

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

加下 雅一[†] 寺田 努[‡] 塚本 昌彦[†] 原 隆浩[†] 西尾 章治郎[†]

[†] 大阪大学大学院工学研究科 情報システム工学専攻

[‡] 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

[†]{kashita,hara,tuka,nishio}@ise.eng.osaka-u.ac.jp

[‡]tsutomu@cmc.osaka-u.ac.jp

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、問合せに関係するテーブルをクライアントが蓄積してローカルに処理を行う方式と、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が考えられる。しかし、前者ではクライアントのディスクサイズが小さい場合、後者では問合せが頻繁に発生する場合に、クライアントが問合せ結果を受け取ることができない。

そこで本稿では、これらの問題点を解決するため、放送型データベースシステムにおいて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を提案する。提案方式では、クライアントからの問合せを受けたサーバが問合せ結果の作成に使用されるタプルに識別子を付加し、クライアントが問合せ結果に必要なタプルだけを蓄積できるようにする。また、サーバがデータの受信方法を指示するルールを作成してクライアントに送信することで、クライアント側において自動的に問合せ結果が作成される。本方式を用いることで、移動型クライアントが問合せ結果を効率的に受け取れる。

A Query Processing Method For Mobile Client on a Database Broadcasting System

Masakazu KASHITA[†] Tsutomu TERADA[‡] Masahiko TSUKAMOTO[†]

Takahiro HARA[†] Shojiro NISHIO[†]

[†]Dept. of Information Systems Engineering, Graduate School of Engineering, Osaka University

[‡]Cybercommunity Division, Cybermedia Center, Osaka University

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 PDAs. There are two methods to process queries in the broadcast database system; one is that a client stores in his/her disk all data that are necessary in producing the result and then processes it locally, and the other is that a server processes a query and broadcasts the query result to a client. However a client cannot properly obtain query results when the disk space of the client is small in the former method or when queries are issued frequently in the latter method.

In this paper, to resolve this problem, we propose a method where a server and a client process a query in collaboration in the broadcast database system. In this method, a server adds identifiers to tuples appearing in the query result when the server receives a query from a client. By referring to the identifiers, the client can store only necessary tuples. In addition, the server creates rules to define the behavior of the client and broadcasts them. By using these rules, the query results are automatically issued on the client. As a result, clients can obtain query results efficiently in this method.

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。放送型の情報システムでは、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さず情報配信ができ、ま

た、スループットの向上が期待できる。

放送型情報システムの応用例としては、次のようなものが考えられる。

- 遊園地で、園内の乗り物やレストラン、イベントに関する情報が放送されている。クライアントは、携帯端末を持ち歩きながらそれらの情報を見て、乗り物、レストランの情報や、その日行われるイベントの情報を得る。
- ショッピングセンターで各店舗の広告が放送されている。クライアントは携帯端末を持ち歩きながらそれらの情報を見て気に入った店に行く。

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に関する研究 [1, 7]、クライアント側のキャッシュ戦略に関する研究 [1]、データ更新の反映 [2]、プッシュ型とプル型の融合戦略に関する研究 [3, 4] 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において柔軟なサービスを提供するためには、データ内容に適したサービスやデータ処理機構が必要となる。そこで本研究では、放送型情報システムを用いたサービスとして、リレーショナルデータベースの内容を放送し、クライアントとしてはPDAなどの移動型クライアントを用いる環境を想定する。また、多くの放送型情報システムでは、一般にユーザは放送されるデータを受信して利用することのみが可能であるため、遊園地でユーザが待ち時間が10分以内の乗り物を検索するなどといった複雑なデータ選択をともなうサービスを提供するには、ユーザが端末上で問合せを行い自分の欲しいデータを能動的に取得できる機構が必要となる。

放送されるデータベースに対して問合せを行う場合、一般に以下の2つの方式が考えられる。

- **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一度クライアントのディスクに蓄え、処理に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。
- **オンデマンド型方式**：クライアントが上り帯域を利用して問合せをサーバに送信し、受け取ったサーバが問合せを処理し、データベースの放送に用いているチャンネルとは別のチャンネルを用いて、問合せ結果をクライアントに配信する。

