大量データを高速に処理する「InfoFrame DataBooster」

川畠 輝聖・濱田 光保
田村 稔・白馬 智博

要　旨

情報爆発の時代となり、近年では大量データに対するさまざまな分析ニーズが出てきています。一方で、ハードウェアの進化から搭載できるメモリ量は飛躍的に増大しています。「InfoFrame DataBooster」はカラムストアをベースとしたインメモリデータ処理技術により、大量データを高速処理するニーズに応えてきました。本稿では一般的なRDBとInfoFrame DataBoosterとの違いや、最新バージョンを搭載したSQLインタフェース開発の背景と開発方法、InfoFrame DataBoosterの利用方法、適用領域（採用事例）について紹介します。

キーワード

●インメモリ ●データベース ●高速バッチ処理 ●カラムストア ●SQL

1. はじめに

グリーンIT推進協議会によると、2025年、社会に流通する情報量は2006年の200倍になると言われています。これは、センサや機器などが生成するデータ、企業内における取引データやログデータといった構造化データ、インターネットやテレビ・ラジオなどの映像・音声・画像といった非構造化データが急激に増加してきているからです。しかし、これら流通しているデータはまだまだ十分に利用されていない状況にあります。今後は、このような大量のデータからいかに新たな価値を創造できるかが課題となってきています。

価値を創造するために、さまざまなタイプの膨大なデータを高速に加工、分析する必要性が飛躍的に増大してきます。NECではビッグデータに対するデータ処理製品群を三層に分けており、それぞれ「分析・メディア解析層」「収集・統合層」「基盤層」と定義しています（**図1**）。

InfoFrame DataBooster（以下、DataBooster）は収集・統合層、そのなかでもインメモリ型の列指向（カラムストア）DBに位置付けられる製品です。なぜこのようなインメモリ型のデータ処理製品が台頭してきたのかを、第2章で説明します。

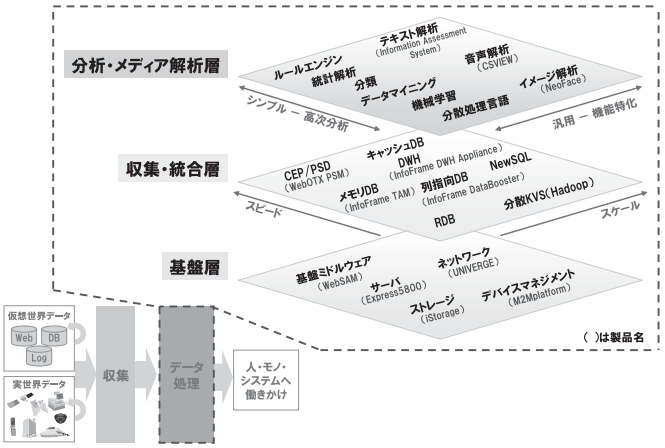

図1　ビッグデータに対するデータ処理製品群

2. インメモリデータ処理登場の背景

データ量が増大したので処理時間を短縮したい、というニーズに対してはCPUを増強する手法もありますが、データ処理においてはI/Oに掛かる時間を短縮する方が効果的です。そこでデータをストレージに保持するのではなく、メモリに保持することでデータ処理時間を短縮する手法が出てきました。ただ、メモリはストレージに比べ高速ですが高価であるというコスト的なデメリットがありました。

しかし、2010年以降、1サーバに搭載されるメモリ量が急激に増加し、コスト単価もより安価になってきました（**図2**）。2012年6月現在では最大2TBのメモリを搭載するサーバ（Express5800/A1080）が存在します。これにより、ストレージとメモリを併用することなく、メモリだけでデータを保持し

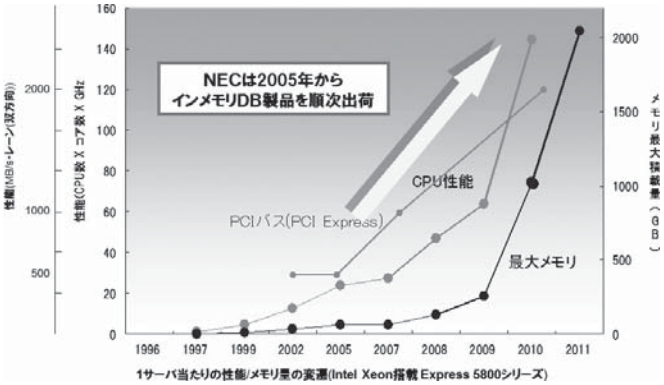

図2　メモリ量の増大とコストの低減

て処理できる環境ができ、データ処理におけるI/O処理が非常に短くなりました。

今後も更にメモリを搭載したサーバが出荷される予定であり、インメモリデータ処理がますます注目を浴びていくことでしょう。

3. 適用領域の異なるデータ処理技術

1980年代から広まった、リレーショナルデータベース（以下、RDB）はOLTP（On-Line Transaction Processing）といった更新処理や、OLAP（On-Line Analytical Processing）といった集計処理などのすべてのデータ処理を担っています。ところが、近年データ量が飛躍的に増加したことで、RDBだけではニーズに応えることが難しくなってきました。そこでスケールアウト性が高く大量更新に向いているKVS（Key Value Store）や大量データの分析が得意なMapReduceなどさまざまな手法に注目が集まっています。そのなかの1つが、カラムストアデータモデル（以下、カラムストア）という手法です。

