

放送型データベースシステムにおける モバイル端末の電力消費を考慮した問合せ処理について

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

^{††} 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: †{kitajima.shinya,hara,nishio}@ist.osaka-u.ac.jp, ††tsutomu@cmc.osaka-u.ac.jp

あらまし 近年、サーバが携帯端末や PDA などのモバイル端末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。これらの方式は電力消費に差があり、また、モバイル端末では利用できる電力に限りがある。そこで本稿では、モバイル端末の電力消費を考慮し、3 方式の中から動的に処理方式を選択する手法を提案する。

キーワード データ放送、放送型データベース、問合せ処理、電力消費

On Query Processing Considering the Energy Consumption of Mobile Clients for Broadcast Database Systems

Shinya KITAJIMA[†], Tsutomu TERADA^{††}, Takahiro HARA[†], and Shojiro NISHIO[†]

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

^{††} Cybercommunity Division, Cybermedia Center, Osaka University
5-1 Mihogaoka, Ibaraki-shi, Osaka 567-0047, Japan

E-mail: †{kitajima.shinya,hara,nishio}@ist.osaka-u.ac.jp, ††tsutomu@cmc.osaka-u.ac.jp

Abstract 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 broadcast database system; (i) the server processes a query and then 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. Generally, mobile clients have limits in power consumption, i.e., battery, and each of the three methods consumes different amount of power for query processing. In this paper, we propose a new query processing method which dynamically chooses a query processing method among the three query processing methods considering power consumption.

Key words data broadcasting, broadcast database system, query processing, power consumption

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], [8], 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 [11] 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 [12] 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を提案した。さらに文献 [13] において、文献 [12] 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した。しかし、これら手法では問合せ処理にかかる消費電力を考慮していないため、利用できる電力に限られるモバイル環境では、電力の少ない端末の生存時間が短くなってしまいう問題があった。

そこで本稿では、文献 [13] における手法をもとに、モバイル端末の電力残量を考慮して問合せ処理方式を選択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、提案手法が従来手法と比べて、特に電力の少ない端末の生存時間を向上できることを示す。

以下、2. で放送型データベースシステムについて述べる。3. で提案手法について説明し、4. では提案手法の性能評価を行う。5. で関連研究を紹介し、最後に 6. 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、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

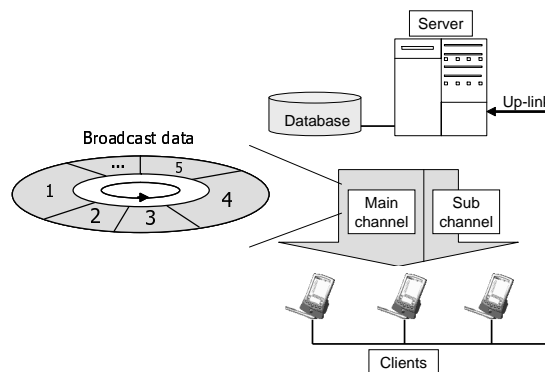


図 1 放送型データベースシステム

Fig. 1 Broadcast database system.

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、処理能力の乏しいモバイル端末を想定する。

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ンネルが存在する。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、ユーザはモバイル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、「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。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を得られない場合は、その問合せは失敗となる。ユーザは一定時間ショッピングセンターに滞在するが、端末の電力には限りがあり、端末の電力を使い果たしたユーザはサービスを受けられなくなる。

放送帯域は 10Mbps 程度とする。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、定期的

に放送データのインデックスを放送する。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。そのため、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せず、問合せ結果を得るために必要な電力も小さい。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、端末のディスク容量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、問合せ処理に大きな電力が必要にな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なタ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 [11]。

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、消費電力も小さくなる。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、問合せの成功率が下がってしまう。

2.3 LRT 方式

2.2 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 [12] 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。

2.4 拡張 LRT 方式

LRT 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。

拡張 LRT 方式 [13] では、LRT 方式において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、割り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。

