データベース統合エンジン

異なるデータベースの横断的な検索を容易に実現

スマートコミュニティでは，個別に構築された複数のシステムが管理するデータを横断的に活用する必要があります。しかし，個々のシステムでは互いに異なるデータベース（DB）が使われています。そこで東芝は，異なるDBを仮想的に一つのDBとみなして透過的にアクセスできるようにする，DB統合エンジンを開発しました。

このDB統合エンジンにより，異なるDBに対する横断検索が，DB問合せ言語（SQL）の1文で簡単に実現できるようになります。各DBに対応するドライバがDB間の差異を吸収することで，SQLの仕様の違いを感じさせない検索を実現しています。

異なるDBの横断的なアクセスへの要求と課題

スマートコミュニティでは，電気，ガス，交通など異なる分野の制御システムが保有するデータを横断的に活用して，インフラの統合的な管理と最適制御を目指しています。しかし，個々のシステムは異なるDBを使用しているため，それらへの横断的なアクセスが必要になります。

異なるDBを併用する場合，それぞれのSQLで個別に検索結果を取得し，それらのデータに対して，結合，並べ替え，及び集計などの検索演算をアプリケーション側で行う必要があります。つまり開発者は，個々のDBの仕組みや実装方法を学習したうえで，アプリケーション上でDBの内部演算と同等の演算プログラムを実装しなければなりません。

DB統合エンジンとは，このような複数のDBを横断したデータの検索演算を，標準的なSQLの1文で簡単に実現できるようにすることを目的としています（**図1**）。DB統合エンジンの課題として，統合対象のDBを容易に増やせる拡張性と高速化がありました。

これらの課題を解決するために開発した技術について以下に述べます。

SQLの仕様の差を吸収するドライバ

DB統合エンジンは，次の手順でSQL文を実行します。

(1) SQL文を解釈
(2) 個々のDBへの検索内容を決定
(3) 検索内容からDB独自のSQL文を生成
(4) DB独自のAPI（Application Programming Interface）でDBへ問い合わせて結果を取得
(5) 全DBから得た結果に対してDB統合エンジン上で検索演算を実施

この中で，個々のDBに依存した処理である(3)と(4)をドライバとして独立させることで，統合可能なDBを追加できるようにしました。

SQLは表形式のデータに対する問合せ言語ですが，表計算ソフトウェアや，テキストデータ，非構造データなどでもデータを表とみなすことができれば，ドライバを通じてそれらへアクセスできるようになります。現在，3種類のリレーショナルDBと1種類のNoSQL型DB(注1)をサポートしています。

DBへの検索要求最適化による高速化

アプリケーションは，DB統合エンジンを介してDBにアクセスすることで，オーバヘッドによる性能劣化が予想されます。性能のボトルネックは，データ通信量及びDB統合エンジン上での検索演算です。今回，これらの問題を解決するために，DBへの検索要求を最適化する処理技術を開発しました。

SQL文には，検索条件式やSQL関数という問合せ要素が含まれます。この問合せ要素をできるだけDB上で計算させることで，高速化を図りました（**図2**）。

そのために，問合せ要素が複数のDB上のデータを対象とするか，ただ一つのDB上のデータだけを対象とするかを判定します。ただ一つのDB上のデータが対象である場合，その問合せ要素をデータが格納されているDBへの問合せとして，検索演算を対象DBで実行します。また，複数DB上のデータを対象とする場合，対象データを取得したうえでDB統合エンジンが検索演算を実行します。

この最適化処理の結果として，可能なかぎり個々のDB上でデータを絞り込むことにより，DBとDB統合エンジン間のデータ転送量やDB統合エンジン上での演算が減ります。

アプリケーションプロセスへの組込みによる高速化

DB統合エンジンとしては，構成形態の異なるクライアント／サーバ型とライブラリ型を用意しています。アプリケーションに組み込めるライブラリ型は，更なる高速化が期待できます。

ライブラリ型では，クライアント／サーバ型で発生するアプリケーションとDB統合エンジン間の通信時間がなくなり，検索結果のデータ量が大きい場合にボトルネックとなる転送時間を削減します。

