

センサデータベースにおける実時間分割応答の提案

金井 圭介[†] 石塚 宏紀[‡] 戸辺 義人^{†*}

東京電機大学工学部情報メディア学科[†] 東京電機大学大学院工学研究科[‡]

独立行政法人科学技術振興機構 CREST^{*}

1. はじめに

近年、センサの小型化・低価格化によりセンサネットワークの展開が進み、センサデータを蓄積するデータベースも研究開発されている。センサデータを利用した応用例に、データベースの応答時間にリアルタイム性の保障が必要な、人命を左右するサービスがある。しかし、既存のリアルタイムデータベースでは、時々刻々と変化するセンサデータに十分対応できていない。そこで、我々は、センサデータに特化した実時間分割応答を提案する。提案手法は、クエリ処理時間を予測し、クエリを適切に分割することで時間制約内に応答する。本研究において、我々は、提案手法のプロトタイプを実装し、簡易実験による実時間応答の検証を行った。

2. 研究課題

センサデータを利用した応用例には、災害時の避難支援や緊急医療支援¹⁾など、人命に係わるサービスがある。このようなサービスに対応するため、センサデータベースは、データの実時間処理が要求される。本提案は、クエリを分割応答することで、実時間処理を実現する。本提案において、データベースは、ユーザが指定した時間内に応答可能なタプル数の推測に基づいて分割クエリを再構築する。指定時間内に応答可能なタプル数の推測を行う際に、以下のセンサデータ特有の課題が生じる。

• 課題 1: センサデータに対するクエリ

センサデータとは、そのデータが観測された時間と位置によって意味付けされる。そのため、センサデータへのクエリは、観測時間、観測位置、観測値のいずれかが指定されたデータへの問い合わせとなる。ユーザが指定した時間内に処理可能なタプル数の推測に、データベース内のデータをそれぞれ観測時間、観測位置、観測値に基づいて解析し、把握しておく必要がある。

• 課題 2: 頻繁に挿入されるデータ

センサは時々刻々と測定し続けるため、センサデータベースには、頻繁に測定されたデータが挿入される。そのため、データベースの内容は、動的に変化する。クエリの分割に必要なタプル数の推測をするために、課題 1 のようにデータベース内のデータを把握しておく場合、データの挿入に合わせて動的にデータを把握しておく必要がある。

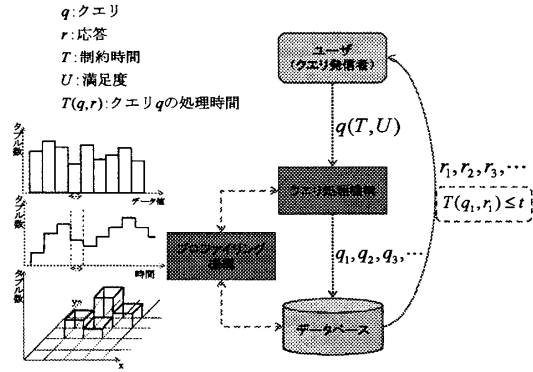


図 1. システム概要図

本論文では、これらのセンサデータ特有の課題を解決したセンサデータベースにおける実時間分割応答機構を提案する。

3. 提案手法

研究課題を解決するために、我々は、センサデータベース内のデータを解析し、動的に更新可能なデータ分布を作成することで、実時間分割応答を実現する。

図 1 にシステムの概要を示す。本システムにおいて、ユーザは、センサデータベースに対して制約時間 T 、満足度 U を含むクエリ $q(T, U)$ を発行する。 U を(1)式と定義する。 $q(T, U)$ を受信したセンサデータベースは、 $q(T, U)$ をプロファイリング機構に送り、作成したデータ分布を利用して q 本来の処理時間 T_q を推測する。次に、クエリ処理機構は T_q と T を比較して、 T 内で応答可能なタプル数をクエリの特徴に基づいたデータ分布を用いて推測し、そのタプル数に基づいてクエリを分割する。最終的に、センサデータベースは、 U が満たされるまでユーザに応答を繰り返す。次に、プロファイリング機構とクエリ処理機構について詳しく述べる。

$$U = \frac{\sum n_i}{N_q} \quad (1)$$

N_q : 本来のクエリによる応答予測タプル数

n_i : i 回目分割したクエリによる応答予測タプル数

• プロファイリング機構

プロファイリング機構は、ユーザが発行した $q(T, U)$ の処理時間を推測するクエリ処理時間推測機能と T 内に処理可能なタプル数を推測するタプル数推測機能で構成される。以下に、2つの機能の詳細を示す。

(ア) クエリ処理時間推測機能

クエリ処理時間推測機能は、 $q(T, U)$ に含まれる条件文のデータ分布を分析することで、 q に対して本来、応答すべきタプル数 N_q を推測する。 N_q に 1 タプルあ

Divided response processing for Real-time Sensor Database System
[†] Keisuke Kanai
[‡] Hiroki Ishizuka
^{*} Yoshito Tobe
 Department of Information Systems and Multimedia Design, Tokyo Denki University (†)
 Department of Information and Media Engineering, Tokyo Denki University (‡)
 CREST, Japan Science and Technology Agency (*)