3. ELEC 方式

2.4 で述べた拡張 LRT 方式では、問合せの成功率の向上と応答時間の低減を目的とし、問合せ処理にかかる消費電力を考慮していないため、電力の少ない端末の生存時間が短くなってしまおうという問題があった。そこで本章では、端末の電力残量を考慮して問合せ処理方式を選択し、電力の少ない端末の生存時間を向上する手法を提案する。

3.1 概要

ELEC (Extended LRT considering Energy Consumption)

方式では、サーバは拡張 LRT 方式に従って問合せ処理方式を選択するが、問合せを発行した端末の電力残量が閾値 TH_{on} 以下ならば、電力残量が少ないと判断し、消費電力の少ないオンデマンド型方式を優先的に選択する。一方、端末の電力残量ももう 1 つの閾値 TH_{cl} 以上ならば、電力残量が多いと判断し、消費電力の多いクライアント型方式を優先的に選択する。ここで、 $0 \leq TH_{on} \leq TH_{cl} \leq 1$ である。

x 個前にサーバに到着した問合せ Q_x を発行した端末の電力残量を P_x 、過去 q 個の問合せを発行した端末の電力残量の最大値を $P_{max}(q)$ ($P_{max}(q) = \max(P_1, P_2, \dots, P_q)$)、新たに問合せを発行した端末の電力残量を P_{new} とすると、問合せ処理方式選択の流れは以下ようになる。

(1) 拡張 LRT 方式に従って問合せ処理方式を選択

(2) $P_{new} < P_{max}(q) \times TH_{on}$ 、かつ、オンデマンド型方式が使用できるならばオンデマンド型方式を選択

(3) $P_{new} > P_{max}(q) \times TH_{cl}$ 、かつ、クライアント型方式が使用できるならばクライアント型方式を選択

3.2 閾値の計算アルゴリズム

最適な TH_{on} 、 TH_{cl} はシステム状況によって異なるため、状況に応じて動的に変化させる。計算の際には、問合せを発行した端末の電力残量が閾値 TH_{on} 以下の問合せのうち、実際にオンデマンド型方式を選択した問合せの割合を、 Q_i ($i = 1, 2, \dots, q$) と、 Q_j ($j = q + 1, q + 2, \dots, 2q$) について比較する。これは、オンデマンド型方式を選択してもデッドラインを越えない端末の割合が、システム状況と密接に関係しているためである。

Q_i ($i = 1, 2, \dots, q$) のうち、 $P_i < P_{max}(q) \times TH_{on}$ を満たす問合せ数を L_{max} 個、そのうちオンデマンド型方式が選択された問合せ数を L_{on} 個とする。また、 Q_j ($j = q + 1, q + 2, \dots, 2q$) に対して、問合せを発行した端末の電力残量の最大値を $P_{max}^{old}(q)$ ($P_{max}^{old}(q) = \max(P_{q+1}, P_{q+2}, \dots, P_{2q})$)、 $x = q + 1, q + 2, \dots, 2q$ における TH_{on} を TH_{on}^{old} 、 TH_{cl} を TH_{cl}^{old} 、 $P_j < P_{max}^{old}(q) \times TH_{on}^{old}$ を満たす問合せ数を L_{max}^{old} 個、そのうちオンデマンド型方式が選択された問合せ数を L_{on}^{old} 個とする。

閾値の変化量は、緩めるときには W_{plus} 、強めるときには W_{minus} を用いることとし、オンデマンド型方式を選択できる問合せには限りがあることから TH_{on} を優先的に変更する。

閾値の計算アルゴリズムを以下に示す。サーバは以下の処理の 1 ステップを、問合せが q 個到着するごとに実行する。初期状態は (1) とし、 TH_{on} の初期値は 0、 TH_{cl} の初期値は 1 とする。

(1) $\frac{L_{on}}{L_{max}} \geq \frac{L_{on}^{old}}{L_{max}^{old}}$ である限りは、 $TH_{on} = TH_{on}^{old} + W_{plus}$ とする。条件を満たさなくなれば (2) に移る。