性能評価

次の三つの実装方法に対して，実行時間を比較しました（**図3**）。

(1) 検索要求を最適化していないDB統合エンジンを使用
(2) 検索要求を最適化したDB統合エンジンを使用
(3) 類似製品を使用

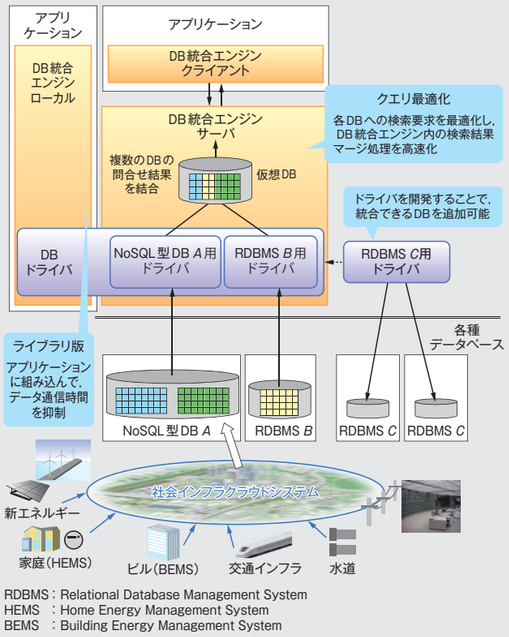
実験に使用したSQL文は，二つの異なるDB上にあるテーブルを結合するものです。一方のテーブルには約7,000行のレコードが格納されていて，その中から約70行を選択します。もう一方のテーブルには約2,000行のレコードが格納されていて，約30行を選択します。そして，それらを結合演算して最終的に6行の結果を生成します。

その結果，類似製品と比べて約1.5倍の高速で検索できることを確認しました。今回のように，条件式をDB上で計算させることでDBからの検索結果を絞れたことが，性能を発揮できた要因と考えています。

今後の展望

DB統合エンジン上での検索演算の性能改善や，SQL文の最適化により，いっそうの高速化を検討しています。更に，分散トランザクション機能(注2)の導入やドライバの拡充を行い，適用範囲の拡大を目指します。

片山　大河
ソフトウェア技術センター
先端ソフトウェア開発担当



RDBMS：Relational Database Management System
HEMS　：Home Energy Management System
BEMS　：Building Energy Management System

図1．DB統合エンジンのモジュール構成 — アプリケーションから受け付けたSQLを解析して，各DBへの検索要求に分割します。そして，ドライバを使ってDBから結果を取得し，最終的に一つの検索結果を返します。

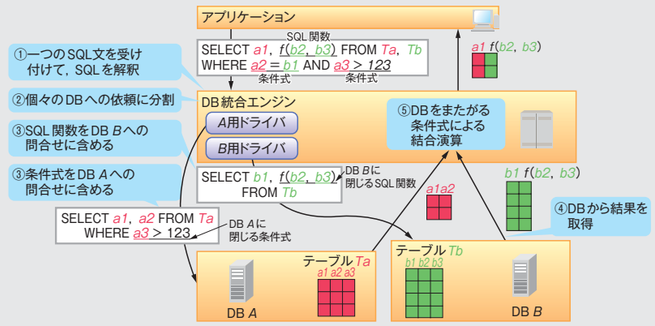


図2．DBへの検索要求の例 — DB上で計算できるSQL関数や検索条件式がある場合，それを含んだSQL文をDBで実行します。

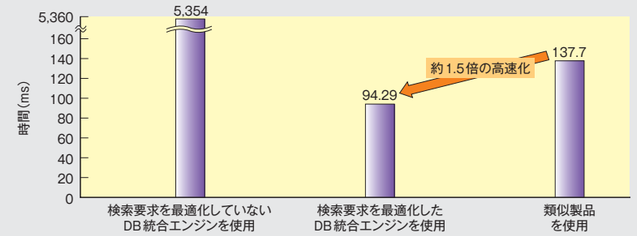


図3．検索性能の比較 — 約7,000行のテーブルと約2,000行のテーブルとを結合するSQLの実行に要する時間を比較しました。その結果，検索要求の最適化により約57倍の高速化を達成し，類似製品と比べ約1.5倍の高速で検索することができました。

(注1) 関係モデルに基づかずに構築されたDBの総称。
(注2) DBをまたがるトランザクション管理機能。