クライアント型方式では、クライアントの記憶領域が小さい場合には問合せに関係するすべてのテ

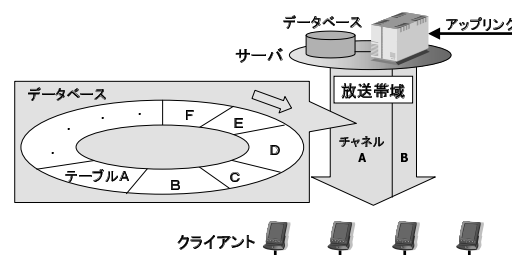


図 1: 放送型データベースシステムの概要

ブルを蓄積することができず、また、オンデマンド型方式では、問合せが頻繁に発生する場合にはサーバによる問合せ結果の放送が間に合わず、応答時間が非常に大きくなるという問題がある。

そこで本稿では、データベースの内容を放送する放送型情報システムにおいて、クライアント型とオンデマンド型方式で生じる問題を解決する処理方式である協調型方式を提案する。協調型方式では、サーバが問合せ結果作成に必要なデータに識別子を付加することで、必要なデータのみをクライアントに蓄積し、消費ディスク容量を低減する。また、サーバはデータの受信方法を指示する処理ルールをクライアントに放送することで、放送帯域を有効に利用できスループットの改善が期待できる。

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、さらに放送型情報システム上での一般的な問合せ処理方式について述べる。3章では本稿で提案する協調型方式について説明し、4章では協調型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における問合せ処理

本章ではまず、放送型データベースシステムについて説明し、その後、放送型データベースシステムにおける一般的な問合せ処理方式であるクライアント型方式とオンデマンド型方式について説明する。

2.1 放送型データベースシステム

本研究では、放送型情報システムにおいてデータベース内容を放送し、ユーザがデータベースに対して問合せできるような環境を想定する。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び、その概念は図1のように表せる。放送型データベースシステムは以下の特徴をもつ。

1. **放送する内容**：サーバは、データベースの内容を周期的に放送する。
2. **クライアント**：放送を受信するクライアントとして、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。また、クライアント端末は無線通信機能を備えているとする。

3. **複数チャンネル**: 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とする。図1に示すように、広帯域のチャンネルAを用いて、データベースの内容を繰り返し放送し、狭帯域のチャンネルBを用いてそれ以外のデータを放送する。
4. **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。

2.2 クライアント型方式

クライアント型方式とは、クライアントが問合せに関係するすべてのテーブルを一度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方式である。この場合、クライアント上ですべての問合せ処理が行われるため、サーバへのアップリンクは使用しない。以下、クライアント型方式における処理手順を説明する。

1. **問合せの発生**
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。
2. **データ受信**
クライアントは問合せ発生後、問合せに関係するテーブルの放送を監視し、問合せに関係するすべてのテーブルをディスクに蓄積する。
3. **問合せ処理**
クライアントは、蓄積したテーブルに対して問合せ処理を行い、問合せ結果を得る。

クライアント型方式では、クライアントが問合せを発生させシステムで問合せ処理を行うため、クライアント数が増加してもクライアントは放送周期1周期以内に問合せ結果に必要なすべてのデータを蓄積できる。しかし、クライアント型方式には以下のような問題点がある。

1. **クライアントのディスク領域を圧迫する。**
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ディスクに蓄えて処理する必要があるため、問合せ結果作成に不要なデータまで蓄積してしまい、クライアントのディスク容量を圧迫することになる。無駄なデータを蓄積する例を図2に示す。乗り物テーブル、待ち時間テーブルを含むデータベースが放送されていて、そのデータベースに対して図2に示すSQLによって問合せを行う場合、クライアントはまずSQL中に含まれるテーブルをシステムのディスクに蓄積し問合せ処理を行う。この問合せにおいて、実際に問合せ結果作成に使用されたタッ



図 2: 無駄なタプルや属性を蓄積する例

プルは斜線の入ったタプルであり、さらに使用された属性は、属性乗り物名、待ち時間だけである。このように、問合せ結果作成に直接使用されないタプルや属性のデータが蓄積され、ディスク容量を圧迫することになる。

