

時系列データ管理へのデータベースアクセス サービス技術の適用検討

竹内 丈志† 永嶋 規充† 山足 光義†

三菱電機株式会社 情報技術総合研究所†

1. はじめに

近年、スマートフォンやカーナビをはじめとした移動体通信端末の高機能化に伴い、従来の位置情報に加えて、車などの乗り物の状態情報なども取得可能となり、これらの活用方法を検討する動きが盛んに行われている。また、クラウドサービス化の流れも加速しており、今後は移動体通信端末から定期的に発生する様々な報時系列データを、クラウドサービスで提供する要求が高まると考えられる。本稿では、位置情報や速度、温度や距離などの時系列データを対象に、当社保有のデータベースアクセスサービス技術の適用検討と性能評価について報告する。

2. データベースアクセスサービス技術

当社では、各種システムのクラウドサービス化に対応するため、データベースアクセスサービス技術(DBAS)を開発している [1][2][3]。DBAS は、データベース内のデータに Web インタフェースでアクセス可能とする技術で、以下を特徴とする。

- データベース内のテーブル構造を隠蔽可能
- データの開示範囲の変更に柔軟に対応可能

DBAS では、任意のサービス名と実行する SQL 文の組合せとの対応付けを、『サービス定義ファイル』に記述することで、外部のアプリケーションに公開するデータを制限すると同時に、データベース内のテーブル構造も隠蔽する。また、サービス定義ファイルを変更することで、サービス内容の変更に容易に対応できる。

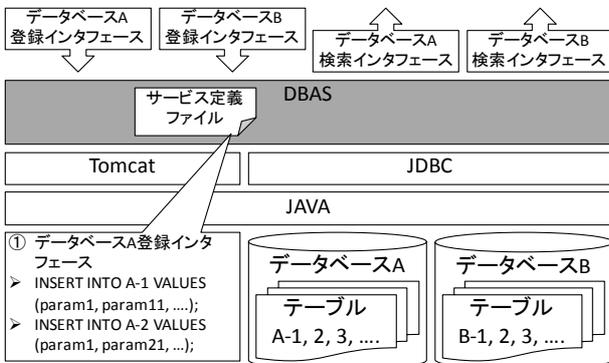


図 1 : データベースアクセスサービス技術

3. 時系列データ管理への対応

3.1. 課題

位置情報や状態情報などの時系列データは、以下のデータ特性をもつ。

- 定周期で一定量のデータが発生
- 長期保存には大規模なストレージ領域が必要

上記特性をもつデータを一般的なリレーショナルデータベース(RDB)で管理する場合、時間の経過に伴ってテーブル当たりのデータ蓄積量が増大し、登録/検索性能が低下する、という課題がある。また、データベースのストレージ容量も単調に増加し続けるため、保存対象期間を経過した古いデータを定期的に削除する仕組みが必要である。

3.2. 実現アプローチ

前述の課題に対応するには、テーブルパーティションの適用が有効である[4]。テーブルパーティションは、テーブル内に複数の継承テーブルを定義することで、データの分散配置を実現する。これにより、データ登録/検索処理の対象範囲を局所化し、データ蓄積量の増大に伴う性能劣化を抑制する。また、一定期間経過後にパーティション単位によるデータ削除を実現でき、任意の保存期間を設定してデータベースのストレージ容量を一定に保つことができる。

本稿では、図 2のように 1ヶ月分のデータを1個のパーティションに格納し、保存期間を1年とする構成について検討し、1個のテーブルが13個の継承テーブルを維持するパーティション運用の自動化を実現した。

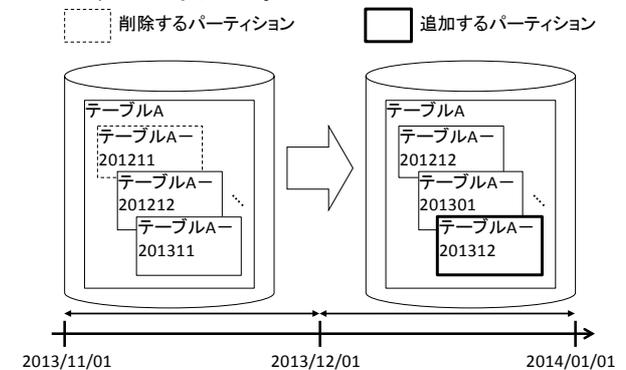


図 2 : テーブルパーティションの運用イメージ

A study for applying the database access service technology to time series data management

† Takeshi Takeuchi, Norimitsu Nagashima, Mitsuyoshi Yamatari, Information Technology R&D Center Mitsubishi Electric Corporation

4. 実現方式

テーブルには2個のトリガを定義する。1個は、パーティションの追加/削除を定期実行する『パーティション管理トリガ』、もう1個は、テーブル内の適切なパーティションへデータ振り分けを実行する『データ振り分けトリガ』である。