(2) $\frac{L_{cl}}{L_{max}} \geq \frac{L_{cl}^{old}}{L_{max}^{old}}$ である限りは、 $TH_{cl} = TH_{cl}^{old} - W_{plus}$ とする。条件を満たさなくなれば (3) に移る。

(3) $\frac{L_{on}}{L_{max}} < \frac{L_{on}^{old}}{L_{max}^{old}}$ である限りは、 $TH_{on} = TH_{on}^{old} - W_{minus}$ とする。条件を満たさなくなれば (4) に移る。

(4) $\frac{L_{cl}}{L_{max}} < \frac{L_{cl}^{old}}{L_{max}^{old}}$ である限りは、 $TH_{cl} = TH_{cl}^{old} +$

W_{minus} とする。条件を満たさなくなれば (1) に移る。

ただし、 $L_{max} = 0$ 、または $\frac{L_{on}}{L_{max}} = 1$ ならば、 TH_{on} に

W_{plus} を加えるとともに、 TH_{cl} から W_{plus} を引く。

また、 TH_{on} 、 TH_{cl} は、 $0 \leq TH_{on} \leq TH_{cl} \leq 1$ なる関係を満たす範囲でのみ変化させるものとし、この関係を満たさなくなる場合には、値の増減は行わない。

4. 評価

本章では、次に示す 3 つの評価基準を用いて、ELEC 方式の有効性をシミュレーション実験の結果から検証する。

- 問合せ成功率

発生した全問合せのうち、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

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、問合せ結果を得るまでの平均時間。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、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。

- 平均生存時間

クライアントが到着してから、予定滞在時間が経過した、もしくは端末の電力残量が 0 になったことにより退出するまでの平均時間。ただし、端末は問合せに関わる処理でのみ電力を消費するものとする。

4.1 評価モデル

本評価では、2.1 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プルサイズは等しいものとする。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結果を受け取れない場合は、問合せは失敗とする。また、端末のディスク容量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ディスク容量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。

サーバは、ジャンルごとのテーブルの放送時刻を示すインデックスを 1 周期ごとに放送する。インデックスのサイズは

データサイズと比べて非常に小さいため、無視できるものとする。また、クライアントからの問合せに対し、3方式のうちどの方式で処理するか、また、問合せが失敗であるかをクライアントに通知する選択方式通知メッセージをサブ放送帯域を用いて送信する。選択方式通知メッセージのサイズは無視できるほど小さいものとし、常にサブ放送帯域の放送キューの先頭に挿入されるものとする。

4.2 消費電力のモデル化

問合せ処理に必要な電力を評価するために、消費電力のモデル化を行う。問合せ処理に必要な電力とは、問合せを発生した後に、必要なテーブルの受信やタプルの結合処理に必要な電力を指し、問合せの送信、問合せ結果の表示に必要な電力は、方式による差がないため考慮しない。

4.2.1 基本となる消費電力

まず、CPU、無線LAN、I/Oに必要な電力をそれぞれ定義する。各消費電力の比例係数は、文献[14]においてMahesriらが行ったラップトップにおける測定結果を参考にした。また、すべての消費電力は時間に比例するものとする。

a) CPU

CPUにかかる電力 E_C は、CPU負荷が高い場合 E_C^h と低い場合の E_C^l で定義する。

$$E_C^h(t) = 2t$$

$$E_C^l(t) = 0.5t$$

b) 無線LAN

無線LANの受信にかかる電力 E_W は、以下のように定義する。

$$E_W(t) = t$$

c) I/O

I/Oにかかる電力は、読み込みにかかる電力 E_I 、書き込みにかかる電力 E_O で定義する。

$$E_I(t) = t$$

$$E_O(t) = t$$

問合せ処理にかかる全電力 E は、これらの和で表せるものとする。

$$E(t) = E_C(t) + E_W(t) + E_I(t) + E_O(t) \quad (1)$$

4.2.2 動作にともなう消費電力

4.2.1で定義した消費電力をもとに、放送受信、処理、書き込み、読み込みに必要な電力を定義する。

a) 放送受信

放送を受信しながら、受信したデータを書き込む際の消費電力 E_{rcv} を以下の式で定義する。

$$E_{rcv}(t) = E_C^l(t) + E_W(t) + E_O(t) = 2.5t \quad (2)$$

b) 処理

タプル結合などの処理にかかる電力 E_{prc} は、処理に必要な時間 t_{prc} を用いて以下のように定義する。 t_{prc} は、端末の処理能力に応じて異なる。