2. クライアントに大きな負荷がかかる。
SQL処理は一般に負荷の高い処理となるため、クライアントの処理能力が低い場合、問合せ処理に処理能力のほとんどを奪われてしまう可能性がある。また、処理コストは問合せの複雑さによって大きく変化するため、必要なスペックをあらかじめ予想することが難しい。
3. 常に放送を監視する必要がある。
クライアントは、問合せ結果作成に必要なデータがいつ放送されるのかわからないため、必要なデータがすべて受信されるまで絶えず放送を受信、監視しておく必要がある。

2.3 オンデマンド型方式

オンデマンド型方式とは、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、サーバが問合せを処理して、結果を放送を用いてクライアントに送信する方式である。以下、オンデマンド型方式の処理手順を示す。

1. **問合せの発生と問合せの送信**
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利用してサーバに送信される。
2. **問合せ処理**
問合せを受け取ったサーバは、データベースに対して実際に問合せ処理を行い、問合せ結果をデータベースの放送に用いているチャンネルとは別のチャンネルを利用して放送する。

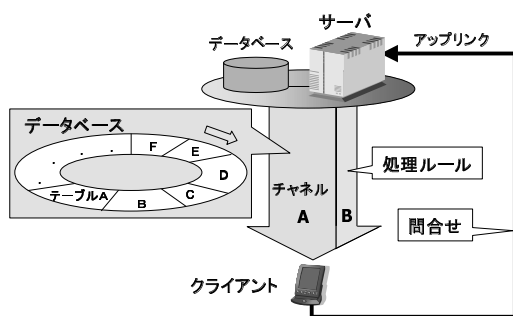


図 3: 協調型方式の概要

3. 問合せ結果受信

クライアントは、自分宛に放送される情報を受信し、問合せ結果を得る。

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、問合せ結果を作成するための作業領域を必要としない。また、サーバに送信される問合せが少ない場合、クライアントは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結果のサイズが大きい場合に、結果放送用の帯域を使いきってしまうため、クライアントが問合せ結果を受け取るまでの時間が長くなってしまふ可能性がある。

このように、クライアント型方式、オンデマンド型方式ともに、状況によっては問合せを処理できない可能性があるため、両方式の問題点を解決し、環境の変化に対しても効率的な処理が可能な問合せ処理方式が必要となる。

3 協調型方式

2章で述べたように、クライアント型方式とオンデマンド型方式では、クライアントが問合せ結果を受け取れない可能性がある。そこで本稿では、放送型データベースシステムにおいて効率的に問合せを処理する方式である協調型方式を提案する。以下、協調型方式の概要を説明し、本方式の処理手順について述べる。

3.1 方式概要

協調型方式とは、図 3 に示すように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で、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応答時間を低減する問合せ処理方式である。協調型方式の特徴を以下に示す。

- **サーバによる SQL 処理の補助**: 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、あ

らかじめサーバ上で SQL の結果作成に必要なタプルに処理用の識別子を付加する。クライアントは識別子によって受信データ中の必要なデータのみ蓄積するため、クライアントのディスク使用量を低減できる。

- **ルールによる受信データ処理の指定**: サーバは、タプルに識別子を付加した後、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、データベースの放送に用いているチャンネルとは別のチャンネルを用いて放送する。クライアントが自分宛に放送されるルールを格納すると、ルールが自動的に問合せ結果作成に必要なタプルを蓄積し問合せ結果を作成する。一般にルールのサイズは非常に小さいため、結果データをそのまま放送するのに比べて短時間に多数のクライアントの要求を処理できる。ルールの書式には、アクティブデータベースの動作記述方式である ECA ルールを用いる。

3.2 アクティブデータベース

協調型方式では、クライアントのデータ受信処理方法を記述するために、アクティブデータベースの動作記述方式である ECA ルールを用いている。アクティブデータベースとは、従来のデータベースとは異なりデータベース内部で起こる事象を監視し、あらかじめ定義された条件に適応する事象の発生に反応して、自動的に更新等の操作を行うデータベースシステムである [5]。ECA ルールとは、発生する事象（イベント）、実行させるための条件（コンディション）、イベントの発生によって実行される操作（アクション）の 3 つを 1 組とした動作記述方式である。協調型方式におけるクライアントの動作記述に ECA ルールを用いることで、以下のような利点が生まれる。

