

現実世界のベンチマーク： Cache vs Oracle データウェアハウス・アプリケーション

Mark Ramsay

Senior Sales Engineer

InterSystems Corporation

要約

実際に使われているアプリケーションのベンチマークを行った。

それは、データウェアハウスにデータをロードして、後にそのデータを分析するものである。ベンチマークを行うのに、Oracle ベースのアプリケーション・モジュールの1つを Cache ObjectScript に複製した。40 人/時間ほどの作業でオリジナルモジュールの機能が Cache に複製できた。

タスク完了時間は、Cache ベースのモジュールを走らせて測定し既存データとの比較をした。その結果、**Cache アプリケーションは、5 倍ほど速かった**。データのロードとタスクのフィルタリングにわずか 286 分かっただけである。それに対し Oracle では、1441 分であった。

序説

大手電力供給会社の Meralco コーポレーションは、企業のデータウェアハウスとして Oracle を使用している。毎月、巨大なテキストベースのデータをデータウェアハウスにロードし、それを分析に使っていた。データウェアハウスのオペレーションを改善する努力の中、Meralco コーポレーションより高速 DBMS である Caché の代理店である Digital Dimentions 社を介して連絡があった。インターシステムズ社は、彼らのデータウェアハウス ETL (Extract 抽出、Transform 変換、Load ロード) アプリケーションであるフィールド・オーダー・モジュールを Caché に複製し、Oracle と Caché のパフォーマンス比較テストを行った。

フィールド・オーダー・モジュール

Meralco 社のフィールド・オーダー・モジュールは、Oracle 8i ベースで、現場で生成されたテキストファイルを入力として2つのテーブル(フィールドオーダー次元テーブルとオーダー実績テーブル)を生成するように設計されている。これらの2つのテーブルは、後にデータ分析とレポートに

使用される。このタスクを実行するのに、次のオペレーションを行う。

- 1) 現場で生成された複数のテキストファイル (ordenes.txt と gcahorde.txt) が2つの実行テーブル(ORDENS と GCAHORDE)にロードされる。Oracle ベースのアプリケーションでは、Oracle SQL Loader を使って実行する。
- 2) GCAHORDE テーブルは、フィルタにかけられ GCAHORDE_FILTERED テーブルが生成される。
- 3) Oracle パッケージ、ストアド・プロシジャを使って GCAHORDE_FILTERED と ORDENES からフィールド次元テーブル DD_FIELD_OEDER が生成される。
- 4) Oracle パッケージ、ストアド・プロシジャを使って、GCAHORDE_FILTERED と ORDENES からフィールドオーダー実績テーブル DR_FIELD_ORDER が生成される。

Cache にフィールド・オーダー・モジュールの複製

Caché は、ODBC と JDBC アクセスが可能で、多くのリレーショナル・アプリケーションは、最小限の変更で Caché データベース上で稼動することが可能である。ただし例外として、ストアド・プロシジャに大変依存する場合がある。Meralco のフィールド・オーダー・モジュールもそれに該当する。リレーショナル・アプリケーションを Caché に移行するとき、ストアド・プロシジャは、Caché ObjectScript のルーチンとして実装する必要がある。そのため Caché ObjectScript にフィールド・オーダー・モジュール機能を複製する方が、既存のコードを採用する方法より速いだろうという結論になった。

このアプローチは、付加の利点があった。それは、新しい Caché ベースのモジュールがインポートの際、テキストファイルを事前にフィルタする Caché の機能を利用することができる点だ。

ファイルのデータは、データベースにロードされる間にフィルタされるので、Cache ベースのモジュールでは、別のフィルタリング・ステップを行う必要がなかった。

Maralco 社の助言により、インターシステムズ社はフィールド・オーダ・モジュールの Cache 版を作成した。このタスクは、Cache をよく知る開発者で、約 40 人/時間の作業量であった。

結果

フィールド・オーダ・モジュールの Cache と Oracle 8i バージョンは、同じセットのテキストベースデータファイルで開始した。この時間は、各バージョンからさまざまなタスクが完了するまでの時間を計った。Cache の実行時間は、タスクによっては 23 倍 Oracle より速かった。その結果の要約は以下の通りである

タスク	Oracle		Cache		実効時間レート (Oracle : Cache)
	経過時間	レコード数	経過時間	レコード数	
ロード ORDENS	138 分	927,857	6 分	927,857	23 : 1
ロード&フィルタ GCAHRDE	245 分	90,349*	23 分	90,349	10 : 1
FO 生成 寸法テーブル	168 分	1,018,216	24 分	1,018,216	7 : 1
FO 生成 ファクトテーブル	890 分	1,018,216	233 分	1,018,216	4 : 1
ロードとフィルタまでの時間	1441 分		286 分		5 : 1

* 13,396,510 ソースレコードがフィルタされ、90,349 レコードをロード

結論

実世界でのデータウェアハウス・アプリケーションのベンチマークでは、データローディングとフィルタリングオペレーションを行うとき、Cache は、Oracle よりも 5 倍のスピードがでた。

40 人時だけの作業で Oracle ベースのアプリケーションが Cache に複製することができた。