$$E_{prc}(t_{prc}) = E_C^h(t_{prc}) = 2t_{prc} \quad (3)$$

c) 書き込み

放送を受信しながらの書き込みに必要な電力は式(2)で表す。その他のデータ書き込みに必要な電力 E_{wrt} は、書き込みにかかる時間 t_{wrt} を用いて以下のように定義する。

$$E_{wrt}(t_{wrt}) = E_C^l(t_{wrt}) + E_O(t_{wrt}) = 1.5t_{wrt} \quad (4)$$

ここで t_{wrt} は、書き込み速度を SPD_{wrt} 、書き込みデータサイズを S_{wrt} とすると、以下の式で表せる。

$$t_{wrt} = \frac{S_{wrt}}{SPD_{wrt}} \quad (5)$$

d) 読み込み

データを処理する前に読み込む際にかかる電力 E_{rd} は、読み込みにかかる時間 t_{rd} を用いて以下の式で定義する。

$$E_{rd}(t_{rd}) = E_C^l(t_{rd}) + E_I(t_{rd}) = 1.5t_{rd} \quad (6)$$

ここで t_{rd} は、読み込み速度を SPD_{rd} 、読み込みデータサイズを S_{rd} とすると、以下の式で表せる。

$$t_{rd} = \frac{S_{rd}}{SPD_{rd}} \quad (7)$$

4.2.3 問合せ処理に必要な電力

4.2.2で定義した消費電力をもとに、オンデマンド型方式、クライアント型方式、協調型方式の問合せ処理に必要な電力を定義する。

a) オンデマンド型方式

オンデマンド型方式では、問合せ結果をサブ放送帯域から受信するだけである。よって、オンデマンド型方式の問合せ処理にかかる電力 E_{on} は、問合せ結果の受信にかかる時間 t_{on} を用いて以下の式で表せる。

$$E_{on}(t_{on}) = E_{rcv}(t_{on}) = 2.5t_{on} \quad (8)$$

ここで t_{on} は、サブ放送帯域幅を WDT_{sub} 、問合せ結果サイズを S_{rst} とすると、以下の式で表せる。

$$t_{on} = \frac{S_{rst}}{WDT_{sub}} \quad (9)$$

b) クライアント型方式

クライアント型方式では、問合せ処理に必要なテーブルをメイン放送帯域から受信した後、問合せ処理を行う。よって、クライアント型方式の問合せ処理に必要な電力 E_{cl} は、必要テーブルの受信にかかる時間 t_{cl}^{rcv} 、必要テーブルの読み込みにかかる時間 t_{cl}^{rd} 、タプル結合処理にかかる時間 t_{cl}^{prc} 、問合せ結果の書き込みにかかる時間 t_{cl}^{wrt} を用いて以下の式で表せる。

$$E_{cl} = E_{rcv}(t_{cl}^{rcv}) + E_{rd}(t_{cl}^{rd}) + E_{prc}(t_{cl}^{prc}) + E_{wrt}(t_{cl}^{wrt}) \quad (10)$$