- イベント駆動型であるため、データの到着をイベントとして検出する機構を用意しておけば、到着データに対する要求をルールに書いておくだけで、必要な動作が自動的に実行される。問合せ処理はこのような要求の集合として表現されることになる。
- 処理はすべて ECA ルールの組で表現されるため、ECA ルールを変更、追加、削除することで機能のカスタマイズが可能である。

本研究と同様に、放送型システムにおいてアクティブデータベースの概念を用いている例としては、SADB (Super Active Database System) [6] が挙げられる。SADB では、放送型システムにおいて、クライアントが放送されるデータから必要な情報を効率的に抽出し、格納、再利用できるシステムにアクティブデータベースを用いている。SADB で

表 1: 使用できるイベント

名称	内容
SELECT	テーブルに対するデータ参照
INSERT	テーブルに対するタプル挿入
DELETE	テーブルのタプル削除
UPDATE	テーブルのタプル更新
RECEIVE	データの到着
TIMER	設定したタイマの発火

表 2: 使用できるアクション

名称	内容
QUERY([クエリー内容])	データベース操作
ENABLE_ECA([ルール識別子])	ECA ルールの有効化
DISABLE_ECA([ルール識別子])	ECA ルールの無効化
INSERT_ECA([ルール内容])	ECA ルールの格納
DELETE_ECA([ルール識別子])	ECA ルールの削除
SET_TIMER([タイマ識別条件], [時間])	新たなタイマの設定
KILL_TIMER([タイマ識別子])	タイマの削除
STORE([問合せ ID],[テーブル名], [テーブル名], [属性], ...)	タプルの格納
MATCH([問合せ ID],[テーブル名], [比較対象テーブル名],[属性], ...)	格納済みのタプルと組み合わせで格納
DISPLAY([問合せ ID])	問合せ結果の表示

は、アプリケーションのすべての機能を ECA ルールで表現しているのに対し、本研究では問合せ処理の効率化に特化して ECA ルール記述の拡張を行っている。本研究で用いる ECA ルールで使用可能なイベントおよびアクションを表 1、表 2 に示す。また、到着したデータの内容を用いて処理を行うルールでは、イベント対象となったタプル情報等が必要になるため、NEW データ、OLD データと呼ぶシステム変数を用意する。イベント発生時にこれらの変数に必要な情報が自動的に格納され、ルール中で自由に使用できる。各イベントに対する NEW データ、OLD データの内容を表 3 に示す。

3.3 問合せ処理アルゴリズム

本方式の問合せ処理手順は以下のとおりである。

1. **問合せの発生と問合せの送信**
クライアント上で問合せが発生すると、その問合せはアップリンクを利用してサーバに送信される。
2. **属性 S_ID, 属性 P_ID への書き込み**
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 ID) を決定する。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 S_ID, P_ID が用意されている。サーバは受信した SQL を解析し、問合せ結果に使用されるタプルを調べ、そのタプルの属性 S_ID に問合せ ID を書き込む。クライアントはこの属性を見ることで、問合せ結果作成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作成される場合、問合せ結果タプルを構成するそれぞれのタプルの属性 P_ID

表 3: NEW データと OLD データの内容

イベント	NEW	OLD
SELECT	参照タプル	-
INSERT	挿入タプル	-
DELETE	-	削除タプル
UPDATE	更新後タプル	更新前タプル
RECEIVE	到着パケット内容	-
TIMER	タイマ識別子	-

SELECT	乗り物, 乗り物名, 待ち時間, 待ち時間
FROM	乗り物, 待ち時間
WHERE	乗り物, 乗り物ID = 待ち時間, 乗り物ID
AND	待ち時間, 待ち時間 <= 15

図 4: SQL 例

に同じ値を書き込む。クライアントは、P_ID が等しいタプルを結合することで問合せ結果を作成する。属性 S_ID および P_ID は、複数の識別子を書き込めるようにある程度の大きさの領域が確保されており、1つのタプルが複数の問合せに使用される場合には、S_ID, P_ID に複数の値が並べて書き込まれる。

