2005－DBS－135　(1)
2005／1／20

放送型データベースシステムにおける モバイルクライアントのための問合せ処理方式

北島 信哉 †　寺田 努 ‡　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 1-5
E-mail: {kitajima.shinya, hara, nishio}@ise.eng.osaka-u.ac.jp

‡ 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
〒 567-0047 大阪府茨木市美穂ヶ丘 5-1
E-mail: tsutomu@cmc.osaka-u.ac.jp

近年，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手法としては，サーバが問合せ処理を行い，結果をクライアントに放送する方式，クライアントが問合せに関係するテーブル全体を蓄積して問合せ処理を行う方式，サーバとクライアントが協調して問合せ処理を行う方式の 3 方式が考えられる．これらの方式は，問合せ発生間隔や問合せ結果サイズ等の環境の変化に応じてその性能に優劣が生じるが，システム環境は常に変化し続けるため静的に最適な方式を選択できない．そこで本稿では，システムの状況に応じて，サーバにおける問合せのキューへの割込み処理や，キュー内の問合せの処理方式の変更を行う問合せ処理手法を提案する．提案方式を用いることで，従来方式と比べて平均応答時間を低減し，問合せ成功率を向上できる．

A Query Processing Method for Mobile Clients in Database Broadcasting Systems

Shinya KITAJIMA†　Tsutomu TERADA‡　Takahiro HARA†　Shojiro NISHIO†

†Dept. of Multimedia Eng., Graduate School of Information Science and Technology, Osaka University
1-5 Yamadaoka, Suita, Osaka 565-0871, Japan

‡Cybercommunity Division, Cybermedia Center, Osaka University
5-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan

In recent years, there has been an increasing interest in the broadcast database system where the 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; (i) the server processes a query and broadcasts the query result to the client, (ii) the client stores all data that are necessary in processing the query and then processes it locally, and (iii) the server and the client collaborate in processing the query. Though the performance of each method changes according to the situation such as the interval of query generation and the size of query results, it is difficult to choose the optimal method among them statically. In this paper, we propose a new query processing method which dynamically changes the order of queries submitted in the queue at the server and also changes processing methods for the queries according to the system situation. This method not only improves the response time but also increases the success rate of query processing compared with traditional methods.

1　はじめに

近年，無線通信技術の発展にともない，放送型通信を用いて情報を配信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，サーバはクライアントへの広い帯域幅を利用して各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とさず情報配信ができ，さらに，データアクセスのスループットの向上が期待できる．

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, 7, 9, 10, 11]，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型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，サーバが問合せ処理を行い，結果をクライアントに放送するオンデマンド型方式，クライアントが問合せに関係するテーブル全体を蓄積して問合せ処理を行うクライアント型方式，サーバとクライアントが協調して問合せ処理を行う協調型方式 [8] の 3 方式が考えられる．これらの方式は，問合せ発生間隔や問合せ結果サイズ等の環境の変化に応じてその性能に優劣が生じるが，システム環境は常に変化し続けるため静的に最適な方式を選択することは困難である．

そこで本稿では，システムの状況に応じて，サーバにおける問合せのキューへの割込み処理や，キュー内の問合せの処理方式の変更を行い，動的に適切な問合せ処理を実行する手法を提案する．さらに，シミュレーション評価により，提案方式が平均応答時間および問合せ成功率において優れていることを確認する．

以下，2 章では放送型データベースシステムについて述べる．3 章で提案方式について説明し，4 章では提案方式の性能評価を行う．5 章で提案方式についての考察を行い，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 のように，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送信し，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，以下に示す要素から構成される．

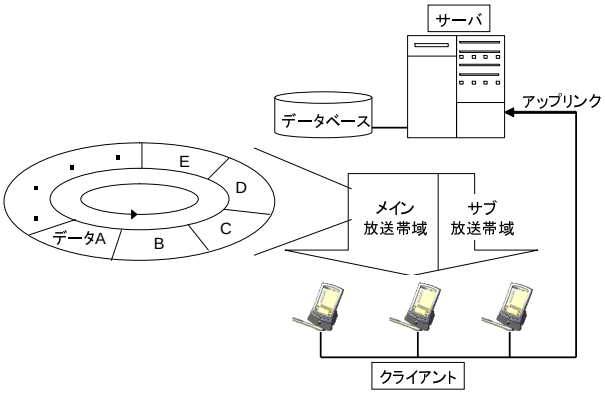

図 1: 放送型データベースシステム

サーバ　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．また，クライアントからの要求に応じて，問合せ処理を実行する．