ここで、一ジャンル分のテーブルサイズを S_{gnr} 、メイン放送帯域幅を WDT_{main} とすると、 t_{cl}^{rcv} 、 t_{cl}^{rd} 、 t_{cl}^{wrt} はそれぞれ次のように表せる。

$$t_{cl}^{rcv} = \frac{S_{gnr}}{WDT_{main}} \quad (11)$$

$$t_{cl}^{rd} = \frac{S_{gnr}}{SPD_{rd}} \quad (12)$$

$$t_{cl}^{wrt} = \frac{S_{rst}}{SPD_{wrt}} \quad (13)$$

t_{cl}^{prc} は端末の処理能力によって異なる。

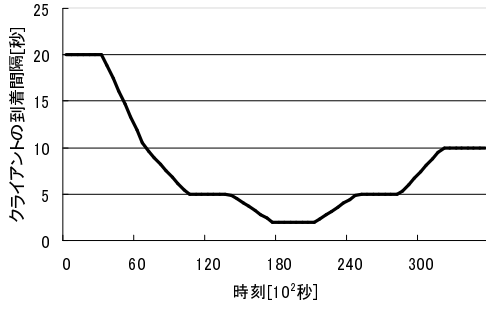


図 2 クライアントの到着間隔の変化

Fig. 2 Change in clients arrival frequency.

c) 協調型方式

協調型方式では、処理ルールを受信した後、処理ルールをもとに必要なタプルのみを受信し、タプル結合処理を行う。よって、クライアント型方式の問合せ処理に必要な電力 E_{co} は、処理ルールの受信にかかる時間 $t_{co}^{rcvrule}$ 、処理ルールの読み込みにかかる時間 t_{co}^{rdrule} 、必要タプルの受信にかかる時間 t_{co}^{rcvtpl} 、必要タプルの読み込みにかかる時間 t_{co}^{rdtpl} 、タプル結合処理にかかる時間 t_{co}^{prc} 、問合せ結果の書き込みにかかる時間 t_{co}^{wrt} を用いて以下の式で表せる。

$$E_{co} = E_{rcv}(t_{co}^{rcvrule}) + E_{rd}(t_{co}^{rdrule}) + E_{rcv}(t_{co}^{rcvtpl}) + E_{rd}(t_{co}^{rdtpl}) + E_{prc}(t_{co}^{prc}) + E_{wrt}(t_{co}^{wrt}) \quad (14)$$

ここで、処理ルールのサイズを S_{rule} 、必要なタプルのサイズを S_{tpl} とすると、 $t_{co}^{rcvrule}$ 、 t_{co}^{rdrule} 、 t_{co}^{rcvtpl} 、 t_{co}^{rdtpl} 、 t_{co}^{wrt} はそれぞれ次のように表せる。

$$t_{co}^{rcvrule} = \frac{S_{rule}}{WDT_{sub}} \quad (15)$$

$$t_{co}^{rdrule} = \frac{S_{rule}}{SPD_{rd}} \quad (16)$$

$$t_{co}^{rcvtpl} = \frac{S_{tpl}}{WDT_{main}} \quad (17)$$

$$t_{co}^{rdtpl} = \frac{S_{tpl}}{SPD_{rd}} \quad (18)$$

$$t_{co}^{wrt} = \frac{S_{rst}}{SPD_{wrt}} \quad (19)$$

t_{co}^{prc} は端末の処理能力によって異なる。

4.3 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 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。タプル利用率は、1 ジャンル分のテーブルの全タプルに対する、問合せ結果に含まれるタプルの割合を表す。クライアントは、指数分布に従った間隔でショッピングセンターに到着するものとし、到着間隔の平均は図 2 のように変化させた。各クライアントはサービスを受けられる端末を 1 台ずつ所持しており、各端末の初期電力量は 100 単位エネルギーから 1000 単位エネルギーの一様分布、各端末のディスク容量は 1MB から 100MB の一様分布とした。ショッピングセンターに到着したクライアントは、クエリ発生間隔に従って問合せを発行する。このとき、デッドラインや端末のディ

表 1 評価に用いるパラメータ

Table 1 Parameter configuration.

