社団法人　情報処理学会　研究報告
IPSJ SIG Technical Report

2006－DBS－138（28）
2006－GN－58（28）
2006／1／27

放送型データベースシステムにおける問合せ発生頻度に基づいた問合せ処理方式

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, tsutomu, hara, nishio}@ist.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って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，問合せの発生頻度に基づいて，動的に問合せ処理方式を選択する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 Based on Query Frequency in Broadcast Database 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-shi, Osaka 565-0871, Japan

In recent years, there has been an increasing interest on 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 broadcast database 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 query processing method based on the query frequency. This method improves not only the response time but also 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]，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適したデータ処理機構が性能向上の重要な要因となる．

そこで本研究では，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．

これまでに筆者らは，文献 [9] において，クライアントからの問合せがサーバに到着した時点で，3 方式のうち応答時間が最も短い方式を選択する問合せ処理手法と，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した．これらの手法では，問合せの発生頻度が高くなると，協調型方式を選択できない場合があった．また，問合せ発生間隔が時間とともに変化する環境を考慮していなかった．

そこで本稿では，文献 [9] における手法を拡張し，オンデマンド型方式を選択する際に，問合せの発生頻度に基づいて，余裕時間を設定する問合せ処理手法を提案する．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ことを示す．

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．最後に 5 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 に示すように，放送型情報システムにおいてサーバが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．

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としては，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．

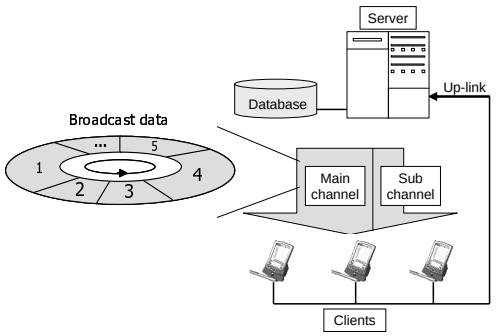

図 1: 放送型データベースシステム

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 つの帯域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，サイズは数百メガバイトとする．ユーザは数千の規模で存在し，各ユーザは常に放送を受信できる環境にあり，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，ユーザ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．問合せの応答時間は短いほど好ましいが，ユーザは発行した問合せの応答時間を知ることができないため，ユーザが自らで問合せを処理すべきか，サーバが問合せ処理を行った結果を待つべきかの判断はサーバが行うものとする．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，ユーザは各問合せにデッドラインを設定でき，デッドラインの時間内にユーザ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10Mbps程度とする．放送を受信するのみで問合せを発行しないユーザも存在すると想定し，サーバは，常に同じ放送スケジュールに基づいてデータベースを繰り返し放送し，放送スケジュールのインデックスの配信は行わない．ここで，サーバにおいて放送データの動的なスケジューリングは行わず，すべてのデータが一度ずつ放送される放送スケジュールが静的に作成されているとする．

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，クライアントは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短く，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

クライアントは問合せに関係するすべてのテーブルを自身の記憶領域にいったん蓄え，必要なすべてのデータが揃ってから，自ら問合せ処理を行う．

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，記憶領域による制約から問合せが処理できない場合がある，クライアントに大きな計算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処理ルールをもとに，問合せ結果の作成に必要となるタップルの放送開始時刻と放送終了時刻を把握し，その時間に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 [8]．

協調型方式では，クライアントは識別子を参照することで，問合せ結果の作成に必要なデータのみを蓄積するため，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できる．また，処理ルールは一般に非常に小さなサイズであることから，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，放送周期が若干長くなる．また，タップルに付加された識別子は，クライアントが問合せ結果を作成し終えるまで解放されない．識別子の最大数はあらかじめ決まっているので，問合せが頻繁に起こる場合には識別子が不足し，問合せ成功率が下がってしまう．

2.3 LRT 方式

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に応じて，その性能に優劣が生じる．そこで，システムの環境の変化に応じて既存の3つの問合せ処理方式の中から最適な方式を選択できれば，システム全体の性能を向上できると考えられる．

LRT（Least Response Time）方式 [9] では，クライアントからの問合せがサーバに到着すると，サーバはオンデマンド型方式，クライアント型方式，協調型方式の各方式を選択した場合の応答時間をそれぞれ計算し，応答時間が最も短い問合せ処理方式を選択する．応答時間が最も短い問合せ処理方式を選択してもデッドラインを越えてしまう場合には，問合せは失敗する．

