合せ結果が放送されるまでの待ち時間が少なく、すぐに結果を取得できる。

しかし、オンデマンド型方式では、問合せが頻繁に起こる場合や問合せの結果サイズが大きい場合にサブ放送帯域が枯渇するため、クライアントが問合せ結果を受信するまで非常に長い時間がかかる可能性がある。

### 2.2 クライアント型方式

クライアント型方式における問合せ処理手順は以下のようになる。

1. クライアント上で、放送されるデータベースに対する問合せが発生する。
2. クライアントは問合せ発行後、メイン放送帯域を監視し問合せに関係するすべてのテーブルをローカルディスクに蓄積し、蓄積したテーブルに対して問合せ処理を行う。

クライアント型方式では、クライアント上だけで問合せ処理が完結するため、クライアント数が増加しても、1放送周期以内に問合せに関係する必要なすべてのデータを蓄積でき、問合せ結果を得られる

という利点がある。また、本方式はアップリンクが無い環境でも利用できる。

しかし、クライアント型方式では、問合せに関連するすべてのテーブルをディスクに蓄えて処理するため、クライアントのディスク領域を圧迫する。そのため、クライアントが十分な空きディスク領域をもっていない場合、問合せを処理できない可能性がある。

## 2.3 協調型方式

協調型方式における問合せ処理手順は以下のようになる。

1. クライアント上で、放送されるデータベースに対する問合せが発生する。問合せはアップリンクを経由してサーバに送信される。
2. サーバは、受け取った問合せに対して、一意の識別子（問合せ識別子）を割り当てる。放送するデータベースの各タプルには、処理用の識別子を書き込むための属性として、問合せ識別子、組合せ識別子が用意されている。サーバは受信した問合せを解析し、問合せ結果に含まれるタプルを調べ、そのタプルの問合せ識別子に割り当てた値を書き込む。クライアントはこの属性を参照することで、問合せ結果の再現に必要なタプルだけを蓄積できる。また、結合演算のように複数のテーブルから問合せ結果が得られる場合、問合せ結果のタプルを構成するそれぞれのタプルの組合せ識別子に同じ値を書き込む。クライアントは、組合せ識別子が等しいタプルを結合することで問合せ結果を再現できる。
3. サーバは、クライアントが問合せ結果に含まれるタプルのみを受信し、結果を再現するための処理ルール群を作成し、サブ放送帯域を用いて放送する。クライアントは自分当てのルール群を受信する。
4. クライアントは、受信したルール群によりメイン放送帯域から自動的に必要なタプルを受信し、問合せの結果を再現する。

協調型方式では、放送するルールのサイズが小さいため、問合せ結果をそのまま放送するオンデマンド型方式に比べ、放送帯域を圧迫しにくい。また、クライアントは問合せ結果に使用されるタプルだけを受信するため、クライアント型方式に比べ蓄積されるデータサイズが小さくなる。

表 1: 各方式の性能

方式	応答時間		蓄積する データサイズ
	問合せ発生頻度：密	疎	
オンデマンド型	大	小	小
協調型	中	大	中
クライアント型	小	中	大

## 2.4 各方式の特徴比較

ここまで述べてきた3方式の特徴を、表1に示す。問合せが頻繁に発生する場合、オンデマンド型方式ではサブ放送帯域を使いきってしまうため、問合せの発生からクライアントが問合せ結果を受け取るまでの平均応答時間は大きくなる。協調型方式では問合せ発生頻度の変化に影響を受けにくく、クライアント型方式では各クライアントが問合せに必要なタプルの受信と問合せ処理を行うため、発生頻度が変化しても平均応答時間は変化しない。問合せがほとんど発生しない場合、オンデマンド型方式では、サブ放送帯域が空いており、すぐに問合せ結果を放送できるため、平均応答時間を低くおさえられる。

また、オンデマンド型方式では問合せ結果のみが送信されるため、問合せ処理のためのデータをクライアントに蓄積する必要はないが、クライアント型方式では問合せに関係するすべてのテーブルを蓄積するため、蓄積データのサイズが大きくなる。協調型方式では、問合せ結果を構成するタプルだけを受信するため、クライアント型方式に比べて蓄積データのサイズが小さくなる。