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を想定する．

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 つのチャネルに分割されているものとする．サーバは，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ネルが存在する．クライアントは，このアップリンクを用いて問合せをサーバに送信する．

2.1　想定環境

本研究では，街中で不特定多数のユーザに周辺情報を配信するといったアプリケーションを想定している．その一例として，ショッピングセンターにおける情報サービスが挙げられる．このサービスでは，サーバがショッピングセンター内の広告情報や店舗情報，また店舗で扱っている商品情報を含むデータベースを放送し，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．サーバが放送しているデータベースは，店舗の地図画像や商品画像を含み，画像の数はデータベース全体で数千枚，サイズは数十メガバイトとする．クライアントは数百の規模で存在し，各クライアントは，放送

されている情報を絶えず受信していると想定する．通常，クライアントは放送を受信することのみで要求を満たしているが,「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」といった自然結合演算などの複雑な演算をともなう情報検索を行いたい場合には，サーバに対して問合せを発行するものとする．ユーザは，ショッピングをしながら欲しい商品を検索するため，数分程度の応答時間なら許容でき，問合せ処理にはリアルタイム性を要求しないものとする．ただし，各問合せにはデッドラインを設け，デッドラインの時間内にクライアントが問合せの結果が得られない場合は，その問合せは失敗となる．また，放送帯域は 5Mbps 程度とする．

2.2 問合せ処理方式

放送型データベースシステムにおいて，クライアントによる問合せを処理する方式として，以下の 3 方式がある．

2.2.1 オンデマンド型方式

クライアントがアップリンクを利用して問合せをサーバに送信し，サーバが問合せ処理を行った後でサブ放送帯域を用いて問合せ結果をクライアントに配信する．

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ため，問合せを処理するためのディスク領域を必要としない．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なく，クライアントはすぐに結果を取得できる．しかし，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，応答時間が長くなる可能性がある．

2.2.2 クライアント型方式

問合せに関係するすべてのテーブルをクライアントのディスクにいったん蓄え，必要なすべてのデータが揃ってからクライアント上で問合せ処理を行う．

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し，問合せ結果を得ることができる．また，アップリンクを使用しないため，アップリンクを用意できない環境でも動作する．しかし，クライアントの記憶領域を圧迫する，クライアントに大きな負荷がかかる，といった問題点がある．

2.2.3 協調型方式

クライアントは，アップリンクを利用して問合せをサーバに送信する．問合せを受け取ったサーバは，問合せを処理し，問合せ結果に含まれるタップルに処理用の識別子を付加するとともに，クライアントがデータを処理するためのルールを作成し，サブ放送帯域を用いてクライアントに送信する．クライアントは，自分宛に送信されたルールを受信すると，メイン放送帯域を用いて放送されるデータベースのうち，識別子を参照して必要なデータのみを蓄積し，問合せ結果を再現する．

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処理を行うことにより，クライアント型方式に比べてクライアントのディスク使用量を小さくし，オンデマンド型方式に比べてサブ放送帯域の占有時間を短くすることができる．しかし，各タップルにあらかじめ識別子領域を用意する必要があるため，放送周期が長くなる．また，問合せが頻繁に起こる場合には識別子数が不足するため，応答時間が長くなる可能性がある．

3 問合せ処理方式の選択手法

2.2 節で述べたオンデマンド型方式，クライアント型方式，協調型方式をそれぞれ単独で用いた場合，問合せ発生間隔や問合せ結果のサイズなどのシステム環境の変化によって，その性能に優劣が生じる．しかし，システムの環境に応じて既存の 3 問合せ処理方式の中から最適な方式を選択し，利用することができれば，システム全体の性能を向上できるものと考えられる．

そこで本章では，動的に処理方式を選択する手法を提案する．まず応答時間に基づいて単純に方式を選択する LRT 方式について述べ，次にその拡張方式について述べる．

3.1 LRT 方式

LRT (Least Response Time) 方式は，クライアントからの問合せがサーバに到着した時点で，応答時間が最も短い問合せ処理方式を選択する．

LRT 方式における問合せ処理の流れは，次のようになる．

1. **問合せの発生**
   クライアントは，SQL で記述された問合せと，端末の記憶容量や問合せのデッドラインなどの付加情報をアップリンクを利用してをサーバに送信すると同時に，メイン放送帯域を用いて放送されているデータベースの中から，問合せ処理に必要なテーブルを蓄積し始める．

