



まとめ

Visual Basic.NET + Oracle接続方法大検証

初音 玲 HATSUNE, Akira

Visual Basic.NETとOracle Databaseを接続するには、

- Oracle Objects for OLE (oo4o)
- OLE DB .NET Data Provider
- Oracle Data Provider for .NET

のそれぞれを利用した3種類の方法が

あり、各々のプログラミングスタイルの相違についても紹介してきた。しかし、実際に、Visual Basic.NETでOracleクライアントアプリケーションを作成しようとしている読者の中には、「では、一体どの接続方法を使うべきなのか」という疑問が湧いてきた方もいるだろう。

どの接続方法を使うべきなのか

一般的なSQL文を使う限りにおいては、oo4o、OLE DB .NETデータプロバイダおよびOracle Data Provider for .NETの3者に優劣はないように思われる。そうすると、この3者を比較検討するためには、もうひとつの評価基準である「処理速度」という面に注目しなければならないだろう。

処理速度の測定

そこで、図1のような性能測定サンプルを作成し、以下の項目について処理速度を計測してみた。

Check1 初回ログオン時間

ログオン処理はプログラム起動直後に行なうことが多い。そこで、性能測定サンプルを起動して、すぐに1回だけ

本稿で前提となるもの

O S	Windows 2000 Server (SP3) 以降
開発環境	Visual Studio.NET
	.NET Framework 1.0.3705.288 (SP2)
	Visual Basic .NET Compiler 7.00.9466
	Internet Explorer 6.0.2800.1106
	Oracle 9i Client R9.2.0.1.0
	Oracle 9i Database Enterprise Edition R9.0.1.1.1



この記事で解説したサンプルプログラムは、付録CD-ROMの¥DMAG¥F01_06フォルダ以下に収録しています。

¥PERFORMANCE：性能チェック

ログオンを行ない、その時間を測定する。なお、測定区間には、データベース接続用オブジェクトの生成から接続処理完了までとする。

Check2 100回連続ログオン時間

初回ログオンの場合、.NET FrameworkによるプログラムのJITコンパイル時間なども含まれているので、初回ログオン後に100回連続してログオンとログオフを繰り返した時間を測定し、初回との差異を確認する。

Check3 100回連続参照時間

ログオン後、100回連続してレコードを検索して、プライマリーキー以外を参照したときの時間を測定する。なお、測定区間は、検索用オブジェクトの生成から解放までを100回ループするループブロックの開始直前から終了直後とする。

検索時にはプライマリーキーを使うので、それ以外を参照することで、検索時にプライマリーキーの値がキャッシュされていても性能が向上しないように配慮している。

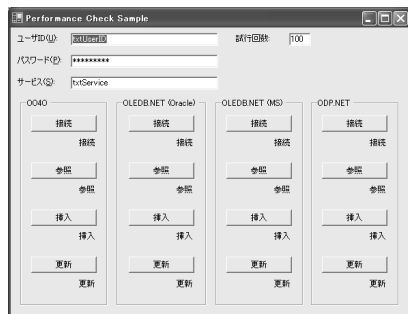
Check4 100レコード連続挿入時間

ログオン後、100回連続してレコード挿入したときの時間を測定する。なお、測定区間は、レコード挿入用オブジェクトの生成から解放までを100回ループするループブロックの開始直前から終了直後とする。

Check5 100レコード連続更新時間

ログオン後、連続して100レコードのプライマリーキー以外を更新したとき

図1：性能チェック（サンプルPerformance）



の時間を測定する。なお、測定区間は、レコード更新用オブジェクトの生成から解放までを100回ループするループブロックの開始直前から終了直後とする。

このようなテスト項目に基づいてテスト結果は表1ようになる。表1は、oo4oの結果を1とした相対値として比較している。これは、実際の処理時間はネットワーク環境やハードスペックに左右されるためあまり意味がない値だからだ。

テスト結果の考察

表1の結果から導き出せる重要ポイントは以下の通りだ。

重要ポイント①

表には記載していないがoo4oの1回目のログオン処理時間と2回目以降の

100回ログオン処理時間はあまり差がない。つまり、2回目以降は1回目よりも100倍近く速いということだ。このことからわかるのは、1回目の処理時間の大半はJITコンパイルの時間で、今回のサンプル実行環境では、15秒程度の処理時間がJITコンパイル時間であった。

重要ポイント②

2回目以降の100回ログオンの時間は、COMコンポーネントであるoo4oと.NETコンポーネントであるその他のものとではかなり開きがある。しかし、oo4o以外については、連続ログオンを行なった場合、Oracle Databaseとのセッションが複数残っていた。これは、OLE DB .NETデータプロバイダがADO.NETの接続プール機能を自動的に利用している影響からである。そこで、

```
odbConn.Open()
odbConn.Dispose()
```

というコードを、

```
odbConn.Open()
odbConn.Close()
odbConn.ReleaseObjectPool()
odbConn.Dispose()
```

のように書き換えて、接続プールへの

表1：接続方法別の速度差（oo4oの速度を1）

処理（100BASE-T 100Mbps）	oo4o	OLEDB .NET Data Provider		ODP.NET
		Oracle	Microsoft	
①プログラム起動直後のログオン	1.00	0.98	1.00	1.02
②2回目のログオン	1.00	543.80	543.80	1087.60
③プライマリーキー以外の参照	1.00	1.61	1.72	2.11
④挿入	1.00	1.02	0.79	1.11
⑤プライマリーキー以外の更新	1.00	1.14	0.63	1.50

※数値が大きいかいほど、oo4oと比べて速度が速いことを表わす。