協調型方式では、データベースに識別子領域が必要となるため、放送周期が長くなり、結果としてクライアント型方式よりも平均応答時間が長くなる。

## 3 適応的選択型方式

前章で述べたように、各方式を単独で用いた場合、システム環境の変化によって、その性能に優劣が生じる。

そこで本稿では、システム環境の変化に応じて、適応的に3方式を選択する適応的選択型方式を提案する。提案方式では、特に問合せ発生間隔に着目し方式の選択を行うことで、システム全体の性能を向上させる。

### 3.1 方式選択基準

2.4節で述べたように、各方式の優劣は主に問合せ発生頻度によって決まる。そこで、適応的選択型方式では問合せ発生頻度を表すパラメータとして、放送待ちデータサイズを用いる。放送待ちデータサ

イズとは、サブ放送帯域用の放送キューに蓄積されているデータのサイズである。放送待ちデータサイズが小さい場合は、データサイズの大きい問合せ結果を放送するオンデマンド型方式、中程度の場合は比較的小さいデータサイズの処理ルールを放送する協調型方式、小さい場合はクライアント型方式を用いることで、放送待ちデータサイズを常に小さく保ち、平均応答時間を小さくできる。

そこで、提案方式では 2 つの閾値  $OC\_Th$ 、 $CC\_Th$  ( $OC\_Th \leq CC\_Th$ ) を用いる。 $OC\_Th$  は、オンデマンド型方式と協調型方式を、 $CC\_Th$  は、協調型方式とクライアント型方式をそれぞれ切り替えるための閾値である。適応的選択型方式では、これらの閾値と放送待ちデータサイズを比較し、選択する方式を決定する。放送待ちデータサイズ  $S$  が  $OC\_Th$  より小さい場合、帯域が空いていると判断してオンデマンド型方式を、 $S$  が  $OC\_Th$  と  $CC\_Th$  の間の場合には、適度に空いていると判断して協調型方式を、 $S$  が  $CC\_Th$  より大きい場合、混雑していると判断してクライアント型方式を選択する。

### 3.2 方式選択アルゴリズム

適応的選択型方式における問合せ処理の流れは、次のようになる。

1. クライアントは、問合せをサーバに送信する。
2. サーバは、閾値と放送待ちデータサイズを比較して問合せ処理方式を決定する。
3. 各方式において問合せ処理に必要なデータを、サブ放送帯域を用いて放送する。放送されるデータは、オンデマンド型方式では問合せ結果、協調型方式では処理ルール、クライアント型方式ではメッセージ(クライアントに独自で問合せ処理をさせる要求メッセージ)である。
4. クライアントは、サブ放送帯域から自分宛てのデータを受信し、各方式に基づいて処理を行う。

## 4 評価

本章では、次に示す 2 つの評価基準について、オンデマンド型、クライアント型、協調型方式と比較し、適応的選択型方式の有効性を検証する。

平均応答時間: 問合せ発生から問合せ結果を得るまでの時間(但し、クライアントにおけるデータ処理にかかる時間は無視する。)

```
SELECT [Shop Table].*, [Goods Table].*
FROM [Shop Table], [Goods Table]
WHERE [Goods Table].Name = "Shoes"
```

図 2: 問合せ例

問合せ成功率: 発生した全問合せ数のうち、クライアントが問合せ結果を受け取れたものの割合

### 4.1 評価モデル

今回の本評価では、アプリケーションの例としてショッピングセンターにおける情報サービスを想定して、データベーススキーマと問合せモデルを決定した。このサービスでは、サーバはショッピングセンター内の店舗情報とその店舗で扱っている商品情報を放送し、ユーザは携帯端末を持ち歩きながら放送される情報を受信し、利用する。ユーザは「商品 A の画像とその商品を扱っている店舗の地図が欲しい」といった、画像を要求する問合せを行うものと想定する。

データベーススキーマは、飲食店や衣料店などの各ジャンルごとに店舗テーブル {店舗 ID, 店名, 画像, ...}, 商品テーブル {商品 ID, 店舗 ID, 商品名, 画像, ...} をもつものとした。店舗テーブルは '店舗 ID' を主キーとし、店舗の名前や地図画像を属性としてもち、商品テーブルは、'商品 ID' を主キーとし、その商品が販売されている店の識別子と商品画像を属性としてもつ。

