

## 検索結果の高速転送方式と評価

秩父 かおり 佐藤 重雄 早川 孝之

三菱電機株式会社 情報技術総合研究所

### 1. はじめに

データウェアハウス (DWH) は、企業において、日々の業務活動で収集したデータを蓄積し経営戦略立案のために利用することが一般的であった。しかし近年は、個人情報や企業機密の漏洩防止のためにログを蓄積し分析することによりセキュリティを高めるといった使い方もされ、利用目的が高度化し適用範囲も拡大している。

DWH の適用拡大に伴い、DWH を利用するためのアプリケーションも自製だけでなく、他社製のものとの連携する必要も出てきている。その場合に使われるのが、汎用的な接続インタフェース (ODBC、JDBC など) である。(図1参照)

本論文では、汎用的な接続インタフェースにおいて DWH の検索結果を転送する方式を改良したので、その方式と性能評価について述べる。

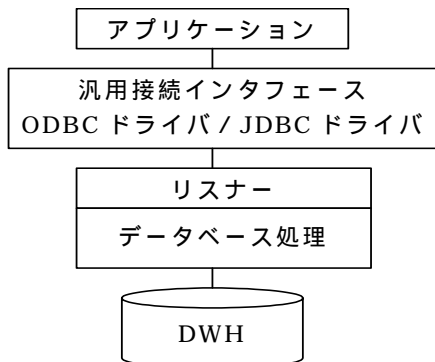


図1. 構成図

### 2. 検索結果の転送方式

DWH で検索した結果をクライアントへ転送する方法には、レコードを1行ずつ転送する方法と一括して転送する方法 (一括転送方式) がある。

1行ずつ転送する方法は、単純な方法であるが、1行ずつ転送することのオーバーヘッドが大きく、一括転送方式に比べて、転送性能が大きく劣ることが課題となる。

一方、一括転送方式としては、例えば、DWH サーバで検索結果を共有ファイルに格納しクライアントからアクセスして一括転送するといった方法がある。この方法は、1行ずつ取得する方法に比べて高速化を実現できるが、プラットフォーム環境に依存するため、汎用性を持たせることが難しいという問題がある。

さらに、DWH にログを格納する場合、検索結果の項目 (列) によってはスペースが多く入る可能性があるため、通信部分でそれを吸収し、転送性能を向上させることも課題である。

### 3. 転送方式の高速化

2. で述べた問題の解決法として、次の実装を試みる。

#### (1) 複数行の結果セットの一括転送

アプリケーションで指定されたフェッチサイズ (行数) を上限として、一度に複数行の結果を転送する。そのためのメッセージ形式を新たに規定する。アプリケーションで指定される値とは別に、通信部においても一度に送受信する最大行数の既定値を定め、アプリケーションで指定されたフェッチサイズがこれを超える場合は、通信部が既定値の行数で区切って複数回メッセージを送受信する。この通信部のフェッチサイズ最大行数は初期化ファイルで変更可能とする。

#### (2) 結果セットの文字列型データのスペース処理

転送量を減らすために、結果セットに含まれる固定長文字列型データの後ろスペース文字を削除してクライアントに送信する。データ長が短いとオーバーヘッドが大きくなるだけでデータ量削減の効果が小さいので、データ長により対象とするか否かを判断する。通信データに、スペースが削除されているか否かを通知するデータを設ける。クライアントの通信部では、元のデータ長までスペースパディングして上位に返す。

### 4. 性能評価

本方式を当社製 DWH システムに実装し、性能評価を行った。ここでは特に JDBC ドライバの結

果について報告する。

表 1 に測定環境を示す。表 2 には測定に使用したデータを示す。レコード件数、列数の値は DWH に格納されている元データではなく、実際に転送される検索結果の値を示す。レコード件数、列数、データ型により性能が変化すると考えられるため、これらのパラメータに幅を持たせるモデルを設定した。

表 1. 測定環境

サーバ	
モデル	FT8600/440Ra
プロセッサ	XeonMP 2GHz × 4
メモリ	2GB
OS	Windows Server 2003 Enterprise Edition SP1
クライアント	
モデル	apricot CX240
プロセッサ	Pentium4 1.6GHz
メモリ	640MB
OS	Windows XP Professional SP2
Java	J2SE v 1.4.2_08 JRE
ネットワーク	
測定マシンを 100base ハブ経由で直結。	