また，サブ放送帯域の放送キューにおいて，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問合せ結果よりも前に挿入する．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．

LRT 方式における問題点を解決するために，筆者らは次の 3 つの拡張方式を提案した [9]．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，使用できる方式のうち，応答時間が最も短い問合せ処理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．

LRT-I 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，すで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-I（LRT-Inserting）方式では，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．また，そのようなオンデマンド型方式の問合せが複数存在する場合は，該当する問合せのうち，一番後ろにある問合せの後ろに挿入する．このとき，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．

LRT-C 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，キュー内のデータ量が増加し，デッドラインの制約から，以降の問合せ処理でキューに送信データを追加できない状況が続いてしまう可能性がある．

LRT-C（LRT-Change）方式では，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でキューを短くする．

ただし，この変更処理を行っても問合せが失敗する場合は，変更処理自体を行わない．

LRT-C/I 方式　LRT-I 方式および LRT-C 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，さらに問合せ成功率を向上できるものと考えられる．

LRT-C/I（LRT-I & LRT-C）方式は，LRT-I 方式と LRT-C 方式を組み合わせたものである．LRT-C/I 方式では，LRT-I 方式，LRT-C 方式を適用した場合のキュー内のすべての問合せの応答時間の増減を合計し，その値が小さい方式を適用する．また，LRT-I 方式，LRT-C 方式のどちらか一方のみが使用できる場合には，応答時間の増減の計算は行わず，その方式を適用する．

3　提案手法

従来の LRT 方式やその拡張方式では，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．したがって，挿入位置以降の問合せの応答時間がわずかに長くなり，協調型方式の選択数が増加すると，この応答時間の増加が無視できなくなる．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ることは許されないため，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．

そこで，オンデマンド型方式を選択する際に，問合せの発生頻度に基づいて，余裕時間を設定する拡張 LRT 方式を提案する．拡張 LRT 方式では，オンデマンド型方式の応答時間を計算する際，協調型方式の処理ルールを挿入するための余裕時間を設定しておき，デッドラインが余裕時間の分だけ短いものとして応答時間を計算する．これにより，オンデマンド型方式を選択された問合せは，キューに挿入された際，デッドラインまで必ず余裕時間以上の余裕が生じる．したがって，協調型方式の選択数が増加した場合でも，割込みによる応答時間の増分を余裕時間で相殺できるため，協調型方式が選択できない状況が少なくなる．

以下，3.1 節で拡張 LRT 方式における余裕時間の計算方法について説明し，3.2 節で拡張 LRT 方式における問合せ処理アルゴリズムについて述べる．

3.1　余裕時間の計算

十分な余裕時間があれば，オンデマンド型方式の送信データをキューに追加してから，そのデータが放送されるまでにキューに挿入されるすべての協調型方式の処理ルールによる応答時間の増分を相殺することができる．したがって，オンデマンド型方式の送信データをキューに追加してから，そのデータが放送されるまでにキューに挿入されるすべての処理ルールの送信にかかる時間を余裕時間とすればよい．しかし，キューに挿入される処理ルールの数

は，問合せの発生間隔などのシステム環境の変化に応じて異なる．そこで，余裕時間は問合せ処理方式を選択するたびに計算によって決定する．

キュー内に同時に存在できる処理ルールの数は，識別子の最大数に等しい．簡単のために，放送キューに，識別子の最大数と同数の処理ルールが挿入されていると仮定する．処理ルールは，クライアントに送信された後，問合せ処理が終わるとともに解放される．オンデマンド型方式が選択された問合せが 1 つ処理されるまでの間に解放される識別子の平均数 n_{av} は，オンデマンド型方式のみの平均応答時間と，協調型方式のみの平均応答時間の比で表すことができる．