問合せは SQL によって記述されるものとした。簡単化のため、店舗テーブルと商品テーブルのタプルサイズは等しいものとし、ユーザは店舗テーブルと商品テーブルを自然結合する問合せのみを行うものとした。ただし、自然結合を行う店舗テーブルと商品テーブルは同一ジャンル内であるとし、自然結合した結果のタプルに射影演算は行わないものとした。問合せ例を図 2 に示す。

### 4.2 シミュレーション環境

- 問合せ発生間隔は指数乱数で与え、指数分布の平均をパラメタとして指定する。
- タプル利用率は正規分布で与え、正規分布の平均と分散をパラメタとして指定する。タプル利用率とは、商品テーブルのうち、問合せの結果に使用されるタプルの割合を示す。
- 設定した時間内にクライアントが問合せ結果を受け取れない場合は、問合せ失敗とする。

表 2: 評価に用いるパラメタ

パラメタ名	値
店舗ジャンル名	10[個]
ジャンル内ショップ数	40[店]
ショップ内商品数	100[品]
タプルサイズ	5[KByte]
処理ルールサイズ	140[B]
メイン放送帯域	10[Mbps]
サブ放送帯域	1[Mbps]
タプル利用率の平均	3, 5*, 7[%]
タプル利用率の分散	1
クライアントのディスク容量	20[MByte]
OC.Th	0.5, 1*, 3[MByte]
CC.Th	2, 4*, 8[MByte]
タイムアウト	200[秒] (約 1.5 周期)
試行回数	10000

\*: デフォルト値

- クライアントのディスクサイズには制限があるものとし、問合せ処理の過程でクライアント側に許容サイズ以上のデータを蓄積しようとした場合は、問合せ失敗とする。
- 問合せ失敗が起ころっても、クライアントは問合せを再発行しない。

シミュレーションで用いたパラメタを表 2 に示す。

## 4.3 シミュレーション結果

### 4.3.1 問合せ発生間隔の影響

表 2 に示すデフォルト値を用いて行ったシミュレーションの結果を図 3、図 4 に示す。

図 3 は、問合せ発生間隔を変化させた場合の平均応答時間の変化を表す。オンデマンド型方式は、問合せがあまり発生しない場合に他の方式と比べて平均応答時間が小さくなるが、問合せが頻繁に発生するにつれて急激に平均応答時間が大きくなる。これは、オンデマンド型方式において、サブ放送帯域を用いて放送するデータ（問合せ結果）のサイズが大きく、すぐに帯域を使い切ってしまうためである。クライアント型方式では、問合せ発生間隔が変化しても平均応答時間は変化しない。また、協調型方式を用いた場合でも、処理ルールのサイズは小さいため平均応答時間がほとんど変化しない。協調型方式の平均応答時間が、クライアント型方式のものに比べて少し大きい理由は、各タプルの識別子領域の分だけ放送周期が長くなっていることと、ルールを受信するのに時間を要するためである。適応的選択型方式を用いた場合、ほぼすべての問合せ発生間隔において平均応答時間を低減できている。しかし、問合せ発生間隔が 3 秒以下の領域では、クライアント型方式に比べ平均応答時間が大きくなっている。この領域では問合せが頻繁に発生するため、すべての問合せがクライアント型方式で処理されるべきであるが、適応的選択型方式を用いた場合、協調型

やオンデマンド型方式で処理される問合せがまれに発生するため、このような差が生じる。また、問合せ発生間隔が 20 秒以上の領域では、オンデマンド型方式に比べ平均応答時間が大きくなっている。これは、問合せ発生間隔が大きい場合でも、問合せがパース的に発生すると放送待ちデータサイズが急激に増加し、オンデマンド型以外の方式が選択されてしまうためである。