いずれのトリガも、実行のタイミングは新規データ登録(INSERTのSQL文実行)時とし、パーティション管理トリガ→データ振り分けトリガの順序でテーブルに定義する。

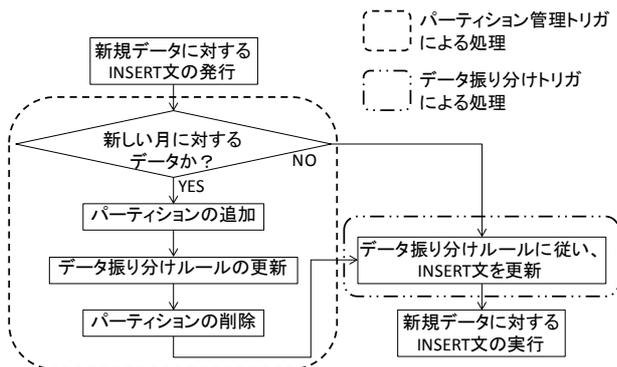


図3：各トリガによるデータ処理の流れ

各トリガによるデータ処理の流れと処理の内訳は、図3に示すとおりである。なお、図3における条件分岐は、既存のテーブル名に含まれる年および月と、新規データ内の年および月をそれぞれ比較することで実現した。

5. 評価

5.1. 評価環境

評価は、24[Byte/レコード]の2カラムで構成されるテーブルAと、276[Byte/レコード]の26カラムで構成されるテーブルBを定義し、100万件のデータを1,000件単位でPOSTし、リクエスト～レスポンスの応答時間を測定した。評価環境は表1に示すとおりである。また、データをPOSTするプログラムはRubyで実装した。

表1：評価環境

CPU	Intel® Core™ i5-3570 3.40[GHz]
Memory	16.0[GB]
SSD	MTFDDAK128MAM-1J1
OS	Windows 7 64bit
DB	PostgreSQL 9.3 64bit
その他	<ul style="list-style-type: none"> • Java 7 • Tomcat 7.0 • Ruby 2.0

5.2. 登録性能

図4は、1,000件単位のデータ登録した場合において、各テーブルの累計処理時間の推移を示したものである。テーブルAとテーブルBともに、登録レコード数の増加に関わらず、線形に推移し

ていることを確認できる。また、登録性能は、本評価環境において、テーブル当たり1.8[msec/件]であった。

4に示すトリガの実装により、任意の保存期間においてデータベースのストレージ容量を一定に保つ、テーブルパーティション運用の自動化実現を確認できた。これにより、テーブルパーティション運用のユーザ負荷を軽減できる。

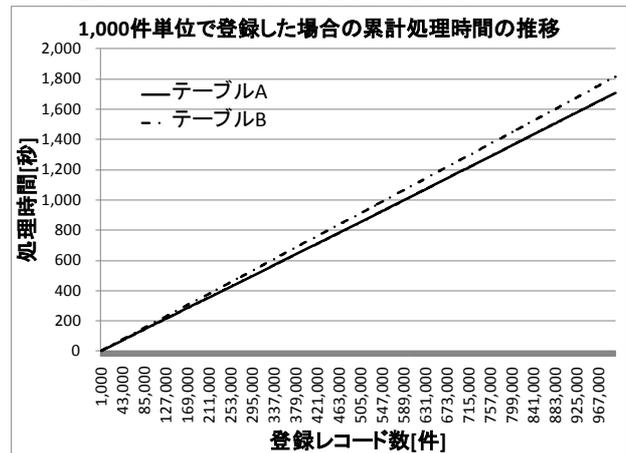


図4：データ登録における累計処理時間の推移

6. まとめ

本稿では、PostgreSQLとDBASによるシステム構成において、時系列データを対象とした場合に必要なテーブルパーティション運用の自動化について検討と評価を実施した。

検討の結果、各対象テーブルに2個のトリガをそれぞれ設定することで、パーティション運用の自動化を実現した。また、本実現方式にもとづく評価環境を構築して評価した結果、テーブル当たり1.8[msec/件]の登録性能を示し、線形に推移することを確認した。

今後は、テーブルごとに設定されたトリガにおける、同一判定処理を効率化することで、テーブル数が増加した場合の性能劣化に対応していく。

参考文献

- [1] 小杉,他,「マルチテナント対応データ切替制御に関する研究」, 2011, 第74回情報処理学会全国大会論文集,4H-5
- [2] 小杉,他,「マルチテナント型テナントアクセス制御方式に関する研究」, 2012, 第75回情報処理学会全国大会論文集,3H-1
- [3] 木幡,他,「大規模情報系システムにおける統合ID管理ソリューションの適用」, 2012,三菱電機技報, 86, No.7, 399~403
- [4] <http://lets.postgresql.jp/documents/technical/partitioning/2>