

高性能センサデータベース

郡 光則*
米田貴雄*

High-performance Sensor Database

Mitsunori Kori, Takao Yoneta

要旨

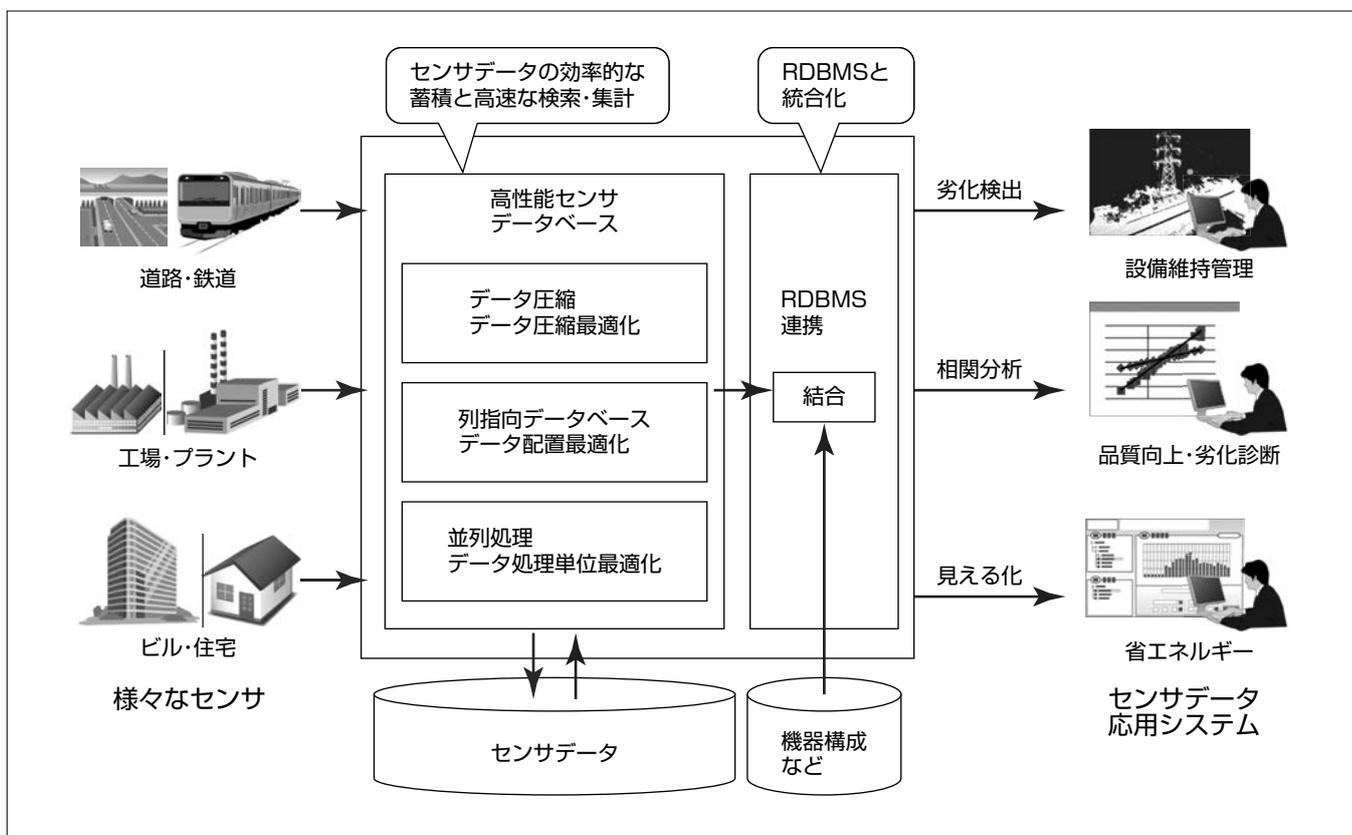
様々な機器がネットワークに接続されるIoT(Internet of Things)の時代を迎え、機器の生み出す大量のセンサデータの活用が進んでいる。道路や鉄道などの社会インフラ設備の維持管理における劣化箇所の検出、工場やプラントの稼働状況の分析による製品品質の向上や劣化診断、ビルや住宅の電力消費量の見える化による省電力など、様々な分野で大量のセンサデータを活用する事例が増えつつある。