2. **問合せ処理方式の選択**
   サーバは，オンデマンド型方式，クライアント型方式，協調型方式のうち，使用できる方式について，それぞれを選択した場合の応答時間を計算し，応答時間が最も短い方式を選択する．

3. **問合せ処理**
   サーバは決定した処理方式に基づいて，問合せを実行する．選択された方式がオンデマンド型方式の場合には問合せ処理を行い，結果を作成する．クライアント型方式の場合には，サーバでは何も行わない．協調型方式の場合には処理ルールを作成し，放送するタップルに対して識別子を付加する．

4. **クライアントへのデータ送信**
   サーバは，各方式において問合せ処理に必要なデータを，サブ放送帯域の放送キューに追加する．キューに追加されたデータは，先頭から順に放送される．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である．

5. **データ受信と問合せ結果の作成**
   クライアントは，サブ放送帯域から自分宛のデータを受信すると，各方式に基づいて処理を行う．問合せ結果が送信されてきた場合は，クライアントは結果をそのまま表示する．処理ルールが送信されてきた場合は，ルールに基づいて，識別子が付加されたタップルの受信を行い，問合せ結果を作成する．メッセージが送信されない場合は，クライアントは問合せ処理に必要なテーブルの受信が終わり次第，自ら問合せ処理を行う．

LRT 方式では，どの方式が選択されるかはあらかじめ分からないため，クライアントはサーバに問合せを送信するとともに，クライアント型方式が選択される場合を考慮して，問合せ結果の作成に必要なテーブルを蓄積し始める．

また，協調型方式のルールサイズはオンデマンド型方式の送信データサイズに比べて非常に小さいので，クライアントへの送信データを放送キューに追加する際，キュー内のオンデマンド型方式の送信データよりも前になるようにする．ただし，割り込んだ処理ルールにより，すでにキュー内にあるオンデマンド型方式の問合せがデッドラインを越えてしまう場合は，協調型方式を選択できないものとする．

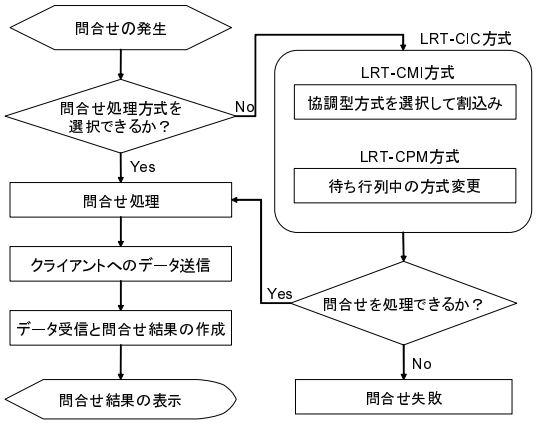

図 2: 拡張方式における問合せ処理の流れ

3.1.1 LRT 方式の問題点

LRT 方式では，問合せがサーバに到着した時点で応答時間が最も短い方式を選択するため，送信データサイズの大きなオンデマンド型方式の選択数が増加すると，放送待ちのキューが破綻する可能性がある．

また，協調型方式の処理ルールは，キュー内のオンデマンド型方式の送信データの前に割り込むことになるため，以降のオンデマンド型方式の問合せの応答時間がわずかに長くなる．協調型方式の選択数が増加すると，この割込みによる応答時間の増加が無視できなくなり，オンデマンド型方式を選択したキュー内の一部の問合せの応答時間がデッドラインを越えてしまう可能性がある．このような状況では，協調型方式を選択できなくなってしまう．

3.2 改良方式

LRT 方式において，オンデマンド型方式，協調型方式，クライアント型方式のうち，使用できる方式のいずれを選択しても応答時間のデッドラインを越えてしまう場合，サーバは問合せ処理方式を選択することができない．そこで，応答時間を低減しつつ，問合せ処理の成功率を上げるために，次の 3 つの拡張方式を提案する．