パラメータ名	値
シミュレーション時間 [秒]	36000
クライアントの予定滞在時間 [秒]	7200
1 クライアント当たりのクエリ発生間隔 [秒]	300
応答時間のデッドライン [秒]	80
ジャンル数 [個]	10
ジャンル内ショップ数 [個]	5
ショップ内商品数 [個]	200
1 タプルのサイズ [KByte]	10
1 タ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
タプル利用率の平均	0.003
タプル利用率の標準偏差	0.001
余裕時間の計算に用いるサンプル数	5

スク容量、電力残量のデータを、問合せとともにサーバに送信する。各クライアントはショッピングセンターに到着したのち、予定滞在時間が経過するとショッピングセンターを出る。簡単化のため、端末の電力残量が 0 になったクライアントはサービスを受けられなくなるため、ショッピングセンターから退出するものとする。

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。消費電力の計算では、簡単のためすべての端末の処理速度は同じであるとし、式 (10) における t_{cl}^{prc} を、測定実験の結果から 1.5 秒とした。測定に用いたのは TOSHIBA 社製 PocketPC e740 である。また、式 (14) における t_{co}^{prc} は、以下の式から決定した。

$$t_{co}^{prc} = t_{cl}^{prc} \times \frac{\text{タプル利用率}}{\text{ジャンル内ショップ数}} \quad (20)$$

データの書き込み、読み込み速度は、それぞれ 10MB/s、15MB/s とした。

4.4 シミュレーション結果

4.4.1 従来方式との比較

図 2 のようにクライアントの到着間隔を変化させたときの、拡張 LRT 方式と ELEC 方式における問合せ成功率、平均応答時間を表 2 に示す。また、端末の初期電力量をもとに 100 単位エネルギーごとに分類した平均生存時間を図 3 に示す。ELEC 方式における q は 50、 W_{plus} は 0.002、 W_{minus} は 0.005 とした。

図表から、ELEC 方式では拡張 LRT 方式と比べ、問合せ成功率、平均応答時間もわずかに性能が悪くなっているが、初期電力量の少ない端末の平均生存時間が大きく改善しているこ

表 2 問合せ成功率と平均応答時間
Table 2 Success rate and response time.

手法	問合せ成功率	平均応答時間
拡張 LRT 方式	94.3%	44.3 秒
ELEC 方式	92.9%	45.8 秒

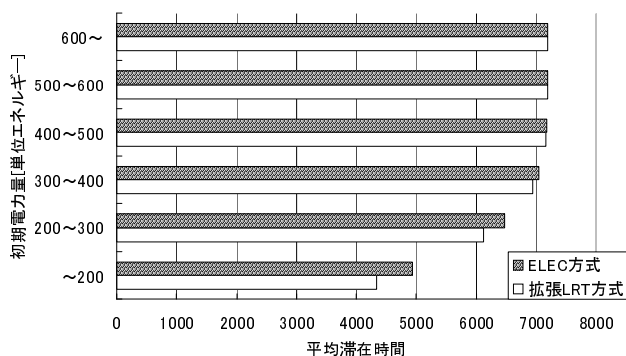


図 3 平均生存時間
Fig. 3 Average lifetime.

表 3 問合せ成功率と平均応答時間への q の影響
Table 3 Impact of q on success rate and response time.

q	問合せ成功率	平均応答時間
$q = 10$	92.8%	47.8 秒
$q = 30$	92.9%	46.5 秒
$q = 50$	92.9%	45.8 秒
$q = 70$	93.0%	45.5 秒
$q = 90$	93.3%	45.3 秒

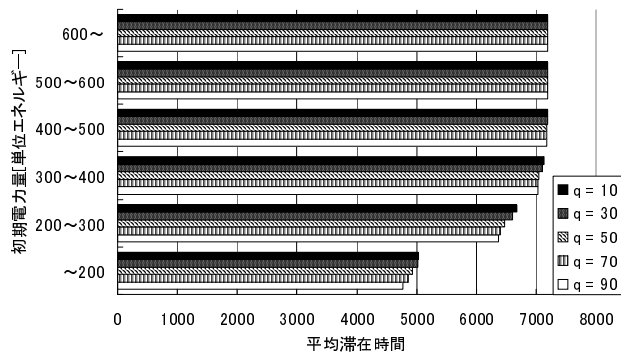


図 4 平均生存時間への q の影響
Fig. 4 Impact of q on average lifetime.