表 2. 測定データ

モデル	レコード件数	列数	レコード長	備考
1	237808	21	684	スペース処理の対象データ有り
2	45664	2	12	数値型が整数型
3	45664	2	16	数値型が実数型
4	50809	3	116	スペース処理の対象データ有り
5	100000	999	8192	スペース処理の対象データ有り

3. で示した方式の実装前後の通信性能向上度を図 2 に示す。このときアプリケーションで指定するフェッチサイズは、検索結果のレコード件数より大きいサイズを指定するようにした。実装前の性能を 1 とした場合、2~17.5 倍の性能を出すことができた。効果が大きいのは、以下の場合である。

- (a)レコード件数が多い
- (b)列数が少ない
- (c)数値型データの場合整数型である

文字列型データのスペース処理については、多少のオーバーヘッドはあると思われるものの性能向上に貢献していると思判断できる。

更に、フェッチサイズの最適値を探るために、フェッチサイズと検索結果のレコード件数や取

得列数の関連性を測定した(図 3、図 4)。どのケースにおいても概ねよい性能が得られるのはフェッチサイズが 4096 件のときである。性能が劣化するケースに関しては、測定用クライアントマシンのメモリが不十分なためディスクスワップが起きていると考えられる。

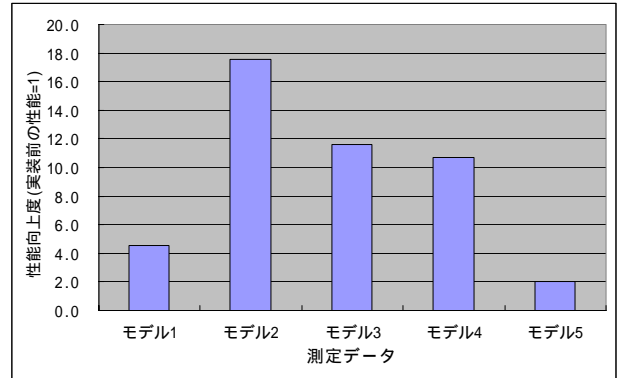


図 2. 性能向上度

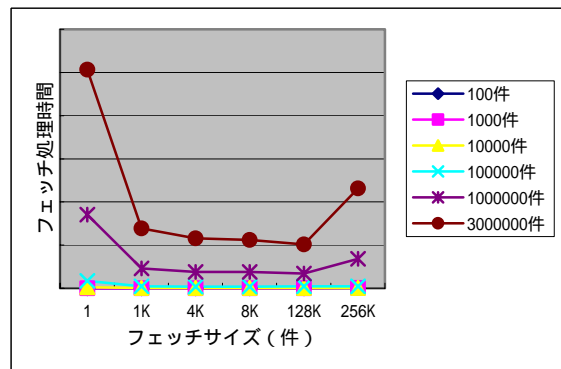


図 3. 結果件数特性

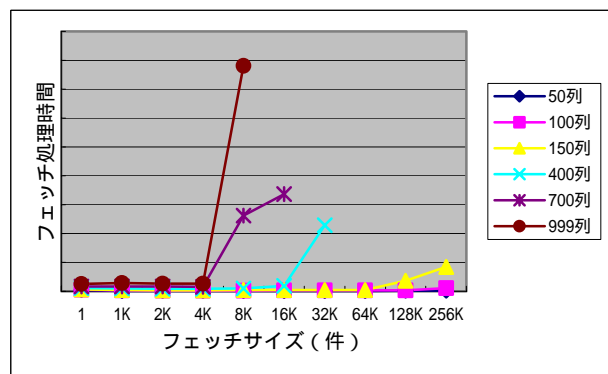


図 4. 結果列数特性

## 5. おわりに

本稿では、DWH を検索した結果を高速に転送する方式について述べ、その性能評価結果を報告した。今後は更なる高速化のために、フェッチの先読み等の方式を検討していく予定である。