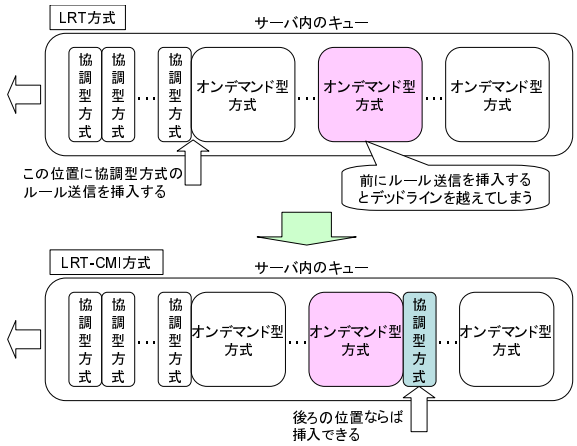

図 3: LRT-CMI 方式

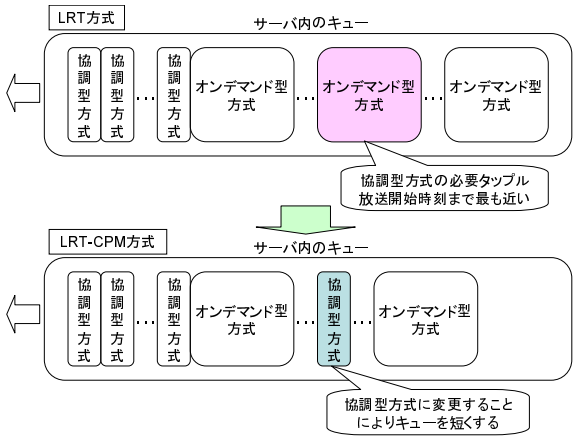

図 4: LRT-CPM 方式

拡張方式では，サーバはまず，LRT 方式に基づいて，オンデマンド型方式，クライアント型方式，協調型方式の中から問合せ処理方式を選択するが，選択できる方式がない場合や，どの方式を選択しても応答時間がデッドラインを越えてしまう場合は，特定の処理を実行する．拡張方式における問合せ処理の流れを，図 2 に示す．

3.2.1 LRT-CMI 方式

LRT 方式では，協調型方式を選択した問合せをキューに挿入すると，既にキュー内にあるオンデマンド型方式の問合せの一部の応答時間がデッドラインを越えてしまう場合，協調型方式を選択できなくなってしまう．

LRT-CMI (Cooperative Method Inserting) 方式では，図 3 に示すように，協調型方式を選択した際，その処理ルールをキューに挿入することで応答時間がデッドラインを越えてしまうオンデマンド型方式の問合せが存在する場合，その問合せの後ろに処理ルールを挿入する．この時，処理ルールを挿入しようとする問合せの応答時間がデッドラインを越えてしまう場合は，問合せは失敗となる．

3.2.2 LRT-CPM 方式

LRT 方式では，送信データサイズの大きなオンデマンド型方式の選択数が増加すると，キューが破綻してしまう．

LRT-CPM (Change of Processing Method) 方式では，図 4 に示すように，キュー内のオンデマンド型方式の問合せのうち，協調型方式に変更した場合に，処理ルール受信から必要タップル放送開始時刻までが最も短くなる問合せを，オンデマンド型方式から協調型方式に変更することにより，キューを短くする．ただし，LRT 方式によって失敗する問合せが，この変更処理によって成功する場合のみ，実行される．

3.2.3 LRT-CIC 方式

LRT-CMI 方式および LRT-CPM 方式はそれぞれ独立した処理であるため，これらを組み合わせることにより，さらに性能を向上できるものと考えられる．

LRT-CIC (Cooperative mehod Inserting and Change of processing method) 方式は，LRT-CMI 方式と LRT-CPM 方式を組み合わせたものである．LRT-CIC 方式では，LRT-CPM 方式における協調型方式への変更処理を優先して行い，変更処理を用いることができない場合は，LRT-CMI 方式における協調型方式を選択した問合せの挿入処理に移る．

4 評価

本章では，次に示す 2 つの評価基準を用いて，LRT，LRT-CMI，LRT-CPM，LRT-CIC の各方式の有効性をシミュレーション実験により検証する．

- **問合せ成功率**
  問合せ成功率は，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を示す．

- **平均応答時間**
  平均応答時間は，問合せが成功した場合における，クライアントが問合せを発行してから，問合せ結果を受け取るまでの平均時間を表す．ただし，クライアントおよびサーバにおけるデータ処理にかかる時間は，十分に小さいため無視する．

4.1 評価モデル

本評価では，2.1 節で示したようにショッピングセンターにおける情報サービスをアプリケーション例として想定し，データベーススキーマと問合せモデルを決定した．

データベーススキーマは，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する．簡単化のため，店舗テーブルと商品テーブルのタップルサイズは等しいものする．また，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，自然結合した結果のタップルには射影演算は行わないものとする．