しかし、RDBMS(Relational DataBase Management System)に代表される従来のデータベースは大量のセンサデータを効率的に扱うことができなかった。近年では、多数のサーバによる並列分散処理、大容量主メモリを利用したインメモリ処理、フラッシュメモリを利用した高速ストレージの

利用などの対策がとられることがあるが、これらは高価なハードウェアを必要とする。

三菱電機では、これらの課題を解決する“高性能センサデータベース”を開発した。当社が従来研究開発を行ってきた“高速集計検索エンジン”にセンサデータの特徴に応じたデータ圧縮、データ配置、データ処理単位の最適化を適用することによって、RDBMS比で10~1,000倍の高速な蓄積・検索・集計とストレージ容量の1/10~1/1,000の削減を実現した。

また、高性能センサデータベースをRDBMSと統合化し、JDBCなどRDBMSのインタフェースによる検索・集計や機器構成データとの結合などの円滑なデータ連携を実現した。



高性能センサデータベースとその応用分野

高性能センサデータベースは社会インフラの設備維持管理、工場やプラントの稼働状況監視、エネルギー管理などの分野でセンサデータの迅速な活用方法を提供する。センサデータの特徴に合わせた最適化によってRDBMS比10~1,000倍の高速処理を実現し、ストレージ容量を1/10~1/1,000に削減した。RDBMSのインタフェースによる検索・集計やRDBMSで管理するデータとの結合などのデータ連携も可能となる。

1. ま え が き

様々な機器がネットワークに接続されるIoTの時代を迎え、機器の生み出す大量のセンサデータの活用が進んでいる。道路や鉄道などの社会インフラ設備の維持管理における劣化箇所の検出、工場やプラントの稼働状況の分析による製品品質の向上や劣化診断、ビルや住宅の電力消費量の見える化による省電力など、様々な分野で大量のセンサデータを活用する事例が増えている。

増え続けるセンサデータ活用の要求に応えるため、当社ではセンサから得られた大量のデータを高速に蓄積・検索・集計する“高性能センサデータベース”を開発した。

2. センサデータ管理の動向

2.1 センサデータの急増

ネットワークやストレージなどの価格低下を背景に、近年、各種機器の生成するセンサデータの増加が著しい。ある調査によると⁽¹⁾、全世界で生成されるデータの総量は年率約40%で増加しているが、中でも機器の生成するデータの増加が著しく、2013年では全データの2%がセンサデータであったが、2020年には10%を占めると予想されている。

2.2 従来の高速処理技術の課題

センサデータには数値、ビット、IDなどの種類があり、1件当たりのサイズは小さいが膨大な件数が蓄積されることが多い。また、時系列的に連続的に変化する値を取りやすく、しばしば大量一括処理されるといった特徴がある。

ところが、RDBMSに代表される従来のデータベースはこのような特徴を持つセンサデータの管理を主目的として設計されてこなかった。このため大量のセンサデータを扱うシステムでRDBMSを使用するとしばしば処理速度の不足が課題となってきた。

この問題の対策として、1,000台を超える多数のサーバによる並列分散処理⁽²⁾、数十テラバイトに及ぶ大容量主メモリによるインメモリ処理、ビット単価が高いが高速なフラッシュメモリの使用等の方法が知られている。しかし、これらの方法は高価なハードウェアを必要とする。