カラムストアは通常のリレーショナルデータモデル（以下、リレーショナル）のように表形式でデータを格納します。異なる点は、リレーショナルでは基本的にレコード（行）単位にデータを格納してくのに対し、カラム（列）の要素をまとめて格納することにあります。

これは大きく2つのメリットを生みます。1つ目は参照の高速化です。分かりやすく説明するためにインデックスを無視すると、リレーショナルでは該当レコードに到達するまでのレコードをすべて読み込む必要があります。カラムストアではデータ

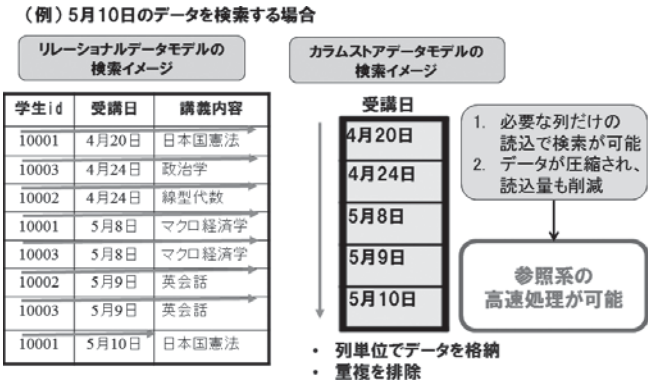

図3　リレーショナルとカラムストアの違い

が列単位のため、必要な列だけを読み込めばよいのです。

2つ目は、格納データ量の削減です。カラムストアではデータを列単位で格納しているため、一般的には重複が多く、容易にデータの圧縮が可能になります。例えば**図3**では、「受講日」が5月10日のデータの有無を確認するために8件のレコードを読み込む必要がありますが、カラムストアでは受講日列の単位でデータが格納されているため、「学生id」や「講義内容」のデータを読み込む必要はありません。また、この圧縮効果は更なる参照の高速化にも寄与します。

4. InfoFrame DataBoosterの特長

このカラムストアを更にインメモリに最適化した技術を採用している製品がDataBoosterです。

図4のように表形式のデータを独自構造（順序集合のOrdset、値番号VNo、値リストVLという3つのベクトル成分）に分解し、値リストはソートした形でメモリ上に展開しているため、使用メモリ量を削減したうえで、データ検索を高速に行うことができます。更に、従来のレコード単位の逐次処理をカラム単位の一括処理で行うことで、処理の高速化を実現できます。

このメモリアクセスに特化した技術と大容量化されたメモリにより、集計やデータクレンジングのような一括に行う処理については、非常に高速な性能を発揮します。

例えば夜間バッチでのデータマート作成において、データ量の増加に伴い朝までにバッチ処理が終わらず業務に影響を及ぼしてしまうケースなどでは、DataBoosterによる高速化で時間内に処理が終わるだけでなく、バッチ処理以外のバックアップなどの業務にも時間を回すことが可能になります。

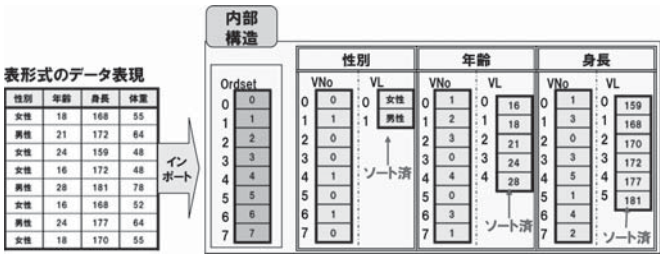

図4　DataBoosterの内部データ構造

5. SQLインタフェースの開発と実装

5.1 PostgreSQLの利用

DataBoosterの標準的なAPIはC言語とJavaであり、各APIが1つの機能、例えば、ソート、集計、検索、ジョイン、計算を実行するため、慣れることができればプログラム作成は簡単であり、生産性も高いものです。

ただ、DataBoosterの製品としての位置付けがデータベースの補完ということもあり、この製品を使うユーザーにはデータベースシステムの経験者が多いため、DataBooster V4.1（2011年11月出荷）において、JDBC APIを使ってSQL文を実行するインタフェースを追加しました。

