D-015

センサデータベースにおけるメモリ消費量を考慮した実時間分割応答の提案

A RealTime Query Processing considering Memory Consumption for Massive Sensor Databases

金井 圭介† 石塚 宏紀‡ 戸辺 義人§*
Keisuke Kanai Hiroki Ishizuka Yoshito Tobe

1. はじめ

近年，センサネットワークの普及とともに，センサデータの蓄積と処理を行うためのセンサデータベース技術が必要となってきている．センサデータは時々刻々と複数センサから取得されるため膨大なデータ量となる．また，この膨大なデータへの問い合わせは，取得するデータ量も膨大となるため，メモリに負荷をかけ，クエリ処理速度を著しく低下させる．そこで我々は，ユーザが指定した時間内に応答する実時間分割応答を提案する．提案手法では，ユーザが指定した時間内に処理可能なデータ量，およびメモリ負荷を考慮してサブクエリを構築することで，本来応答するデータの 1 部だけでもユーザが指定した時間内に応答する．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

膨大なセンサデータを利用するサービスにおいて，サービスが予期した時間内に応答がないとサービスに支障が生じてしまう．このようなサービスに対応するために，センサデータベースは問い合わせの実時間処理が要求される．従来のデータベース技術としてクエリ処理の高速化[1]・クエリ伝搬の効率化[2] などがあるが，我々の目指すものと異なる．そこで，我々はクエリを分割し，サブクエリを構築することで実時間応答を可能にする実時間分割応答[3]を提案した．しかし，膨大なデータに対する問い合わせは，取得するデータも膨大となるため，メモリ資源を消費してしまう．これにより，クエリ処理速度が低下し，クエリ処理時間予測が困難になることが考えられる．我々は，まず，膨大なセンサデータに対して問い合わせを発行する基礎実験を行い，メモリ使用量とクエリ処理時間の関係を明らかにした．

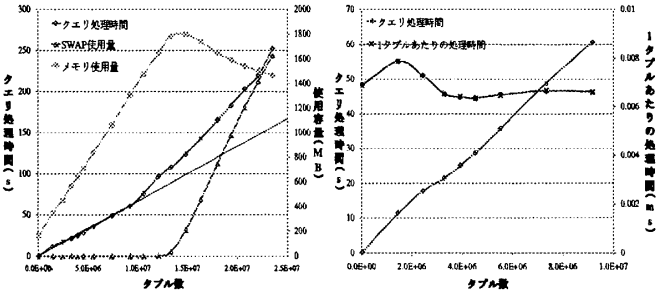
3. 基礎実験

膨大なセンサデータに対してクエリを発行する基礎実験を行った．実験環境を表 1 に示す．テストセンサデータには，UScan[4]で実際に街にセンサを配置して取得した 25,424,688 タプルのデータを利用した．本実験では，取得するデータ量が徐々に増えるようにクエリを発行し，その際の処理時間とメモリ使用量に注目した．実験結果を図 1，図 2 に示す．

図 1 より，取得するデータ量が増えるにつれ，処理時間，メモリ使用量，SWAP 使用量が増加している．また，

表 1: 基礎実験環境

OS	Linux Centos5
メモリ	2GB
SWAP	2GB
DBMS	PostgreSQL 8.1.11
言語	C言語



(左) 図 1: タプル数増加に伴うクエリ処理時間および使用容量の推移
(右) 図 2: タプル数増加に伴うクエリ処理時間および 1 タプルあたりの処理時間の推移

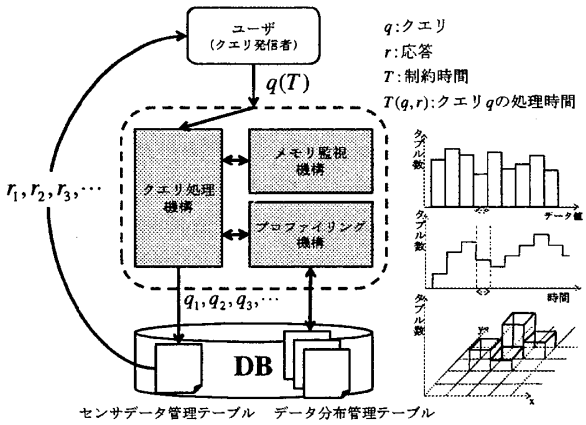


図 3: システム概要図

メモリ使用量が 75%以上に達すると処理時間が著しく低下し始めていることがわかる．

図 2 より，メモリ使用量 75%以下において，取得するデータ量と処理時間には比例関係があり，処理時間をデータ量で割った 1 タプルあたりの処理時間がほぼ一定であることがわかる．

本論文では，基礎実験より，取得データ量増加に伴うメモリ使用量を考慮した実時間分割手法を提案する．

4. 提案手法

基礎実験より，取得データ増加に伴うメモリ使用量増加は，処理時間を著しく低下させる．そこで，我々は，クエリ処理前のメモリ空き容量を取得し，クエリ処理速度が低下しない範囲のメモリ使用量で，クエリを分割処