また、センサデータは一般に機器構成など様々なデータと組み合わせて利用されるため、RDBMSで管理されることの多い各種データとの円滑な連携が必要になる。

3. 高性能センサデータベース

3.1 高性能センサデータベースの特長

2.2節で述べた課題を解決するため、当社では次の特長を備えた“高性能センサデータベース”を開発した(図1)。

(1) センサデータの効率的な蓄積と高速な検索・集計

当社が従来研究開発をしてきた“高速集計検索エンジン”⁽³⁾⁽⁴⁾をセンサデータ向けに最適化し、RDBMS比10~

1,000倍の速度、1/10~1/1,000のストレージ容量を実現した。

(2) RDBMSと統合化

“RDBMS連携”⁽⁵⁾によって、高性能センサデータベースをRDBMSの拡張機能として利用可能とし、RDBMSで管理する機器構成等のデータとの結合を実現した。

3.2 高性能センサデータベースの基本機能

高性能センサデータベースはRDBMSと同様の行・列構造の表(テーブル)の形式でデータを管理する。高性能センサデータベースでは最大100万の列を定義可能とした。これによって、約100万のセンサから同一時刻に計測されたセンサデータを1行に格納することができる(図2)。

高性能センサデータベースでは表1に示す各種のデータ型を扱うことができる。また、高性能センサデータベースの主要機能を表2に示す。

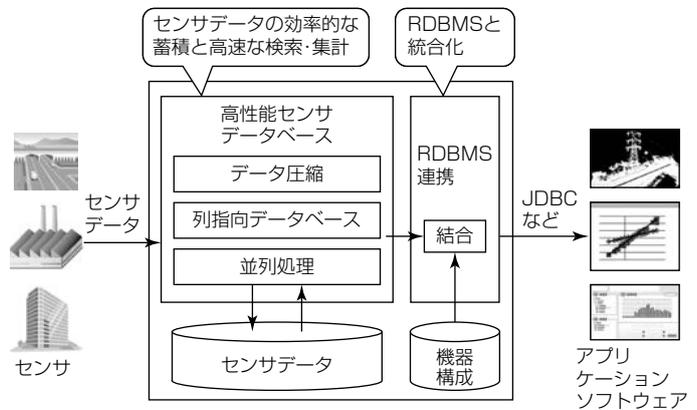


図1. 高性能センサデータベースの構成

最大100万列

行ID	ロード時刻	計測時刻	センサ#1	センサ#2	センサ#3	...	センサ#N

図2. データ格納形式の一例

表1. 高性能センサデータベースのデータ型

データ型	表記
文字列型	char(n), varchar(n)
整数型	int(n)
実数型(十進数)	dec(p, s)
浮動小数点型	real, double precision
日付型	date
タイムスタンプ型	timestamp

表2. 高性能センサデータベースの主要機能

分類	説明
定義	<ul style="list-style-type: none"> ユーザーの定義、削除、パスワード変更 スキーマ/表/パーティション/ビューの生成/削除/情報取得
ロード	<ul style="list-style-type: none"> 追加ロード、トランケート、トランケートロード
検索	<ul style="list-style-type: none"> 選択条件(where)による行の抽出(選択)、列の抽出(射影) 集合演算(group by)、集合演算結果の抽出(having) 四則演算、ビット演算、論理演算、文字列操作 算術関数(sin/log/exp等)、統計演算(分散、相関係数等) ソート(order by)、取得件数指定(limit/offset) 副問合せ、合併(union)
運用	<ul style="list-style-type: none"> バックアップ、リストア、データ配布

3.3 高性能センサデータベースのシステム構成

高性能センサデータベースはメモリ最小4GBのPCサーバ上で動作し、OSとしてWindows^(注1)又はLinux^(注2)、連携対象のRDBMSとしてPostgreSQL又はOracle^(注3)を使用できる。

高性能センサデータベースは、サーバ1台から利用可能であるが、規模や要求性能に応じてISN(Intelligent Storage Node)と呼ぶサーバを最大256台まで追加できる(図3)。センサデータを長期間保存するシステムでは、運用開始後時間とともに蓄積されるデータ量が増加する。また、小規模の実データで試験運用を行い、効果を確認した後にこの運用へ移行する場合がある。このような場合でもデータの移行やアプリケーションソフトウェアの変更なしにストレージ容量の拡張と処理速度の向上を実現できる。

(注1) Windowsは、Microsoft Corp. の登録商標である。
 (注2) Linuxは、Linus Torvalds氏の登録商標である。
 (注3) Oracleは、Oracle Corp. の登録商標である。