3. ECA ルール作成

サーバは、クライアントが問合せ結果作成に必要なタプルを受信し、問合せ結果を作成するための ECA ルールを作成する。ECA ルールは、問合せに関係するテーブル数、集約関数の有無など、問合せの種類ごとに用意されたテンプレートに、必要な値を代入し作成される。サーバは、問合せ結果作成に使用されるタプルが放送される時間を調べ、その時間だけ放送を監視する ECA ルールを作成するため、クライアントの電力消費を低減できる。

4. ECA ルール放送

サーバは、作成した ECA ルールをデータベースの放送に用いるチャンネルとは別のチャンネルを用いて放送する。クライアントは、自分宛の ECA ルールを格納する。

5. 問合せ結果表示

格納した ECA ルールにより自動的に必要なタプルの受信、タプルの組み合わせが行われ、問合せ結果が作成、表示される。

3.4 問合せ例

実際に本方式を用いた問合せ例として、クライアントが、図 4 に示す SQL によって問合せを行う場合の処理手順を、図 5 から図 7 に示す。

クライアントは、問合せを上り帯域を利用して送信する。サーバは、受信した SQL に対して問合せ ID を決定する。ここでは仮に 3 とする。次に、図 5 に示すように、問合せ結果に使用されるタプルを調べ、それぞれのタプルの属性 S_ID に問合せ ID を書き込む。そして、図 6 に示すように、結果

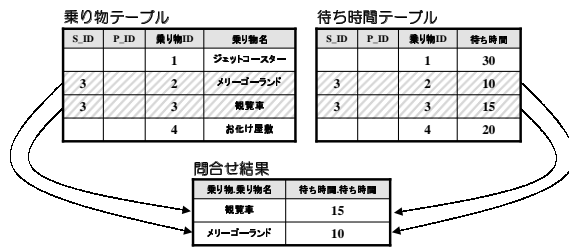


図 5: 属性 S_ID への書き込み

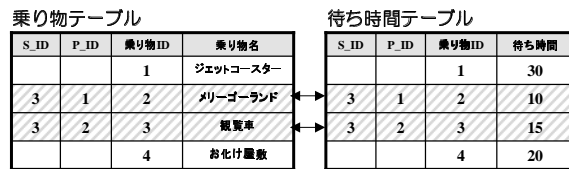


図 6: 属性 P_ID への書き込み

テーブルを構成するタプルがそれぞれどのタプルと結びつくのかを調べ、結びつくタプルの属性 P_ID に同じ値を書き込む。

最後に ECA ルールを作成する。今回の例で作成されるルール群を、図 7 に示す。'Rule3-1' は、問合せ結果のテーブルを作成するためのルールであり、'Rule3-2'、'Rule3-3' は、放送データ受信と問合せ結果表示のタイミングを計るためのルール、'Rule3-4'、'Rule3-5'、'Rule3-6'、'Rule3-7' は、データを受信し蓄積するためのルールである。'Rule3-8'、'Rule3-9' は結果を表示し、その後すべてのルールおよびタイマを削除するルールである。これらのルールが連鎖的に発火し、問合せ結果を作成、表示する。

4 評価

本章では、以下に示す 2 つの評価基準について、クライアント型方式、オンデマンド型方式と協調型方式を比較し、協調型方式の有効性を検証する。

1. クライアントのディスク使用量
問合せ結果を作成する段階で、クライアントが必要とする作業領域（ディスク使用量）を比較する。ディスク使用量には、問合せ結果サイズは含まず、純粋に作業のために必要となったディスク容量だけを比較する。
2. 応答時間
特定の頻度で問合せを発生させ、問合せ発生から問合せ結果を受け取るまでの時間を比較する。