シミュレーション評価では，各問合せに対し，デッドラインをパラメータとして与え，設定したデッドラインの時間内にクライアントが問合せを受け取れない場合は，問合せは失敗とする．また，クライアント型方式における必要な全テーブルを受信することができない端末の割合をパラメータとして与え，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，問合せは失敗とする．

4.2 シミュレーション環境

表 1 に，評価で用いるパラメータとその値を示す．また，問合せをサーバに送信する際，クライアントは 20 秒から 120 秒の間でデッドラインを選択できるものとし，その選択割合を表 2 に示すように設定した．

サーバには，ポアソン分布に従った間隔で問合せが到着するものとし，単位時間当たりの問合せ発生数をパラメータとして与えた．

表 1: 評価に用いるパラメータ

パラメータ名	値
全テーブルのサイズ [KByte]	15000
処理ルールのサイズ [KByte]	1
メイン放送帯域 [Kbps]	4000
サブ放送帯域 [Kbps]	500
最大識別子数 [個]	50
記憶領域の不足する端末の割合	0.5

表 2: デッドライン

秒数	20	30	40	50	60	70
割合	5%	5%	10%	20%	20%	10%
秒数	80	90	100	110	120	
割合	5%	10%	5%	5%	5%	

4.3 シミュレーション結果

4.3.1 問合せ成功率の比較

単位時間当たりの問合せ発生数を変化させたときの，LRT，LRT-CMI，LRT-CPM，LRT-CIC の各方式における問合せ成功率を図 5 に示す．

LRT 方式では，単位時間当たりの問合せ発生数が極端に少ない場合はオンデマンド型方式が，単位時間当たりの問合せ発生数が多くなると協調型方式が，単位時間当たりの問合せ発生数が非常に多くなるとクライアント型方式が多く選択される．単位時間当たりの問合せ発生数が増加するとともに，サブ放送帯域と識別子数が枯渇するため，問合せ成功率は低下する．

LRT-CMI 方式では，LRT 方式よりも問合せ成功率が高くなっている．これは，LRT 方式を用いた場合に失敗する問合せを，LRT-CMI では協調型方式の挿入位置を変更することで処理できる場合があるためである．同様に，LRT-CPM 方式も，LRT 方式より問合せ成功率が向上している．LRT-CPM ではキュー内のオンデマンド型方式の問合せ処理を協調型方式に変更するため，協調型方式の選択数が増加する．協調型方式における送信データサイズは，オンデマンド型方式における送信データサイズに比べ非常に小さいため，サブ放送帯域の枯渇を緩和し，結果として問合せ成功率が上がるものと考えられる．また，LRT-CMI はキューに問合せを追加

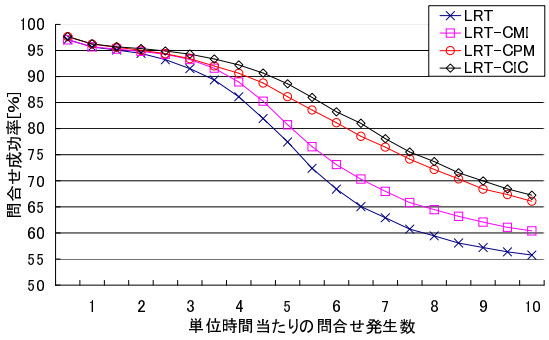

図 5: 単位時間当たりの問合せ発生数と問合せ成功率

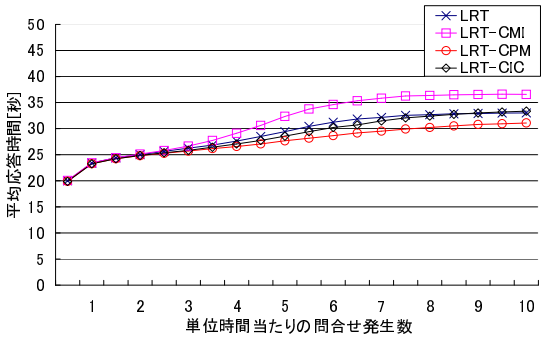

図 6: 単位時間当たりの問合せ発生数と平均応答時間

するのみの処理である一方，LRT-CPM はキューを短縮する効果もあわせもつため，処理できる問合せ数も増加し，LRT-CMI よりも問合せ成功率の上昇幅が大きくなる．

LRT-CIC 方式は，両方式の利点を組み合わせて用いるため，最も問合せ成功率が高くなっている．ただし，識別子数には限りがあるため，協調型方式の選択割合の増加は LRT-CPM に比べてわずかであり，飛躍的に問合せ成功率が上がることはない．