† 東京電機大学大学院 工学研究科
‡ 東京電機大学大学院 先端科学技術研究科
§ 東京電機大学 工学部 情報メディア学科
* 科学技術振興機構 CREST

Algorithm1 *Query Division Processing Algorithm*

```
1.  procedure QueryProcesing(q(T))
2.    var
3.      T_past:=0 //経過時間
4.      T_end //終了処理のオーバーヘッド
5.      P:=0 //クエリ分割点
6.      q_i:=q
7.    while
8.      M_free:=GetFreeMemory()
9.      N_est:=GetEstimateTuple(q_i)
10.     N_time:=GetProcessableTuple(T-T_past)
11.     N_mem:=GetProcessableTuple(M_free)
12.     if (N_time < N_est) || (N_mem < N_est) then
13.       N_i:=GetProcessableTuple(N_time,N_mem)
14.       q_i:=DivideQuery(q,N_i,P)
15.       P:=GetDividePosition(q_i)
16.     else
17.       Exec(q_i)
18.       if (T-T_past < T_end) || P:=EndOfQueyr then
19.         break
20.       else
21.         q_i:=ReassembleQuery(q,P)
22.       end if
23.     end if
24.   end while
25. end procedure
```

図 4: クエリ分割処理アルゴリズム

理する．これにより，タプル数と 1 タプルあたりの処理時間を用いた処理時間予測を可能にするとともに，単位時間あたりのデータ処理量を増加させる．

4.1 処理手順

図 3 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 を含むクエリ q(T)を発行する．q(T)を受信したセンサデータベースは，q(T)をプロファイリング機構に送り，時間・領域・値毎に作成しておいたデータ分布を利用して q によって取得されるタプル数 N_{est} と T 内に処理可能と推測されるタプル数 N_{time} を得る．次に，メモリ監視機構により現在のメモリ状態を取得し，メモリ使用量が閾値を超えない範囲で処理できるデータ量 N_{mem} を得る．最後に，クエリ処理機構は，N_{est}，N_{time}，N_{mem} を用いて適切にクエリを分割する．最終的に，センサデータベースは，T が経過するまで応答を繰り返す．

4.2 クエリ処理機構

クエリ分割処理アルゴリズムを図 4 に示す．クエリ処理機構では，プロファイリング機構・メモリ監視機構を用いて q(T)の N_{est}，N_{time}，N_{mem} を得る(line9-11)．次に，q_i が T 内に処理でき，かつメモリ負荷を抑えられるか判定し(line12)，処理できない場合，クエリの分割処理を行う(line13-15)．本提案では，処理を簡略化するためにクエリに含まれる条件文の端から分割処理を行っていくものとする．分割されたクエリは DB に発行され(line17)，T から経過時間を引いた残り時間が閾値以上あった場合，処理を続ける(line18-22)．

5. 実験と評価

我々は，メモリ使用量を考慮した提案手法と，メモリ使用量を考慮しない既存手法の比較を行った．本実験では，実験環境およびテストデータは基礎実験と同じものを利用し，時間幅指定によるデータ要求のみに注目した．

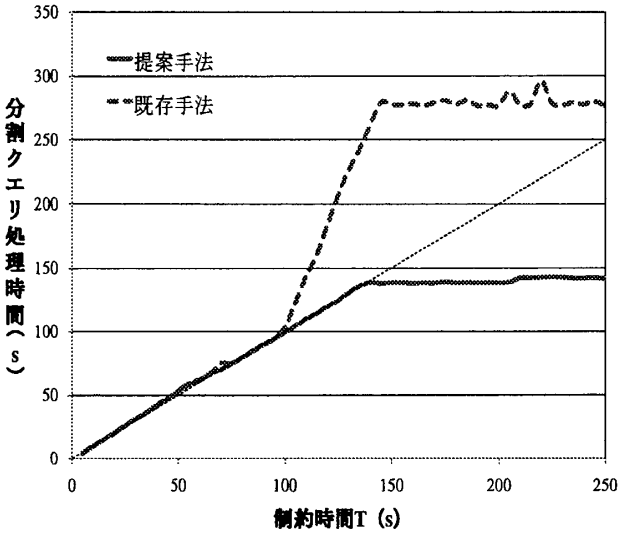


図 5: 実時間分割応答の精度

◆ 時間幅指定によるデータ要求

```
SELECT * FROM data WHERE time BETWEEN
'2007-08-01 00:00:00' AND '2007-08-17 23:00:00';
```

このクエリを，制約時間 5s から 250s まで 5s ずつずらしながら発行し，実時間分割応答の評価を行った．図 5 に実験結果を示す．

実験結果は，X 軸に制約時間を，Y 軸にクエリ処理時間を示す．既存手法では，メモリ負荷を考慮していないため，制約時間が増え，取得するデータ量が増加すると実時間応答の精度が悪くなっている．しかし，提案手法では，データ量が増加しても，制約時間を破ることなくクエリを処理していることがわかる．

6. まとめ

本研究では，大規模なセンサデータベースにおけるクエリ処理の問題を明らかにし，メモリ負荷を考慮した実時間応答手法を提案した．また，既存手法との比較を行い，提案手法の有用性を評価した．

今後，複数条件指定時の処理に対応していく必要があるとともに，応答するデータの優先度を考慮していきたい．

参考文献

[1] Kawashima, H.: Kraft: A Real-Time Active DBMS for Signal Streams, In Proc. of the International Conference on Networked Sensing Systems, (2007).

[2] Abadi, J.D, Carney, D., Cetintemel, U., Cherniack, M., Convey, C., Lee, S., Stonebraker, M., Tatbul, N., and Zdonik, B.S.: Aurora : a new model and architecture for data stream management, VLDB Journal, 12(2), (2007).

[3] 金井 圭介, 石塚 宏紀, 戸辺 義人: センサデータベースにおける実時間分割応答の提案, 情報処理学会 第 70 回全国大会, (2008).

[4] UScan: Urban Scanning http://uscan.osoite.jp