次に、問合せ発生間隔を変化させた場合の問合せ成功率の変化を図 4 に示す。オンデマンド型方式を用いた場合、問合せが頻繁に発生すると平均応答時間が長くなるため、多くの問合せがタイムアウトしアクセスに失敗する。クライアント型方式を用いた場合、蓄積したデータのサイズがクライアントのディスク容量より大きくなった場合は問合せ処理に失敗する。協調型方式を用いた場合、問合せが頻繁に発生するとタプルに付加されている識別子領域が不足する。識別子領域を確保できなかった問合せは領域が開放されるまで待たなければならないため、平均応答時間が大きくなり、多くの問合せがタイムアウトする。一方、適応的選択型方式を用いると、問合せ発生間隔が小さい場合にはオンデマンド型方式が、大きい場合にはクライアント型方式が多く選択されるため、タイムアウトやディスク領域の不足が起ころにくく、問合せ発生間隔が変化しても高い問合せ成功率が保たれる。

### 4.3.2 タプル利用率の影響

タプル利用率  $r$  を、デフォルト値の 5[%] に対して 3, 7[%] と変化させた場合の平均応答時間と問合せ成功率の変化を、図 5 から図 8 に示す。

クライアント型方式では、クライアントは問合せに必要なすべてのテーブルを受信するため、タプル利用率の変化は結果に影響しない。協調型方式でも、タプル利用率の変化が処理ルールを受信するまでの時間や必要なタプルを受信し終えるまでの時間に影響しないため、平均応答時間、問合せ成功率ともにほとんど変化しない。オンデマンド型方式では、タプル利用率が大きくなると問合せ結果のサイズも大きくなるため、平均応答時間が大きくなる。また、平均応答時間の増加にともない、多くの問合せがタイムアウトしてしまうため問合せ成功率も悪くなる。

適応的選択型方式は、オンデマンド型方式が選択される割合に強く影響を受けるため、タプル利用率が大きくなると平均応答時間が大きくなる。

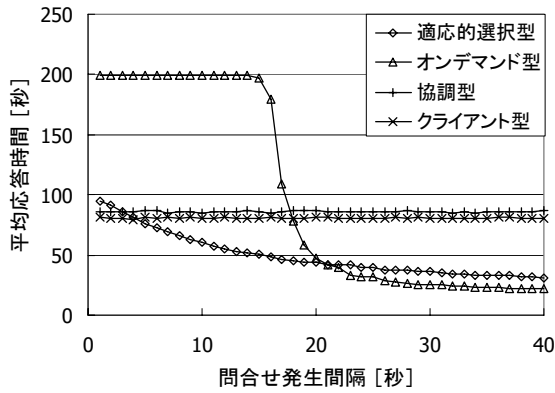


図 3: 平均応答時間

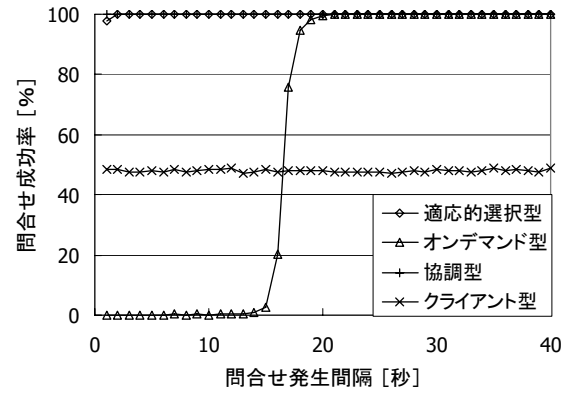


図 4: 問合せ成功率

#### 4.3.3 $OC\_Th$ の影響

適応的選択型方式において、 $OC\_Th$  をデフォルト値の 1[MB] に対して 0.5, 3[MB] と変化させた場合の平均応答時間と問合せ成功率の変化を、図 9, 図 10 に示す。

図 9 において、問合せ発生間隔が約 10[秒/問合せ] の点で、各  $OC\_Th$  のグラフが交差している。この理由は下記のとおりである。問合せ発生間隔が大きい場合、 $OC\_Th$  の値を小さく設定すると、問合せがバースト的に発生したときに、多くの問合せがオンデマンド型方式で処理されなくなるため平均応答時間が長くなる。また、問合せ発生間隔が小さい場合、 $OC\_Th$  の値を大きく設定すると、オンデマンド型方式で処理される問合せの数が増え、平均応答時間が長くなってしまふ。