4.1 評価のパラメタ

テーブルサイズ、タプルサイズ、タプル数などのパラメタは、想定するアプリケーションに強く

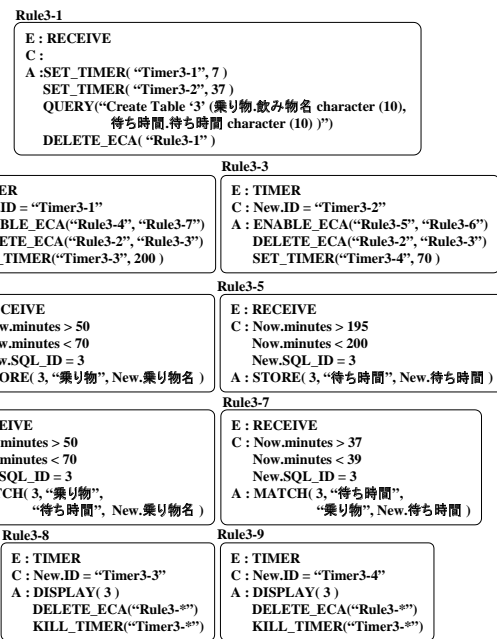


図 7: 作成される ECA ルール

影響を受け、一般的な乱数で与えることは難しいため、本稿では定数を用いた。使用するデータベースは、タプルサイズを 10K バイトとし、タプル数 1000 個のテーブルを 200 個含むとした。

協調型方式においては、各タプルには 1000 組の識別子を付加できるように識別子領域を確保した。また、データベースの放送に用いるチャンネルは 100Mbps、問合せ結果や ECA ルールを放送する帯域は 5Mbps とした。この場合、放送周期は、クライアント型方式では 160 秒、協調型方式では識別子領域の分だけ放送量が増えるため 183 秒となる。

4.2 ディスク使用量の比較

本節では、3 つの方式において、問合せ結果を作成する時にクライアントが必要とするディスク使用量を比較する。3 方式のディスク使用量 D は、解析により以下のように求められる。ここで、 S はテーブルの大きさ、 T は問合せに関するテーブル数、 R はデータ利用率を表す。ここでデータ利用率とは、問合せに関するテーブルのうち、実際に問合せ結果作成に使用されたデータの割合である。

- クライアント型方式： $D = ST$
- 協調型方式： $D = STR$
- オンデマンド型方式： $D = 0$

クライアント型方式では、クライアントは問合せに関するすべてのテーブルをディスクに蓄積する必要があるため、ディスク使用量はテーブルサイズと問合せに関するテーブル数の積で表される。協

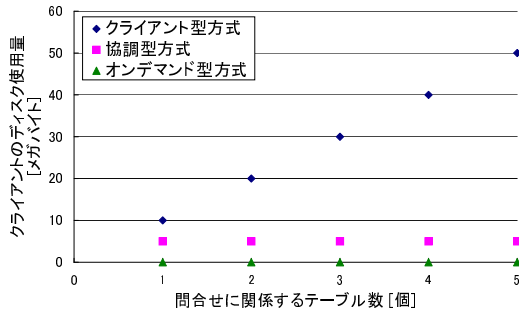


図 8: 問合せに関するテーブル数に対するディスク使用量の変化

調型方式では、テーブルサイズと問合せに関するテーブル数の積に、さらにデータ利用率を掛け合わせることでディスク使用量が得られる。オンデマンド型方式では、問合せ結果がそのまま放送されるため、問合せ結果の作成にクライアントのディスクが使用されることはない。

上記の式を用いて、問合せに関するテーブル数 T とデータ利用率 R をパラメタとしてクライアントのディスク使用量 D を解析した。

4.2.1 テーブル数をパラメタとした場合

問合せ結果サイズを 5M バイトとし、問合せに関するテーブル数 T を 1 から 5 個まで変化させた場合のクライアントのディスク使用量 D を解析した。結果を図 8 に示す。

T が変化しても、オンデマンド型方式では D は 0 であり、協調型方式では 5M バイトで一定であるが、クライアント型方式は右上がりの直線となる。 T が増加しても、協調型方式はクライアント型方式に比べ、常に少ないディスク使用量で問合せを処理できる。一般的に T の値が高々 3 であると仮定した場合、協調型方式における D の値は最大で 3M バイトとなるが、クライアント型方式における D は最大で 30M バイトとなる。このため、クライアント端末が PDA (Personal Digital Assistant) のように 10M バイト程度しか記憶領域をもたない端末の場合、クライアント型方式では問合せ処理に必要なすべてのデータを蓄積できない場合があるが、協調型方式では T が大きい場合でも処理できる。