SQLインタフェースの開発において、SQL文の処理部にはPostgreSQLを利用しています。PostgreSQLの利用は字句解析や構文解析だけでなく、それ以降の処理である、実行プラン作成、実行部においても利用しています。そのため、DataBooster機能だけでは実行できないSQL文、例えばDataBoosterにはない関数などは、PostgreSQLを利用して処理を行っています。

DataBooster SQLは、PostgreSQLがサポートしている、ほとんどのSELECT（参照）、INSERT（挿入）、UPDATE（更新）、DELETE（削除）文が実行可能です。そのほかPREPARE、EXECUTE、EXPLAINなども実行できます。

処理対象となるテーブルは、処理（参照、更新など）を始める前に、テーブルの定義やデータのメモリ展開をしておく必要があります。これには、DataBooster独自仕様のCOPY文を使用し、CSV形式の外部ファイルなどから入力することが可能です。なお、一般的なRDBと異なり、JDBC（Java DataBase Connectivity）でのアクセスはアプリケーションにDataBoosterライブラリをリンクする形式を採り、すべてが同じプロセスで動作します。

5.2 DataBooster APIの特徴と課題

DataBooster APIは一般的なSQLに比べて、検索条件や計算式に記述できる式の自由度が小さいため、SQLのような自由度の高い検索条件を処理する場合は、複数の処理を組み合わせて実現しています。例えば、WHERE 列A＝列B＋1 という検索条件があった場合は、まず、新たに列Xを追加し、この列に 列B＋1 を格納します。更に、列Aと列Xが同じなら1、異なるなら0を格納します。最後に、列Xに対して列X＝1という条件で検索を実行します。このようにして、SQLの字句解析や構文解析、プラン作成などを省略してなるべくDataBoosterで処理することで、その高速性を発揮できるようにしています。

また、DataBoosterではジョインの際、列転送という機能があります。これは、ジョインされた一方の表の列をもう一方の表に追加する機能です。この機能を使うことによりジョインを高速に行えるようになりますが、どの列を転送するのか、また転送するだけで後の処理ができるのかを判断することは困難です。将来は、これらの処理を効率的に実行できるように実装することで、性能とユーザーの使い勝手の向上を図る予定です。

SQLインタフェースを追加することでSQL言語に慣れたアプリケーション開発者による生産性は向上できるようになりました。しかし、DataBoosterの標準APIはC言語及びJavaであり、十分な高速性の確保や効率的なプログラムを作成する場合は、C/Java APIの使用をお勧めします。C/Java APIでは、機能ごとに処理できるため、プログラム設計、中間結果の再利用、性能見積もり・チューニング、デバッグなどが容易に行えます。

6. InfoFrame DataBoosterの活用イメージ

ここで、これらの技術の活用方法を2つの例で紹介します。

企業が社外向けに公開しているWebページに対して、ユーザーがどのようにアクセスしたか把握することは、自社製品やサービスの向上に欠かせません。人気企業ではアクセスログのデータ量が膨大なため、データを収集して分析できる形に加工し、分析用データベース（RDB）に格納するまでには長時間を要します。そのため次の2点が大きな課題となります。

1) 分析結果をタイムリーに参照できない
2) データ加工速度を上げるため、並列処理用にサーバの台数やデータ加工用ソフトウェアライセンスを際限無く追加購入しなければならない

DataBoosterは、これらの課題を解決します。DataBoosterの高速なデータ加工により、従来に比べて分析用RDBへ格納する時間を何倍も短縮できるため（**図5**の例では数時間から15分単位）、ユーザーはより鮮度の高い分析データを入手することが可能になります。また、同じデータ加工を行う場合、ETL（Extract/Transform/Load）などの他のデータ加工が可能なソフトウェアに比べて少ないコア数で動作します。そのため、同じコア数を搭載したサーバを使用する場合は少ないサーバ台数でよく、処理量が増加した場合でもサーバを追加する回数は、他のデータ加工が可能なソフトウェアを使用した場合に比べて少ないというメリットがあります。

次に、DataBoosterを直接データ分析ツールとして使用する例を紹介します。

例えばダイレクトメールや製品カタログをエンドユーザーに送付する場合、むやみに送付しては費用と時間の無駄になります。また、エンドユーザーにまったく当てはまらない条件の製品に関する情報を送付してしまった場合、企業イメージを損なう可能性もあります。そのため一般的には、どこにアプローチすれば最も効果的かを企業の販売データから事前に確認し、ダイレクトメールや製品カタログを送付します。ただし、ここに課題があります。蓄積された販売データが膨大な場合、分析用に抽出するだけで多大な時間が掛かります。更に、抽出したデータを年齢や地域、性別、過去の購入品目などで分析を繰り返すと、その結果の入手に時間を費やすことになり、場合によっては時間が掛かりすぎて分析をあきらめてしまうこともあります。DataBoosterを導入した場合（**図6**）、次のようなメリットが生まれます。