Algorithm 1 Query Processing Algorithm.

```

1. procedure
2. var
3.    $N_c$ ://本来のクエリから推測されるタプル数
4.    $N_c:=0$ ;//分割したクエリから推測されるタプル数の和
5.    $P:=0$ ;//クエリ分割点
6.   if  $T_0 \leq T$ 
7.     then exec(q);
8.   else if  $T_0 > T$ 
9.     then while  $N_c/N_c < U$  do
10.       $n_i:=getEstimateTuple()$ ;
11.      //プロファイリング機能へ T 内に処理可能なタプル数を
12.      //問い合わせ
13.       $q_i:=divideQuery(q, n_i, P)$ ;
14.       $P:=getDividePosition(q_i)$ ;
15.      exec( $q_i$ );
16.       $N_c:=N_c+n_i$ ;
17.       $i:=i+1$ ;
18.   end procedure

```

図 2. クエリ処理アルゴリズム

りの処理時間 T_{tuple} をかけることで、クエリ処理時間の推測を行う。 T_{tuple} を求めるために、プロファイリング機構は、定期的にデータベースに対してクエリを発行する。このクエリによって得られた処理時間を実際に得られたタプル数で割ることにより、 T_{tuple} を求める。また、ユーザが発行するクエリに対しても同様に T_{tuple} を求め、平均値を求める。平均値をとることで、データベースの更新に対して、 T_{tuple} の誤差を修正する。

(イ) タプル数推測機能

タプル推測機能は、 T 内に処理可能なタプル数 n_i を推測する。 n_i は、 T を、 3.1.1 節で述べた T_{tuple} で割ることにより求める。

• クエリ処理機構

図 2 にクエリ処理アルゴリズムを示す。まず、クエリ処理機構は、プロファイリング機構に問い合わせ、 n_i を得る。クエリに含まれる条件文の小さい値の方から、データ分布の値を足していき、 n_i を超えない分割点 q_i を求める。 q_i を用いて、 q を分割する。

4. プロトタイプの実装と評価

我々は、提案手法の実装を行い、簡易実験を行った。テストセンサデータとして、UScan²⁾で実際に街の中にセンサを配置し、細粒度に取得したデータを利用する。本実験では、このUScanのデータの中でも、2007年8月3日から2007年8月7日までの約11万タプルのデータを利用した。また、UScanでは、温度データを取得しているため、観測値は温度とした。実験は、3つのシナリオを想定して行う。

- ▶ **実験 1** : 時間幅指定によるデータの要求
SELECT * FROM data WHERE time BETWEEN '2007-08-03 00:00:00' AND '2007-08-04 00:00:00';
- ▶ **実験 2** : 温度幅指定によるデータの要求
SELECT * FROM data WHERE value BETWEEN 36 AND 40;
- ▶ **実験 3** : 領域指定によるデータの要求
SELECT * FROM data WHERE (latitude BETWEEN 139.761 AND 139.763) AND (longitude BETWEEN 35.692 AND 35.693);

これらのクエリを、制約時間 0s から 60s まで 1s ずつずらしながら実行し、実時間分割応答の評価を行った。図 3 に実験結果を示す。

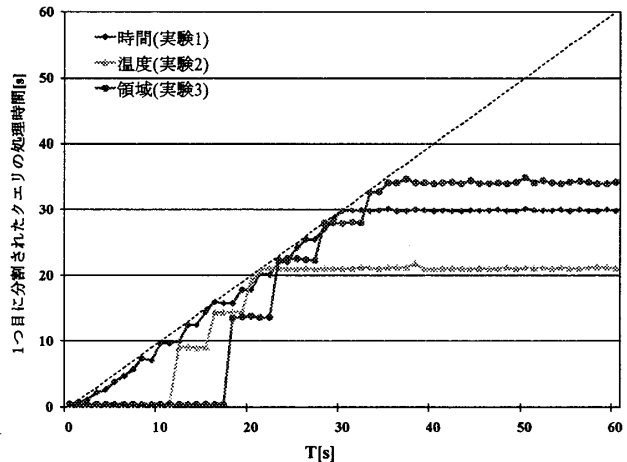


図 3. 実時間分割処理の精度

実験結果は、それぞれ実験 1、実験 2、実験 3 のクエリに対して、1 つ目に分割されたクエリの処理時間を Y 軸に示す。分割されたクエリは、実時間処理として有効といえる精度があり、また、ユーザが指定する制約時間がクエリの処理時間を上回ると、クエリを分割する必要がなくなるため、応答時間が一定になることが明らかになった。

5. 関連研究

本研究は、頻繁に生じる膨大なセンサデータに対するデータベース処理技術の 1 つを提案するものである。クエリの実時間処理機能を有するシステムに Aurora³⁾がある。しかし、Aurora の実時間クエリ処理技術は、データベースに対する問い合わせ処理の高速化ではなく、分散データベース群に対する効率的なクエリ伝搬のスケジューリングに注目しているため、本研究の目指すものと異なる。また、KRAFT⁴⁾は処理の高速化を追求するものであり、我々とは異なる目標を目指している。

6. まとめ

本研究では、センサデータに特化した実時間分割応答のための手法を提案した。また、プロトタイプを作成して簡易実験を行い、クエリを分割することで、実時間処理が可能となることを確認した。

参考文献

- 1) CodeBlue : Wireless Sensor Networks for Medical Care, <http://www.eecs.harvard.edu/~mdw/proj/codeblue/>
- 2) UScan: Urban Scanning <http://uscan.osoite.jp/>
- 3) Daniel J. Abadi, Donald Carney, Ugur Cetintemel, Mitch Cherniack, Christian Convey, Sangdon Lee, Michael Stonebraker, Nesime Tatbul, and Stanley B. Zdonik. Aurora : a new model and architecture for data stream management. VLDB Journal, 12(2), (2007)
- 4) Hideyuki Kawashima. Kraft: A real-time active dbms for signal streams. In Proc of the International Conference on Networked Sensing Systems, (2007)