一方、図 10 より  $OC\_Th$  の値が大きくなると問合せ成功率が小さくなるのがわかる。これは、 $OC\_Th$  を大きくすることでオンデマンド型方式で処理される問合せの数が増加するためである。

以上の結果から、問合せ発生間隔が大きい場合は、 $OC\_Th$  の値を大きくすれば平均応答時間を低減でき、問合せ発生間隔が小さい場合は、 $OC\_Th$  の値を小さくすれば平均応答時間を低減し、問合せ成功率も高くできることがわかる。

#### 4.3.4 $CC\_Th$ の影響

適応的選択型方式において、デフォルト値の 4[MB] に対して  $CC\_Th$  を 2, 8[MB] と変化させた場合の平均応答時間と問合せ成功率の変化を、図 11, 図 12 に示す。

$CC\_Th$  の値を大きく設定すると、クライアント型方式が選択されにくくなるため、問合せ発生間隔が小さい場合の平均応答時間が大きくなる。また、クライアント型方式が選択されにくくなることで、クライアントのディスク領域が不足する可能性が減

少し、問合せ成功率が向上する。

以上の結果から、 $CC\_Th$  の値を小さくすると平均応答時間を低減でき、 $CC\_Th$  の値を大きく設定すると問合せ成功率を向上できることがわかる。

## 5 考察

### 5.1 バースト的な問合せ発生について

提案方式を用いた場合、問合せがバースト的に発生すると放送待ちデータサイズが大きくなり、多くの問合せがオンデマンド型方式以外の方式で処理される。この現象は、問合せ発生間隔が大きく、オンデマンド型方式単独で十分処理可能な状況においても発生する。その結果、オンデマンド型方式を単独で使用した場合に比べて平均応答時間が長くなる。

選択アルゴリズムの性能向上については今後の検討課題であるが、この問題を解決する方法として、サーバが問合せ発生間隔の平均を統計的に求め、その値に基づいて各方式の理論的な選択割合を導出する手法が考えられる。サーバにおいてサブ放送帯域を用いて放送されるデータのキューに着目し、それに待ち行列理論を適用することで、理論上最適な選択割合が導出できるものと考えられる。

### 5.2 クライアント毎の方式選択について

提案手法では、クライアントのディスクサイズを考慮せず、放送帯域の混み具合のみを判断基準にして方式の選択を行っている。しかし、ディスクサイズの小さいクライアントに対してはオンデマンド型方式を優先的に用いるなど、サーバがクライアントのディスクサイズを考慮すれば、システム全体の問合せ成功率を向上できる。

具体的な方式については今後の検討課題であるが、問合せ自体にディスクサイズ情報を付加して送

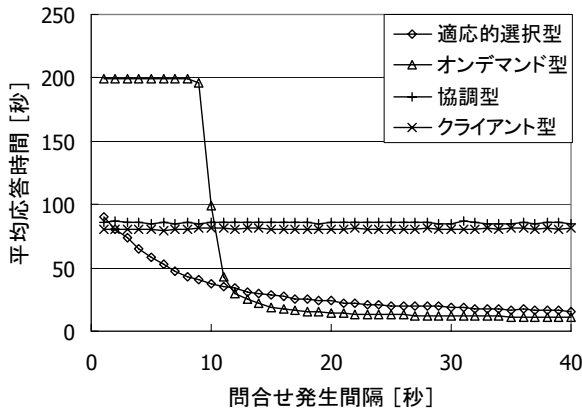


図 5: 平均応答時間 (r=3)

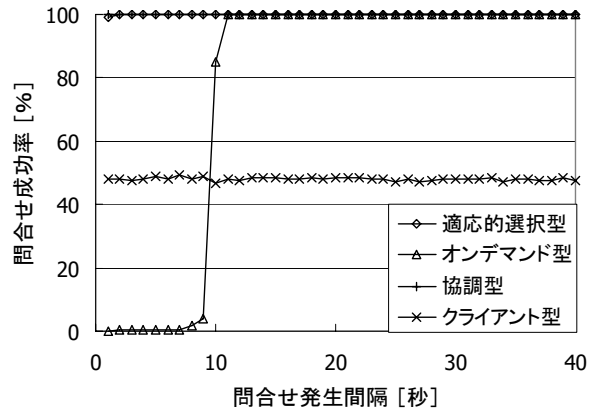


図 6: 問合せ成功率 (r=3)