4.2.2 データ利用率をパラメタとした場合

問合せに関するテーブル数 T を 2 とおき、データ利用率 R を変化させた場合のクライアントのディスク使用量 D を解析した。結果を図 9 に示す。

クライアント型方式では、問合せに関するすべてのテーブルをディスクに蓄えるため、 R が変化しても D は常に一定となる。オンデマンド型方式では D は 0 となり、協調型方式では右上がりの直線となる。 R が小さい問合せが発生する場合には、

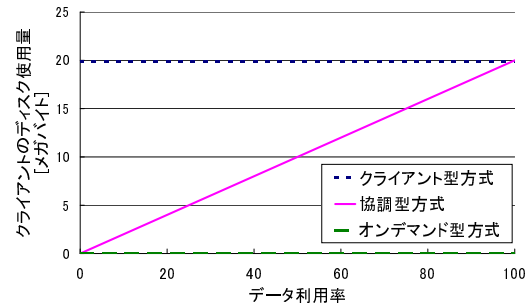


図 9: データ利用率に対するディスク使用量の変化

協調型方式がクライアント型方式に比べ D を大きく低減できることがわかる。また、 R が最悪の値であっても協調型方式はクライアント型方式に比べ D を低減できる。クライアント型方式における D は常に 20M バイトとなるが、一般的な環境での R の値が高々 10% 程度であると仮定した場合、協調型方式の D は 2M バイトとなる。クライアント端末が、PDA のように 10M バイト程しか記憶領域をもたない端末の場合、クライアント型方式では、問合せ処理に必要なすべてのデータを蓄積できない場合があるが、協調型方式では R が 50% まで蓄積できる。

4.3 応答時間の比較

本節では応答時間の評価を行う。応答時間は、具体的にクライアント型方式ではクライアントが問合せを発生させてから問合せに関するすべてのテーブルを蓄積するまでの時間、協調型方式では問合せ結果の作成に必要なすべてのタプルを蓄積するまでの時間、オンデマンド型方式では問合せ結果を受け取るまでの時間となる。

3 方式における応答時間 T は、以下のように表される。ここで、問合せに関するテーブル数を N 、クライアント型方式の放送周期を P_{cli} 秒、協調型方式の放送周期を P_{col} 秒、協調型方式で 1 つのタプルに付加可能な識別子の数を ID 、問合せ結果サイズを S メガバイト、問合せ結果や ECA ルールを放送する帯域を W メガ bps、問合せ発生間隔を d ミリ秒とおく。問合せ発生間隔とはサーバに問合せが到着する間隔を表し、本評価では一定の間隔で問合せが発生するとした。

- クライアント型方式

$$T = P_{cli} \times \left(1 - \frac{1}{N+1}\right)$$

- 協調型方式

$$T = \begin{cases} \infty & \left(0 < d < \frac{1000P_{col} \times \left(1 - \frac{1}{N+1}\right)}{ID}\right) \\ P_{col} \times \left(1 - \frac{1}{N+1}\right) & \left(\frac{1000P_{col} \times \left(1 - \frac{1}{N+1}\right)}{ID} \leq d\right) \end{cases}$$

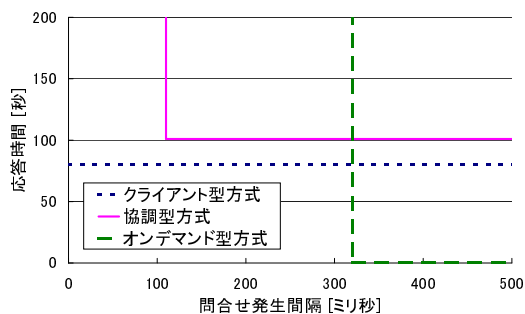


図 10: 問合せ発生間隔をパラメタにした比較

- オンデマンド型方式

$$T = \begin{cases} \infty & (0 < d < \frac{8000S}{W}) \\ \frac{8S}{W} & (\frac{8000S}{W} \leq d) \end{cases}$$