実環境においては，問合せの発生間隔は一定ではないため，それぞれの方式における応答時間は，問合せの発生間隔に応じて大きく異なる．そこで，余裕時間を計算する際には，すべての問合せの平均応答時間ではなく，過去 x 個の問合せの応答時間の平均を用いる．この x をサンプル数と呼ぶことにする．オンデマンド型方式，協調型方式で処理した過去 x 個の問合せの応答時間の平均を，それぞれ $on_{av}(x)$，$co_{av}(x)$ と表すと，n_{av} の計算式は以下のようになる．

$$n_{av} = n_{max} \times \frac{on_{av}(x)}{co_{av}(x)}$$

求める余裕時間は，n_{av} 個の処理ルールを送信するために必要な時間に等しい．サブ放送帯域の帯域幅を B_s，協調型方式における処理ルールのサイズを s_{co} とすると，余裕時間 T_m は次式で求められる．

$$T_m = n_{av} \times \frac{s_{co}}{B_s}$$

余裕時間を計算する際には，識別子を使い切る状況を仮定しているため，問合せ発生数が少なく識別子に余裕がある状況では，余裕時間が余ってしまい，オンデマンド型方式の選択割合が下がる．一方，余裕時間を長めに設定すると，問合せ頻度が高くなった場合でも対応できるというメリットがある．余裕時間を短く設定すると，問合せ発生頻度が高くなった場合に余裕時間が不足し，問合せ成功率が下がる．

また，余裕時間の計算に用いるサンプル数が多すぎると，問合せ発生頻度の時間変化に迅速に適応できなくなる．サンプル数が少なすぎると，問合せごとの応答時間の差による影響が大きくなり，適切な余裕時間を設定できない．サンプル数が性能に与える影響については，4.3 節で調べる．

3.2 問合せ処理アルゴリズム

拡張 LRT 方式における問合せ処理アルゴリズムを以下に示す．

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サーバに送信する．さらに，クライアント型方式の問合せ処理に備えて，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．

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．ただし，オンデマンド型方式における応答時間を計算する際には，余裕時間を反映させるため，デッドラインが余裕時間の分だけ短いものとして計算する．

問合せ処理：サーバは決定した処理方式に基づいて，問合せ処理を実行する．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って結果を作成し，クライアント型方式の場合には何も行わない．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．

クライアントへのデータ送信：サーバは，各方式においてクライアントに配信するデータを，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．クライアント型方式では，サブ放送帯域を用いて放送されるデータはない．

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利用する．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，問合せ結果を再現する．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．

4 評価

本章では，次に示す 2 つの評価基準を用いて，提案方式の有効性をシミュレーションにより検証する．

表 1: 評価に用いるパラメータ

パラメータ名	値
全タップル数 [個]	4000
1 タップルのサイズ [KByte]	25
1 タップルに付加可能な最大識別子数 [個]	200
メイン放送帯域 [Mbps]	10
サブ放送帯域 [Mbps]	1
処理ルールのサイズ [KByte]	1
タップル利用率の平均	0.001
タップル利用率の標準偏差	0.00033
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	60	70	80	90	100	110
割合	5%	5%	10%	20%	20%	10%
秒数	120	130	140	150	160	
割合	10%	5%	5%	5%	5%	

問合せ成功率：過去 500 秒間に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．

平均応答時間：過去 500 秒間にクライアントが問合せ結果を受け取った全問合せの，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ったまでの平均時間．ただし，クライアントが問合せをサーバへ送信するのにかかる時間，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．

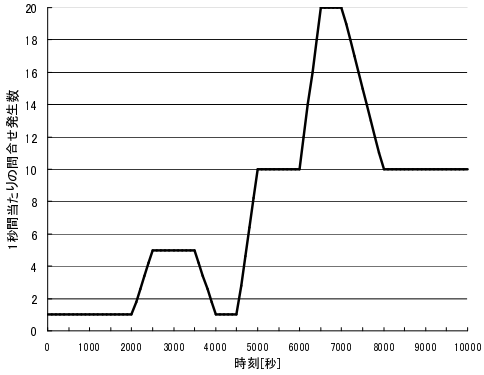

図 2: 問合せ発生数の変化

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結果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．各パラメータは，2.1 節で示した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決定した．タップル利用率は，テーブル内の全タップルに対する，問合せ結果に含まれるタップルの割合を表す．評価の際には，各問合せに対し，クライアント型方式における必要テーブルを，クライアントの問合せ発生時刻から 1 放送周期の範囲内でランダムに選択し，選択したテーブルの放送開始時刻と放送終了時刻をパラメータとして与えた．また，協調型方式における必要タップルを，クライアントの問合せ発生時刻から 1 放送周期の範囲内で問合せ結果サイズに応じてランダムに選択し，選択したタップルの放送開始時刻と放送終了時刻をパラメータとして与えた．さらに，問合せをサーバに送信する際，クライアントは 60 秒から 160 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，指数分布に従った間隔で問合せが到着するものとし，実環境においては問合せ発生間隔