3.4 センサデータのロード

高性能センサデータベースには基本的にデータを追記する機能は提供しているが、データを1行ずつ更新・削除する機能は提供していない。これは、データを細かい単位で更新・削除する処理は一般に負荷が高く処理速度低下要因になるためである。ただし表を“パーティション”に分割して管理することができる。例えば月単位でパーティションを作成し、1年を経過したパーティションを削除するという運用を行う(図4)。

3.5 センサデータの検索

高性能センサデータベースはRDBMSを介して検索できる。高性能センサデータベースの問い合わせ文を埋め込んだSQL文によってRDBMSに対して検索要求を行うことで、

高性能センサデータベースの検索を行うことができる。

また、高性能センサデータベースの検索結果とRDBMSのテーブルを結合するなど、データベースの連携が可能である。図5にRDBMSとしてPostgreSQLを使用した場合の連携SQL文の一例を示す。この例では高性能センサデータベースに格納された温度データを日付、部屋番号ごとに集計し、RDBMSに格納されている部屋名称と結合して出力する。

4. 高性能センサデータベースの高速処理技術

大規模データの高速処理の手法として、従来の“高速集計検索エンジン”では次のような方法を用いていた。

- (1) データ圧縮によるストレージ入出力の削減
- (2) ストレージ上への列指向のデータ配置
- (3) プロセッサ内の多数のコアによる並列処理

高性能センサデータベースでは、これらの技術に加え、センサデータ向けの最適化によって高速処理を実現した(図6)。

4.1 データ圧縮

データを圧縮してストレージに格納することによって、ストレージ容量を削減するとともにストレージ入出力の削減による高速化が可能になる⁽⁶⁾。

センサデータには、値が連続的に変化する数値、各ビットが独立に意味を持つビットデータ、離散的な値を取るIDなど様々な種類があり、変化の大きいデータや小さいデータ等様々な性質を持つ。高性能センサデータベースでは、様々なデータに対応するため、700通り以上の基本符号化方式の組み合わせに対して圧縮後のサイズを計算し、サイズの最小となるものを採用する(データ圧縮最適化)。

なお、センサデータの圧縮では、圧縮率向上が可能だが圧縮前のデータを正確に復元できない非可逆圧縮を用いる場合がある。高性能センサデータベースでは圧縮前データを完全に復元できる可逆圧縮を行う。

4.2 列指向データベース

RDBMSでは同一行内のデータをストレージ上の連続領域に固めて配置する行指向データベースと呼ぶ方式が一般的であるが、情報分析用途のデータベースでは同一の列に含まれるデータを固めて配置する列指向データベースと呼ぶ方式を採用する場合がある。

一例として、図2の表でセンサ#1のデータの平均を求

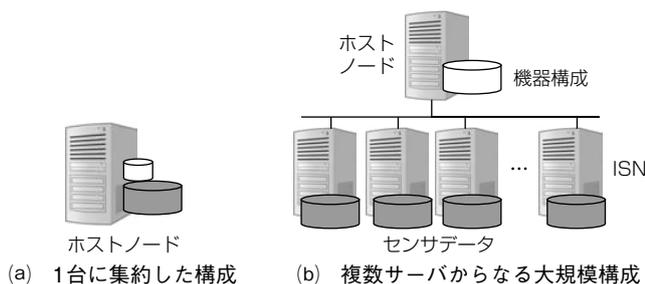


図3. ISNを用いた構成

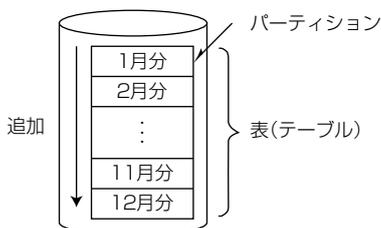


図4. パーティション

```

SELECT T1.date, T2.room_name, T1.avg_temp
FROM rds_sae_link (
    'saeuser', 'pwd', 'sae_db',
    'SELECT date, room_id, AVG (temperature)
     FROM sae_tb
     WHERE date = DATE '20100215'
     GROUP BY date, room_id'
) AS T1 (
    date, date,
    room_id integer,
    avg_temp dec(4,1)
),
pg_tb AS T2
WHERE T1.room_id = T2.room_id;
    
```