- データの検索や分析結果を数秒で入手可能
- 従来に比べて更に細かい条件で分析可能

企業（営業）は、正しいエンドユーザーに、タイムリーにアプローチすることが可能になり、勝率を上げ、無駄な費用と時間を削減することができます。

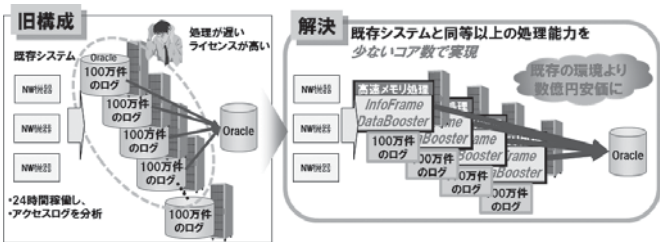

図5　例1　アクセスログ集計システム

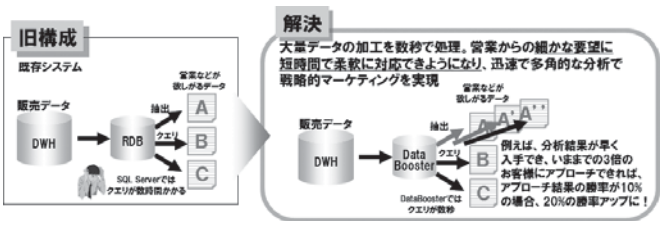

図6　例2　データ分析システム

7. おわりに

データ処理は、用途により対応するソフトウェアを選別する時代がきました。スピーディな処理を実現するDataBoosterは、今後も増え続けるデータ処理時間の解決策として更なる活躍が期待されます。

参考文献

1) グリーンIT推進協議会
https://www.greenit-pc.jp/about/

執筆者プロフィール

川畠 輝聖
ITソフトウェア事業本部
第三ITソフトウェア事業部
主任

濱田 光保
ITソフトウェア事業本部
第三ITソフトウェア事業部
マネージャー

田村 稔
ITソフトウェア事業本部
第三ITソフトウェア事業部
マネージャー

白馬 智博
ITソフトウェア事業本部
第三ITソフトウェア事業部
主任

関連URL

InfoFrame DataBooster/DataBooster Lite:
http://www.nec.co.jp/databooster/

NEC 技報のご案内

NEC技報の論文をご覧いただきありがとうございます。
ご興味がありましたら、関連する他の論文もご一読ください。

NEC技報WEBサイトはこちら

NEC技報(日本語)　　NEC Technical Journal(英語)

Vol.65 No.2　ビッグデータ活用を支える基盤技術・ソリューション特集

ビッグデータ活用を支える基盤技術・ソリューション特集によせて
ビッグデータを価値に変えるNECのITインフラ

◇ 特集論文

データ管理 / 処理基盤

超高速データ分析プラットフォーム「InfoFrame DWH Appliance」
SDN技術で通信フローを制御する「UNIVERGE PFシリーズ」
大量データをリアルタイムに処理する「InfoFrame Table Access Method」
大量データを高速に処理する「InfoFrame DataBooster」
ビッグデータの活用に最適なスケールアウト型新データベース「InfoFrame Relational Store」
高い信頼性と拡張性を実現したExpress5800/スケーラブルHAサーバ
大規模データ処理に対するOSS Hadoopの活用
大容量・高信頼グリッドストレージiStorage HSシリーズ(HYDRAstor)

データ分析基盤

ファイルサーバのデータ整理・活用を支援する「Information Assessment System」
超大規模バイオメトリック認証システムとその実現
WebSAMの分析技術と応用例～インバリアント分析の特長と適用領域～

データ収集基盤

スマートな社会を実現するM2Mとビッグデータ
微小な振動を検知する超高感度振動センサ技術開発とその応用

ビッグデータ処理を支える先進技術

多次元範囲検索を可能とするキーバリューストア「MD-HBase」
高倍率・高精細を実現する事例ベースの学習型超解像方式
ビッグデータ活用のためのテキスト分析技術
ビッグデータ時代の最先端データマイニング
ジオタグ付きデータをクラウドでスケーラブルに処理するジオフェンシングシステム
柔軟性と高性能を備えたビッグデータ・ストリーム分析プラットフォーム「Blockmon」とその使用事例

◇ 普通論文

地デジTVを活用した「まちづくりコミュニティ形成支援システム」

◇ NEC Information

NEWS

スケールアウト型新データベース「InfoFrame Relational Store」が2つの賞を受賞

Vol.65 No.2
(2012年9月)

特集TOP