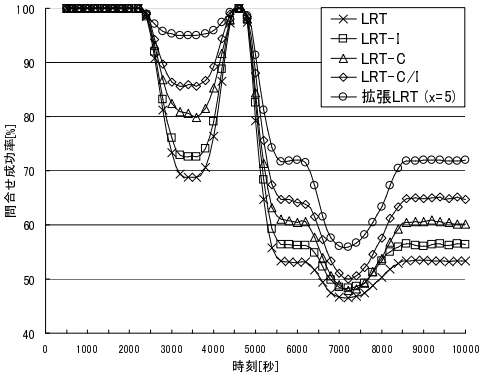

図 3: 1 秒間当たりの問合せ発生数と問合せ成功率

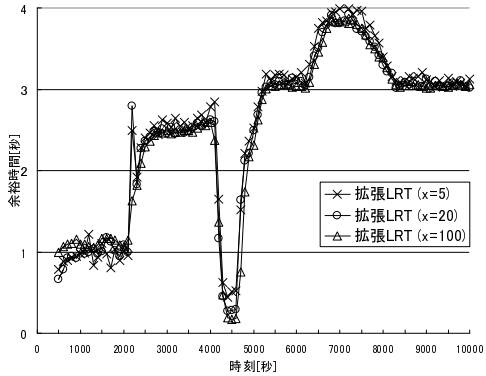

図 5: サンプル数の変化と余裕時間

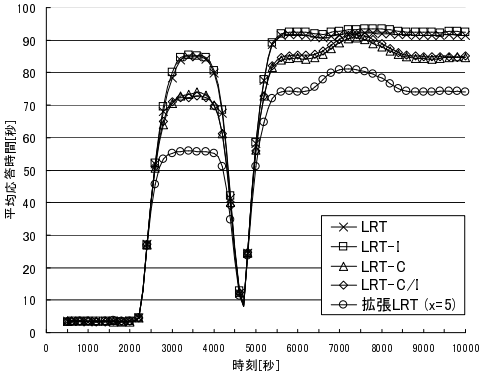

図 4: 1 秒間当たりの問合せ発生数と応答時間

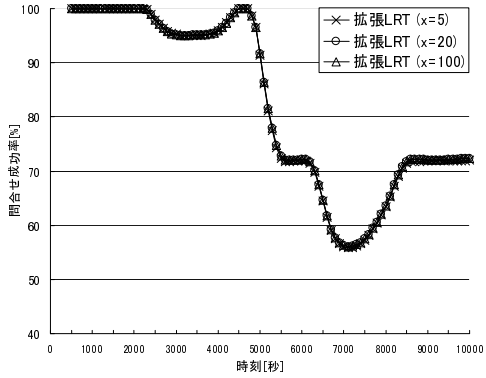

図 6: サンプル数の変化と問合せ成功率

は一定でないことを考慮して，1 秒間当たりの問合せ発生数を図 2 のように変化させた．また，問合せ発生数の時間変化に適応するよう，拡張 LRT 方式におけるサンプル数 x は，小さめの 5 に設定した．

4.3 シミュレーション結果

4.3.1 従来方式との比較

図 2 のように 1 秒間当たりの問合せ発生数を変化させたときの，従来方式と拡張 LRT 方式における問合せ成功率と平均応答時間を，図 3，図 4 に示す．

図 3 から，問合せ発生数の変化に関わらず，拡張 LRT 方式の問合せ成功率が最も高いことがわかる．従来方式では，協調型方式における処理ルールを放送キューに挿入する際，キューの後方にあるオンデマンド型方式を選択された問合せがデッドラインを越える場合には，協調型方式を選択できない．拡張 LRT 方式では，オンデマンド型方式の送信データをキューに挿入する際，あらかじめ協調型方式の処理ルールを挿入するための余裕時間を設定しているため，協調型方式が選択できない状況が減少する．したがって，協調型方式を選択できないために識別子が余ることが少なくなり，結果として問合せ成功率が高くなる．