図5. 連携SQL文の例(RDBMSとしてPostgreSQLを使用した場合)

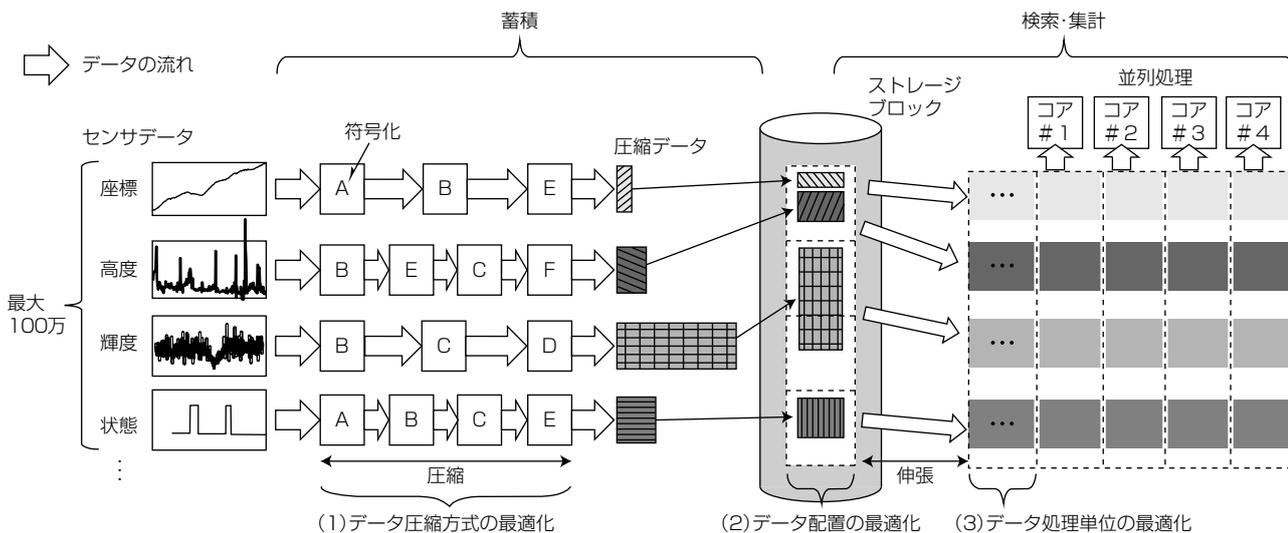


図6. 高性能センサーデータベースの高速処理技術

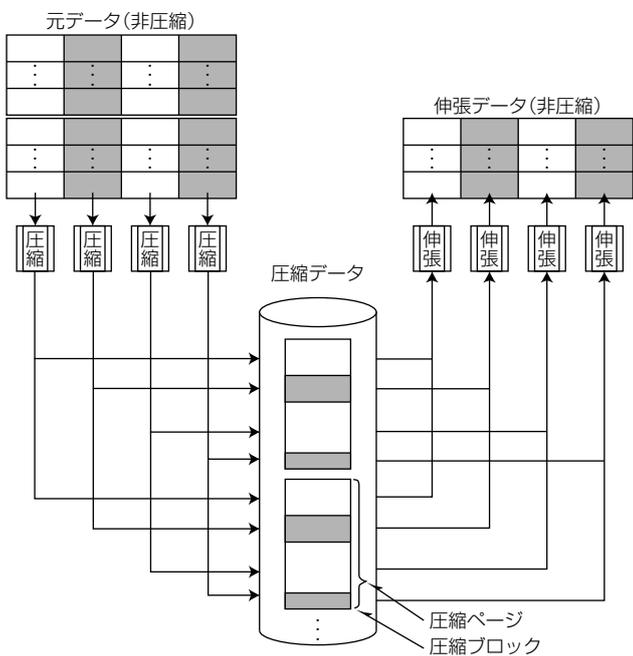


図7. ブロック化トランスポート配置

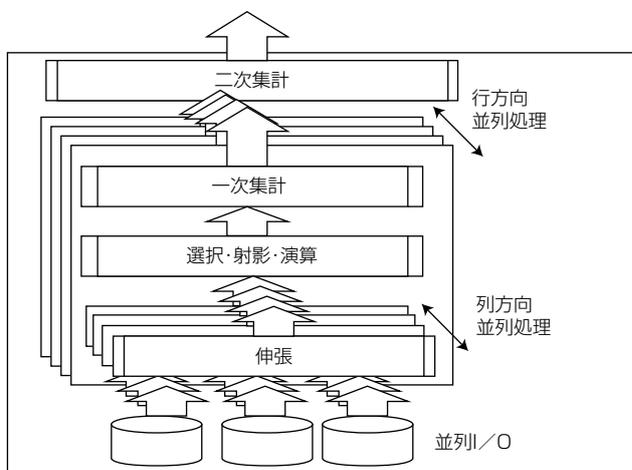


図8. 集計の並列処理の一例

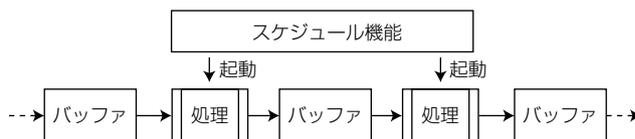


図9. 並列処理の起動制御

める場合、行指向データベースではセンサ#1のデータが分散して配置されるため、全データを読み出した後、センサ#1のデータを抽出する必要がある。列指向データベースではセンサ#1のデータが固めて配置されるため、センサ#1のデータだけを読み出すことが可能である。さらに、センサ#1のデータが連続して配置されるため圧縮率が向上しやすい。

高性能センサーデータベースは列指向のデータ配置を基本とし、行指向を複合した配置(ブロック化トランスポート配置)を行う(図7)。

ブロック化トランスポート配置では、列ごとの圧縮率の差によって、多数の小さい圧縮ブロックと少数の大きな圧縮ブロックが生成されることが多い。ストレージ装置にはストレージブロックと呼ぶ通常512バイト又は4,096バイト固定長の最小アクセス単位があるが、高性能センサ