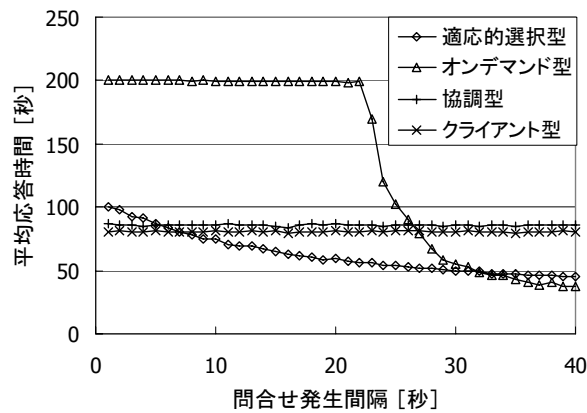


図 7: 平均応答時間 (r=7)

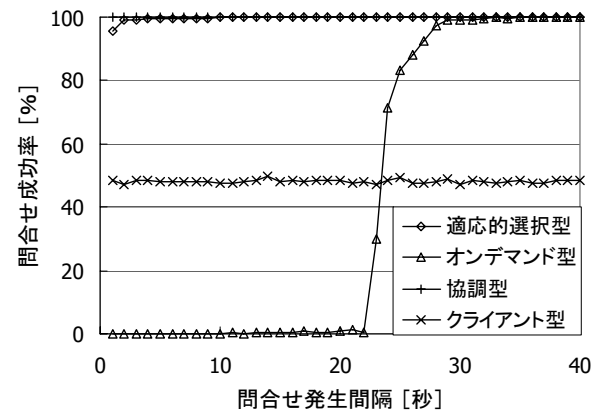


図 8: 問合せ成功率 (r=7)

信する方法や、前もってサーバにクライアントプロファイルとしてディスクサイズ情報をもたせる方法が考えられる。

## 6 おわりに

本稿では、放送型データベースシステムにおける効率的な問合せ処理方式として適応的問合せ処理方式を提案した。提案方式では、オンデマンド型、クライアント型、および筆者らが提案した協調型方式を、システム環境の変化に応じて適応的に選択する。本稿では、特に問合せ発生間隔に注目して方式の選択を行う手法を提案した。また、提案手法の有効性を検証するために、平均応答時間と問合せ成功率についてシミュレーション評価を行った。シミュレーション評価の結果、提案方式を用いることで、3方式を単独で用いる場合に比べ、様々なシステム環境で平均応答時間を短縮し問合せ成功率を向上できることを確認した。

今後の課題としては、待ち行列理論を導入した方

式選択方法の提案やクライアントのディスクサイズ等のクライアントの能力を考慮した方式選択方法の提案が挙げられる。

## 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金)、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」、文部科学省科学技術振興調整費若手任期付研究者支援「フィルタリングの数学的基盤の確立」、日本学術振興会若手研究 (B) (13780331)、および文部科学省特定領域 (14019063) 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] S. Acharya, M. Franklin, and S. Zdonik: "Broadcast Disks: Data Management for Asymmetric Communication Environments," *Proc. ACM SIGMOD*, pp. 199-210 (1995).

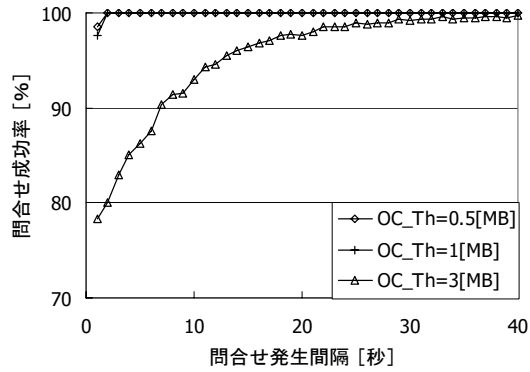
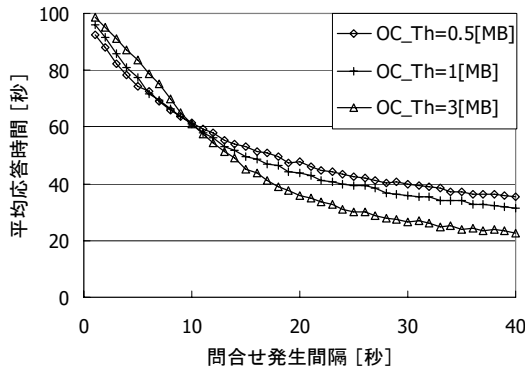