また，図 4 から，拡張 LRT 方式の平均応答時間が最も短いことがわかる．問合せ発生数が少ない場合には，方式間の差はほとんど見られない．一方，問合せ発生数が多い場合には，オンデマンド型方式やクライアント型方式と比べ，必要なタップルのみを受信して問合せ処理を行う協調型方式の応答時間が短くなる．拡張 LRT 方式では，処理ルールを有効に利用することで，協調型方式の選択数が多くなるため，結果として平均応答時間が短くなる．

4.3.2 サンプル数の影響

サンプル数 x を $5, 20, 100$ と変化させたときの，拡張 LRT 方式における余裕時間を図 5 に，問合せ成功率を図 6 に示す．1 秒間当たりの問合せ発生数は，同じく図 2 のように変化させた．

図 5 から，サンプル数が少ない場合には，各問合せの応答時間にばらつきがあるため，余裕時間にもばらつきが出やすいが，問合せ発生頻度の変化にすばやく対応できることがわかる．逆に，サンプル数が多い場合には，過去 x 回の問合せの応答時間が

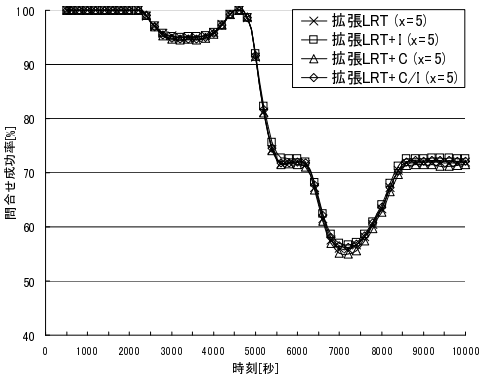

図 7: 従来方式と組合わせた場合の問合せ成功率

平均化されるため，余裕時間は大きく増減することはない．よって，急な問合せ発生頻度の変化には対応できない．

また，図 6 から，$x = 20$ のときの問合せ成功率が最も高く，また，$x = 100$ では，問合せ発生頻度が大きく変化した場合に問合せ成功率が低くなっていることがわかるが，問合せ成功率の差はわずかである．

4.3.3 従来方式との組合せ

従来方式と拡張 LRT 方式における処理は独立しているため，これらを組合わせることにより，さらに問合せ成功率を向上できると考えられる．

図 7 は，拡張 LRT 方式にそれぞれ LRT-I，LRT-C，LRT-C/I 方式を組合わせた場合の問合せ成功率を示している．1 秒間当たりの問合せ発生数は，図 2 のように変化させた．

図 7 から，基本的には拡張 LRT 方式とほとんど変わらないことがわかる．LRT-I 方式における挿入処理は，拡張 LRT 方式における余裕時間が十分でない場合に起こるため，拡張 LRT 方式に LRT-I 方式を組合わせることにより，余裕時間の不足分を補うことができる．しかし，図 7 から，多少の効果はあるものの，問合せ成功率はほとんど変わらないことがわかった．

5 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，問合せの発生頻度に基づいて動的に問合せ処理方式を選択する問合せ処理手法を提案した．提案方式では，協調型方式における処理ルールの挿入を考慮して，オンデマンド型方式を選択する際に余裕時間を設定している．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，時間とともに問合せ発生頻度が変化する環境において，提案方式は従来方式と比べ，問合せの成功率を高めるとともに，応答時間を低減できることを確認した．さらに，提案方式を従来方式と組み合わせることにより，問合せ成功率を向上できることを確認した．

今後は，さまざまな環境における最適なサンプル数を決定する手法について検討する予定である．

謝辞　本研究の一部は，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」，および文部科学省特定領域研究 (16016260)，基盤研究 (A)(17200006)，基盤研究 (B)(2)(15300033) 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] 北島信哉，寺田努，原隆浩，西尾章治郎，“放送型データベースシステムにおけるデッドラインを考慮した問合せ処理方式，” 電子情報通信学会和文論文誌 D（2006 年，掲載予定）.