

.NET Framework

なにを使うか、どう使えるのか アイデアノート

第13回

秋月巖ソリューション事務所
秋月 巖 AKIZUKI, Iwao
<http://www.akizuki.co.jp>

SQL Serverの サーバーカーソルを使用する 自作クラス

Technology Tools

- Visual Basic .NET
- Visual C# .NET
- SQL Server 2000
- Oracle 9i
- Access 2002
- ASP.NET
- Internet Information Services
- Other:

Level



Samples

・この記事で取り上げたソースコードおよびサンプルプログラムは、付録CD-ROMの¥DOTNET¥NOTEディレクトリに収録しています。

¥UDAS
UDAsクラスのコード

¥WINDOWSAPPLICATION1
UDAsクラスの利用例

ADO.NETで サーバーカーソルを使う

本連載の第3回～第5回（2004年1月号～3月号）で、UDAクラスというデータアクセス用のコンポーネントを作成して紹介した。このコンポーネントは、ADO.NETのSQL Server用のクラスとOLE DB用のクラス、そしてDataReaderクラスとDataSetクラスのプログラミング記述の差を吸収するADO.NETのラッパークラスだった。

今回紹介する“UDAsクラス”は、ADO.NETにサーバーカーソルの機能を追加提供するクラスである。使用方法はADOを模したUDAに準拠している。ご存知のようにADO.NETではサーバーカーソルがサポートされなくなった。ADOでは、主力がサーバーカーソルだったのと比較すると、大きな変更である。ちなみにADOのクライアントカーソルサポートには不備な点があっ

た。それについては、もう随分前にVisual Basicマガジンに書いた^[注1]。ADOで、ひとつの接続で作成した複数の結果セットを表示用のコンポーネントに連結するには、Data Environmentコンポーネントを使用する必要があり、このコンポーネントには問題があったのである。一方、ADOのサーバーカーソルは確かに高速だった。SQL Serverで測定した結果、ほとんどすべてのパターンで動的サーバーカーソルがよい結果を出した。通常、高機能と高性能はトレードオフの関係にあるので、高機能な動的サーバーカーソルが優れた結果を出すのは意外だった。

なぜ、 サーバーカーソルか？

とはいえ、私自身はクライアント

注1)「新・SQL Server解体新書」第3回～5回（Visual Basicマガジン2001年7月号～9月号掲載）

トカーソルの強力な支持者なので、ADO.NETが強力なクライアントカーソルをサポートしたことに肯定的である。今日のようにサーバーとクライアントのPCの性能に大差がない場合、できる処理はクライアントで行なったほうが望ましいと考えるからである。それにラウンドトリップの回数が少ないほうがシステムの不安定要因が減る。

では、なぜ、サーバーカーソルに対応するためのクラスを提供するのかといえば、大きい結果セットを取得するようなクエリを実行した場合、最初にすべての結果セットデータをダウンロードしてしまうクライアントカーソルでは、ネットワークトラフィックが増加し、結果としてパフォーマンスが劣化するからである。もちろん、結果セットの行数が適切になるようにクエリとテーブルを設計することが最優先なのはいまでもない。しかし、ユーザーのキーワード入力による検索のように、結果セットの行数を想定できないような場面も多々ある。

サーバーカーソルか DataReaderか?

このような場面にADO.NETで通常対応するには、DataReaderクラスを使用する。DataReaderクラスは内部的にサーバーカーソルは使用せずにデフォルトの結果セットを使用するが、実際には結果セットの行数が多くても、Readメソッドを使ってデータの読み取りを行わない限りデータはオブジェクトにロードはされない。私はこれを実験してみたが、そのときはWebアプリケーションでの動作を前提にしていたため、DataReaderオブジェクトとSQL Serverは同じマシンに配置した。だから実際にトラフィックが少ないか否かの確認はないのだが、少なくとも、巨大な結果セットを返すようなクエリでも、DataSetオブジェクトとは異なり、DataReaderオブジェクトは最初の行の結果を瞬時に返した。だから、当然、結果セットの行数が多くても、読み取る（表示する）行数が少ないのなら、ネットワークトラフィックも少ないと予想するが、これについては確認していない。DataReaderオブジェクトは、Webアプリケーションには最適

だが、Windowsアプリケーションに使用する場合、データバウンドができないという問題点がある。また、逆方向のスクロールができないこともかなり不便である。

UDAsクラスの適所と制限

そこに、今回、紹介するUDAsクラスの存在理由がある。ちなみに、このクラスはWindowsアプリケーション専用として考えている。Webアプリケーションで使うのに、ほとんどメリットがないからである。まず、Webアプリケーションの場合、どのカーソルを使おうが配置的にはサーバーカーソルとして原理的に動作するため、トラフィック面でのメリットはない。また、性能面でもDataReaderクラスよりもSQL Serverとのラウンドトリップが多い分不利だし、Webアプリケーションの場合、逆方向のカーソル移動ができるということのメリットもあまりないだろう。ちなみにこのクラスはSQL Server専用である。

つまり、このUDAsオブジェクトは、SQL Serverをデータベースサーバーに使用したシステムで、Windowsアプリケーションで巨大な結果セットを扱う可能性があり、データ連結して前後方向にスクロールする必要があるケースにおいて使用すべきである。

このようにADO.NETでSQL Serverのサーバーカーソルの使用を実現するUDAsクラスだが、現時点では次のような制限がある。

制限1▶ データグリッドへの連結ができない

つまり、UDAsクラスはカード型のデータ表示、つまり一画面に1レコードを表示するようなケースでしかデータ連結に対応していない。この制限は、現在のUDAsクラスの仕様では回避することが難しい。しかし、解決するアイデアがないわけではないので、次号までに検討したい。

制限2▶ データの追加ができない

データの更新と削除はできるが、今のところ、データ