データベースは、圧縮ブロックが極力ストレージブロックをまたがらないように配置したり、小さい圧縮ブロックを1個のストレージブロック内に配置することによって、ストレージアクセス時間の最小化を図る(データ配置最適化)。

高性能センサーデータベースは範囲索引と呼ぶ機能を備えている。圧縮ページごとに列の値の最大・最小値を保持することによって、データ全体を読み出さずに目的とする値が該当する圧縮ページ内に含まれるか否かを高速に判定する。時刻や連番、移動体の座標や積算値など単調に変化することの多い値を検索条件とする検索の高速化に有効である。

4.3 並列処理

近年のプロセッサ性能の向上はコア数の増加に依存する面が大きく、性能向上のためにメモリアクセス効率に注意を払う必要性が高まっている。

高性能センサデータベースでは、データを行又は列方向に分割して並列処理によって高速化を図っている。集計の例を図8に示す。伸張や一次集計、二次集計などの処理はバッファを介してデータを受け渡す(図9)。

各処理の使用するバッファは一般に大きい方が処理の切替えに伴うオーバーヘッドを削減できるが、大きすぎるとデータがプロセッサ内のキャッシュメモリ内に収まらず、処理速度が低下する。高性能センサデータベースでは処理内容に応じてバッファ内の行数を調整することによって速度向上を図っている(データ処理単位最適化)。

表3. 性能評価に用いたサーバの構成

CPU	Intel Xeon ^(注4) E7-4870 2.4GHz Core 10×4
ストレージ	1.2TB, 10,000rpm, 2.5型SAS×4(RAID5)