協調型方式やオンデマンド型方式では、問合せ発生間隔 d が小さい場合には応答時間が非常に大きくなるため、式中で ∞ と表している。

4.1節より、 $P_{cli} = 160$, $P_{col} = 202$, $W = 5$ とし、 $N = 1$, $ID = 1000$ とする。以下では、問合せ発生間隔 d をパラメタとする応答時間の評価を行う。 S を 200K バイトとし d を変化させた時の T の変化を調べた。

結果を図 10 に示す。クライアント型方式では、問合せ発生間隔 d が変化しても応答時間 T は変化しない。それに対して、協調型方式では、 d が大きい場合には T は変化せず、 d が 110 ミリ秒以下では T が発散する。これは、1 つのタプルには 1000 個の識別子を付加できるが、問合せが頻繁に発生することにより、識別子の数が不足し識別子が解放されるまで待たなければならないためである。また、オンデマンド型方式では、 d が大きい場合には T は非常に小さいが、 d が 320 ミリ秒以下では T が発散する。これは、問合せが頻繁に発生することで、サーバが問合せ結果を放送し終える前に新たな問合せが発生してしまうためである。よって、協調型方式では d が 110 ミリ秒以下の場合、オンデマンド型方式では 320 ミリ秒以下の場合に、クライアントが問合せ結果を受け取れないことになる。

本稿における想定環境では、 d は 180 ミリ秒程度としている。 d が 180 ミリ秒の場合には、オンデマンド型方式では応答時間が発散してしまうためクライアントは問合せ結果を受け取れないが、協調型方式では d が 110 ミリ秒まで受け取れる。

5 おわりに

本稿では、データベースの内容を放送する放送型情報システムにおける効率的な問合せ処理方式として協調型方式を提案した。協調型方式では、サーバが問合せに使用されるタプルに識別子を付加し、

データの受信方法を ECA ルールによってクライアントに指示することで、クライアントが問合せ結果の作成に必要なタプルだけを蓄積できる。また、本方式を解析的に評価し、本方式を用いることでクライアント型方式に比べクライアントのディスク使用量を低減でき、オンデマンド型方式に比べ応答時間を低減できることがわかった。

今後の課題として、より詳細な評価および本方式を実装して実環境における実測評価を行うことが挙げられる。また、問合せ発生間隔や問合せ結果サイズなどのパラメタの変化に応じて、クライアント型方式、協調型方式、オンデマンド型方式を使い分ける自動選択方式を実現する。3 方式を動的に選択し、その時点で最も良い方式を選択することで、どのような環境においても効率的な問合せ処理が実現できると考えられる。

謝辞 本研究は、日本学術振興会基盤研究 (B)(2) (12480095) および日本学術振興会未来開拓学術研究推進事業における研究プロジェクト「マルチメディア・コンテンツの高次処理の研究」(Project No. JSPS-RFTF97P00501) 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," in *Proc. ACM SIGMOD Conference*, pp. 199–210 (1995).
- [2] S. Acharya, M. Franklin, and S. Zdonik: "Disseminating Updates on Broadcast Disks," in *Proc. VLDB Conference*, pp. 354–365 (1996).
- [3] S. Acharya, M. Franklin, and S. Zdonik: "Balancing Push and Pull for Data Broadcast," in *Proc. ACM SIGMOD Conference*, pp. 183–194 (1997).
- [4] 箱守聰, 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," 情処ワークショップ論文集, vol. 97, no. 2, pp. 55–60 (1997).
- [5] G. Lohman, L. Bruce, P. Hamlin, and K. Bernhard, "Extensions to starburst: Object, types, functions, and rules," in *Communications of the ACM*, vol. 34, no. 10, pp. 94–109 (1991).
- [6] 寺田努, 塚本昌彦, 西尾章治郎: "放送型データ受信のためのアクティブデータベースシステム的设计と実装," 電子情報通信学会論文誌, vol. J83-D-I, no. 12, pp. 1272–1283 (2000).
- [7] E. Yajima, T. Hara, M. Tsukamoto, and S. Nishio: "Scheduling and caching strategies for broadcasting correlated data," in *Proc. ACM Symposium on Applied Computing (ACM SAC 2001)*, pp. 504–509 (Mar. 2001).