4.3.2 平均応答時間の比較

単位時間当たりの問合せ発生数を変化させたときの，LRT，LRT-CMI，LRT-CPM，LRT-CIC の各方式における平均応答時間を図 6 に示す．

LRT 方式では，単位時間当たりの問合せ発生数が増加するとともに，平均応答時間は長くなっている．これは，単位時間当たりの問合せ発生数が多くなると，サーバのキューが長くなり，オンデマンド型方式の応答時間が長くなるためである．一方，クライアント型方式，協調型方式の応答時間は，単位時間当たりの問合せ発生数の変化の影響をほとんど受けない．

LRT-CMI 方式では，LRT 方式よりも平均応答時間が長くなっている．これは，LRT 方式においてキューの前方に挿入する協調型方式の処理ルールを，LRT-CMI ではキューの中途に挿入するため，協調型方式の応答時間が長くなるためである．

LRT-CPM 方式では，LRT 方式よりも平均応答時間が短くなっている．LRT-CPM は，キュー内のオンデマンド型方式の問合せを協調型方式に変更するため，その問合せの応答時間は変更前に比べて長くなる場合がある．しかし，この変更処理によってサブ放送帯域の枯渇を緩和できるため，協調型方式の選択数が増加し，結果的に平均応答時間が短くなったものと考えられる．

LRT-CIC 方式では，平均応答時間は LRT-CPM に比べてやや長くなり，単位時間当たりの問合せ発生数が増加するとともに，LRT-CMI と LRT-CPM の中間値に近づく．

5 考察

5.1 問合せ処理方式の選択について

シミュレーション結果では，LRT-CIC 方式の問合せ成功率が最も高いが，平均応答時間は LRT-CPM 方式の方が短い．問合せ成功率の向上と平均応答時間の低減を両立することは難しく，どの提案方式が最も性能が良いかは，一概に判断できない．そのため，状況に応じて提案方式を使い分ける必要があると考えられる．状況に応じた方式の動的な切り替えに関しては，今後の課題である．

5.2 バースト的な問合せ発生について

問合せがバースト的に発生する状況下では，オンデマンド型方式の送信データがキュー内に存在すると，サブ放送帯域の枯渇が顕著に現れる．提案方式ではこの現象が多少改善されるものの，限界があるため，このような状況下ではオンデマンド型方式は選択するべきではない．

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，サーバが単位時間当たりの問合せ発生数の平均を統計的に求め，あらかじめ決定した閾値を越えた場合は，オンデマンド型方式を選択しないという手法が考えられる．

5.3 さまざまな環境における有効性

本稿では 1 つの代表的な状況を想定した評価について示したが，その他の環境での提案方式での有効性については論じていない．しかし，いくつかの異なる環境を想定して同様の評価を行っており，その結果，提案方式が LRT 方式に比べて有効であることを確認している．

6 おわりに

本稿では，放送型データベースシステムにおいて，動的に問合せ処理方式を選択する手法を提案した．提案方式では，特にサーバの放送キューに注目し，キューへの割込み処理やキュー内の問合せの処理方式の変更を行うことで，問合せ処理の効率を高めている．また，提案方式の有効性を検証するために，問合せ成功率と平均応答時間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から，提案方式は従来方式と比べて高い問合せ成功率および平均応答時間を達成できることを確認した．

今後の課題としては，閾値を導入した問合せ処理方式選択手法の提案や，時間とともに問合せの発生間隔が変化する環境における最適な問合せ処理方式選択手法の検討が挙げられる．

謝辞 本研究は，科学研究費補助金基盤研究 (B)(2)(15300033)，特定領域研究 (16016260)，および，文部科学省 21 世紀 COE プログラム（研究拠点形成費補助金）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) .

[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, 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] 箱根 聰, 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のためのサーバと移動型クライアントによる協調型問合せ処理方式,” 情報処理学会論文誌：データベース, Vol.44, No.SIG8(TOD 18), pp. 92–104 (2003).

[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 (ADVIS 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)*, Vol.39, No.2, pp. 152–173 (2001) .

[11] E. Yajima, T. Hara, M. Tsukamoto, and S. Nishio, “Scheduling and Cashing Strategies for Broadcasting Correlated Data,” *Proc. ACM Symposium on Applied Computing(ACM SAC 2001)*, pp. 504–510 (2001) .