とがわかる．拡張 LRT 方式では，初期電力量が多い端末の大半は電力を使い果たすことなく予定滞在時間経過後に退出しているが，ELEC 方式では，電力残量の多い端末に優先してクライアント型方式を割り当てることにより，電力残量の少ない端末のオンデマンド型方式の選択率を上げている．これにより，初期電力量の少ない端末の平均生存時間を向上している．

4.4.2 q の影響

ELEC 方式において， q を 10, 30, 50, 70, 90 と変化させたときの問合せ成功率，平均応答時間を表 3 に示す．また，端末の初期電力量をもとに 100 単位エネルギーごとに分類した平

表 4 問合せ成功率と平均応答時間への W_{plus} , W_{minus} の影響
Table 4 Impact of W_{plus} and W_{minus} on success rate and response time.

(W_{plus}, W_{minus})	問合せ成功率	平均応答時間
$(W_{plus}, W_{minus}) = (0.001, 0.002)$	93.0%	47.1 秒
$(W_{plus}, W_{minus}) = (0.002, 0.001)$	90.8%	47.6 秒
$(W_{plus}, W_{minus}) = (0.002, 0.002)$	91.6%	47.7 秒
$(W_{plus}, W_{minus}) = (0.001, 0.005)$	93.2%	46.0 秒
$(W_{plus}, W_{minus}) = (0.05, 0.1)$	92.8%	47.8 秒

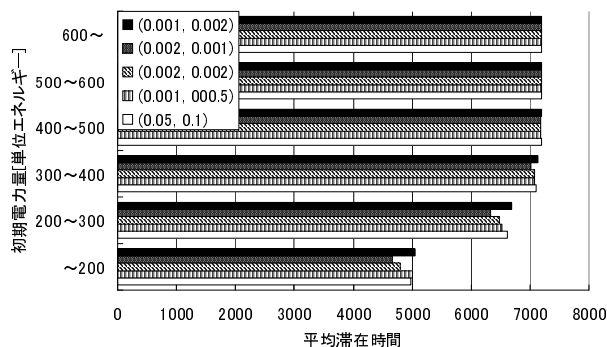


図 5 平均生存時間への W_{plus} , W_{minus} の影響
Fig. 5 Impact of W_{plus} and W_{minus} on average lifetime.

均生存時間を図 4 に示す．ELEC 方式における W_{plus} は 0.002, W_{minus} は 0.005 とした．

図表から， q の値が小さくなるにつれ，問合せ成功率と平均応答時間の性能が悪くなっていることがわかる．一方で，初期電力量の少ない端末の平均生存時間は長くなっている． q の値が小さいと，クライアントの到着間隔の変化に従って，閾値が頻繁に増減する．閾値を変化させる際に問合せ成功率や平均応答時間は考慮していないため，閾値が大きく変化するにつれて性能が悪くなったと考えられる．一方，初期電力量の少ない端末の平均生存時間は，クライアントの到着間隔の変化にともなう閾値の変化に大きく影響を受ける．そのため，クライアントの到着間隔の変化にすばやく対応できるが，平均生存時間が長くなるものと考えられる．

4.4.3 W_{plus} , W_{minus} の影響

ELEC 方式において， (W_{plus}, W_{minus}) の組を $(0.001, 0.002)$, $(0.002, 0.001)$, $(0.002, 0.002)$, $(0.001, 0.005)$, $(0.05, 0.1)$ と変化させたときの問合せ成功率，平均応答時間を表 4 に示す．また，端末の初期電力量をもとに 100 単位エネルギーごとに分類した平均生存時間を図 5 に示す．ELEC 方式における q は 20 とした．