(注4) Xeonは、Intel Corp.の登録商標である。

表4. 性能評価に用いた処理の内容

処理	電力消費量の集計	
	内容	電力計測値
データ	計測周期	10秒
	センサ数	1,000
	期間	10年分
	データ件数	314億件

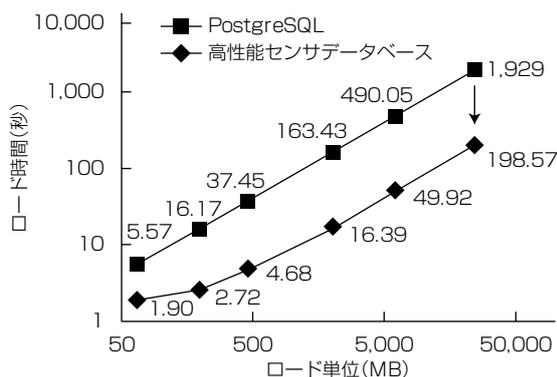


図10. ロード性能の比較

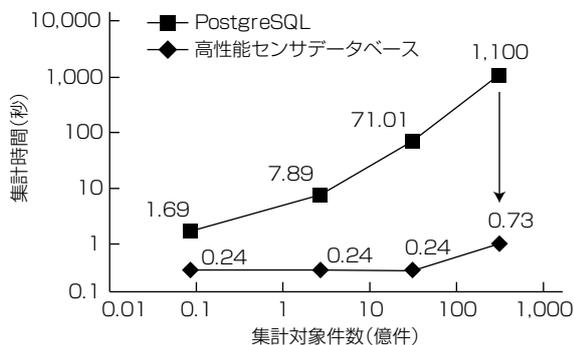


図11. 集計性能の比較

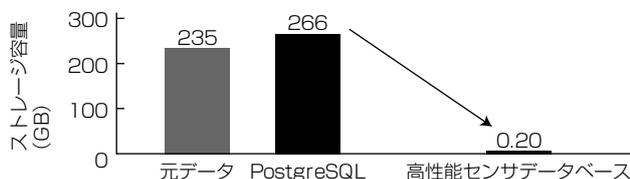


図12. ストレージ容量の比較

5. 性能評価

高性能センサデータベースの性能をPostgreSQLと比較した例を示す。比較に使用したハードウェア構成を表3に、処理の内容を表4に示す。PostgreSQLではロードにpsqlのCOPYコマンドを使用し、検索高速化のため索引を使用した。

ロード性能、集計性能、ストレージ容量の比較をそれぞれ図10、図11、図12に示す。PostgreSQLと比較して10~1,000倍の処理速度向上と1/1,000のストレージ容量削減が実現されている。

6. 高性能センサデータベースの応用分野

高性能センサデータベースは、次の分野を始めとする様々なセンサデータ応用システムへの適用を目指している。

(1) 社会インフラ設備の維持管理

道路や鉄道で周辺構造物の三次元計測データを蓄積し、劣化箇所の検出などを通じて維持管理の効率化を実現。

(2) 工場・プラントの稼働状態管理

電圧、温度、圧力等の計測データを蓄積、製品品質の分析や設備の劣化診断を通じて品質向上や生産効率化を実現。

(3) エネルギー管理

ビルや家庭で計測した電力消費量や温度などのセンサデータを蓄積し、見える化によって省エネルギーに活用。

7. むすび

センサデータの効率的な蓄積と高速な検索・集計を実現し、RDBMSと連携可能な高性能センサデータベースを開発した。今後は様々な応用システムへの適用を進め、そこで得られた知見をフィードバックし、改良を進める予定である。

参考文献

- (1) EMC: The Digital Universe of Opportunities: Rich Data and the Increasing Value of the Internet of Things <http://www.emc.com/leadership/digital-universe/2014iview/index.htm>
- (2) Apache Hadoop <http://hadoop.apache.org/>
- (3) 郡 光則, ほか: 高性能並列情報検索技術, 三菱電機技報, **83**, No.12, 705~708 (2009)
- (4) 山岸義徳, ほか: 高速集計検索エンジンとセンサデータベースへの応用, 三菱電機技報, **83**, No.12, 709~712 (2009)
- (5) 竹内文志, ほか: オープンソースDBMS拡張によるセンサデータベースの実現, 情報処理学会第74回全国大会講演論文集, 545~547 (2012)
- (6) 郡 光則: データウェアハウス向け高性能データ圧縮方式, 情報処理学会論文誌, **47**, No.SIG13 (2006)