図 9:  $OC\_Th$  を変化させた場合の平均応答時間 図 10:  $OC\_Th$  を変化させた場合の問合せ成功率

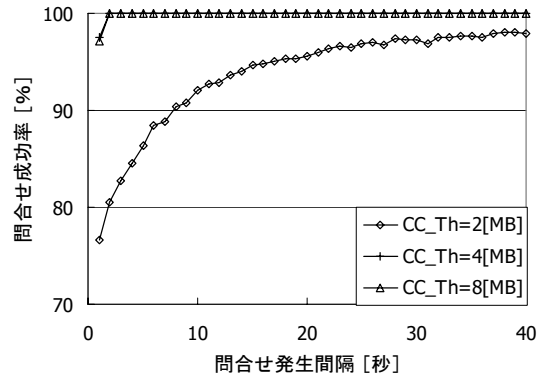
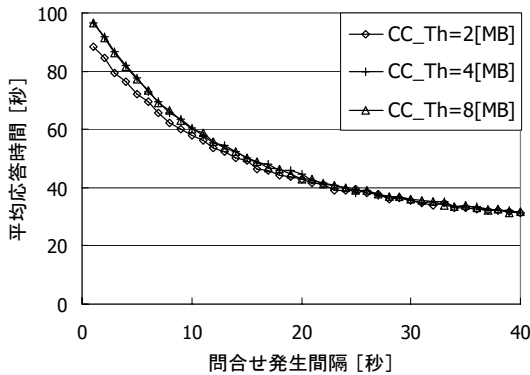


図 11:  $CC\_Th$  を変化させた場合の平均応答時間 図 12:  $CC\_Th$  を変化させた場合の問合せ成功率

- [2] S. Acharya, M. Franklin, and S. Zdonik: "Disseminating Updates on Broadcast Disks," *Proc. VLDB Conference*, pp. 354–365 (1996).
- [3] S. Acharya, M. Franklin, and S. Zdonik: "Balancing Push and Pull for Data Broadcast," *Proc. ACM SIGMOD*, pp. 183–194 (1997).
- [4] D. Aksoy, and M. Franklin: "Scheduling for Large-Scale On-Demand Data Broadcasting," *Proc. IEEE INFOCOM*, pp. 651–659 (1998).
- [5] D. Aksoy, M. Franklin and S. Zdonik: "Data Staging for On-Demand Broadcast," *Proc. VLDB Conference*, pp. 571–580 (2001).
- [6] 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).
- [7] Q. Hu, D. Lee, and W. Lee: "Performance evaluation of a wireless hierarchical data dissemination system," *Proc. MobiCom'99*, pp. 163–173 (1999).
- [8] 加下雅一, 寺田 努, 原 隆浩, 塚本昌彦, 西尾章治郎: "データベース放送システムにおける移動型クライアントのための問合せ処理方式," 情報処理学会研究報告 (モバイルコンピューティングとワイヤレス通信研究会 2001-MBL-17), vol. 2001, no. 46, pp. 47–53 (2001).
- [9] E. Yajima, T. Hara, M. Tsukamoto, and S. Nishio: "Interval Optimization of Correlated Data Items in Data Broadcasting," *Proc. of Int'l Conf. on Advances in Information Systems (ADVVIS 2000)*, pp. 127–136 (2000).
- [10] E. Yajima, T. Hara, M. Tsukamoto, and S. Nishio: "Scheduling Strategies of Correlated Data in Push-Based Systems," *Information Systems and Operational Research (INFOR)*, pp. 152–173 (2001).
- [11] E. Yajima, T. Hara, M. Tsukamoto, and S. Nishio: "Scheduling and Caching Strategies for Broadcasting Correlated Data," *Proc. ACM Symposium on Applied Computing (ACM SAC 2001)*, pp. 504–510 (2001).