図表から， W_{minus} が W_{plus} より大きいときの性能がよく， W_{plus} が W_{minus} より大きいと，同じ場合には性能が悪いことがわかる．また， W_{minus} は W_{plus} の 5 倍のときより，2 倍のときのほうが，初期電力量の小さい端末の平均生存時間が長くなっている． W_{minus} は W_{plus} の 2 倍の場合，閾値を緩めすぎると，2 回前の状態に戻る．これにより，閾値が最適値の付近で留まるため，性能がよくなったと考えられる．

また、 (W_{plus}, W_{minus}) が $(0.001, 0.002)$ の場合と $(0.05, 0.1)$ の場合には、性能にほとんど差がない。しかし一般には、 W_{plus} 、 W_{minus} が小さいと閾値が変化する速度が遅いため、クライアントの到着間隔が急激に変化した場合には対応できない。一方、 W_{plus} 、 W_{minus} が大きいと閾値が急激に変化するため、閾値が最適な値付近で留まらず、性能が悪化してしまうと考えられる。

5. 関連研究

これまでに、放送型情報システムにおいて、クライアント端末の消費電力を削減することを目的とした研究が多く行われている。

主な研究として、放送データにインデックスを付加することにより、クライアントが放送帯域を監視する時間を短縮し、消費電力を削減するというものが挙げられる。インデックスには、放送されるデータ内容と放送時間が含まれており、クライアントはインデックスを受信することで、必要とするデータが放送される時間を知る。インデックスを受信したクライアントは、必要なデータが放送される時間まで電源を落としておくことで、消費電力を抑えられる。文献 [9], [10] では、ツリー構造のインデックスや、インデックスの分散配置について論じている。また、文献 [6] では、放送するデータ内容を考慮したインデックス配置手法を提案している。

本研究では、問合せ処理に必要な電力に主眼を置き、効率的なインデックス構築手法には触れていないが、これらの研究のアイデアは容易に本研究に適用できる。

6. まとめ

本稿では、放送型データベースシステムにおいて、モバイル端末の電力消費を考慮して、動的に問合せ処理方式を選択する手法を提案した。提案手法では、閾値を設定することにより、残存電力が多い端末には優先的に消費電力の多いクライアント型方式を割り当て、残存電力が少ない端末には優先的に消費電力の少ないオンデマンド型方式を割り当てる。また、提案手法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、提案手法が従来手法と比べて、特に電力の少ない端末の生存時間を向上できることを確認した。

今後は、問合せごとの電力消費を考慮した閾値決定アルゴリズム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。

文 献

- [1] S. Acharya, R. Alonso, 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 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] Y. Chehadeh, A. Hurson, and L. Miller, "Energy-Efficient Indexing on a Broadcast Channel in a Mobile Database Access System," Proc. the International Conference on Information Technology: Coding and Computing (ITCC), pp.368–374, 2000.
- [7] K. Foltz, L. Xu, and J. Bruck, "Coding and Scheduling for Efficient Loss-resilient Data Broadcasting," Proc. IEEE International Symposium on Information Theory (ISIT), pp.413–419, 2003.
- [8] 箱根 聡, 田辺雅則, 石川裕治, 井上 潮, "放送型通信とオンデマンド型通信を統合した情報提供システム," 情報処理学会論文誌, Vol.40, No.10, pp.3772–3781, 1999.
- [9] T. Imielinski, S. Viswanathan, and B. R. Badrinath, "Energy Efficient Indexing on Air," Proc. ACM SIGMOD, pp.25–36, 1994.
- [10] T. Imielinski, S. Viswanathan, and B. R. Badrinath, "Data on Air: Organization and Access," IEEE Trans. Knowledge and Data Engineering, Vol.9, No.3, pp.353–372, 1997.
- [11] 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.
- [12] 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, Vol.J89-D, No.2, pp.151–162, 2006.
- [13] 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問合せ発生頻度に基づいた問合せ処理方式," 情報処理学会研究報告, Vol.2006, No.9, pp.175–182, 2006.
- [14] A. Mahesri and V. Vardhan, "Power Consumption Breakdown on Modern Laptop," Proc. the 4th Workshop on Power-Aware Computing Systems (PACS), 2004.

- [1] S. Acharya, R. Alonso, 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,