

.NET Framework

なにを使うか、どう使えるのか アイデアノート

最終回

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

P2Pアプリケーションを使った分散データベース

～マスターと更新系のテーブルが別マシンにある場合の連結クエリの実行

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。

¥NAMESERVER
ネームサーバー

¥P2P
P2Pアプリケーション

P2P DBアプリは自然と分散DBアプリになる

本連載の2004年8月号～10月号でP2Pアプリケーションを使った分散データベースシステムを紹介した。個々のP2Pアプリケーションがそれぞれデータを保持すると、自然と分散処理になる。今回は以前のプログラムを一步進めて、異なるクライアントに連結テーブルが存在する場合でもデータ連結ができるようにした。たとえば、顧客の受注状況を示すデータを個々のP2Pアプリケーションが保持するとしても、顧客マスターは一括管理したいかもしれない。あるいは一括管理しないにしても別のP2Pアプリケーションに登録されている顧客の受注管理をしたいかもしれない。しかし、以前のシステムでは、更新系の受注関係のテーブルと得意先マスターが同じP2Pクライアントに存在する必要があったため、そのよう

な処理はできなかった。今回はあくまで照会機能だけであって、分散管理されたデータを登録する機能はない。

より有効なデータベースにするには

分散データベースはまだ発展途上の技術だし、私自身も勉強不足である。通常分散データベースといった場合、あくまでデータベースサーバーが分散配置される。P2Pアプリケーションによって、クライアント自体が分散データベースサーバーとなるというのは分散データベースシステムとしては特殊な部類だといえるだろう。しかし、前述のようにP2P自体が分散コンピューティングだということを考えれば、P2Pデータベースアプリケーションが分散データベースシステムになるのは必然的である。これはデータを参照するユーザー自体がデータの

提供者になるということであり、この構図はインターネットの利用法ともマッチする。

テレビ放送のようにコンテンツの提供者と受容者が区別されるマスメディアに対して、インターネットの情報のやりとりは、その区別があいまいである。インターネットに含まれるコンテンツがより充実するには、コンテンツを提供するより多くの参加者が必要であり、そのためにはデータ提供のためのフォーマットを特定することが望ましい。

以前は、コンテンツを提供する人がHTMLを学習して情報を提供していたが、たとえばブログのようにフォーマットが提供されることで参加者は以前以上に増えている。情報提供の柔軟性が低くなるとはいえ、コンテンツの提供者がコンテンツの内容に注力できるからである。ただし、ブログも表示や記入のフォーマットが特定されているとはいえ、まだまだフリーフォーマットである。たとえば、文章が得意でない人は参加しにくい。そのような人が参加するためには項目に必要な内容を入力していくだけで、有用なデータを提供することができるようなフォーマットが必要である。それにより、参加のための障壁をより下げることができるだろう。一方、フォーマットが固定されることでユーザーが提供するデータの柔軟性は低下する。それを補うには、多種類のフォーマットを用意するか、あるいは一部のユーザーがフォーマットを変更できるような仕組みが必要だろう。だから、P2P分散データベースアプリケーションも特定のフォーマットを処理できるだけでなく、やはり汎用的に対応できないといけない。とはいえ、この記事のサンプルプログラムもあくまで特定のデータベースのための専用設計である。これを汎用的にするには、どのようにしたらいいのかも難しい。

テーブルが分散している場合の連結クエリの実行

今回のサンプルは以前のサンプルを変更したものが、データベースエンジンにはJetデータベースエンジンを使用している。前回までは、ADO.NET自体をデータ

ベースとして使用し、データベースエンジンレスで実装していたのを変更したのは、プログラミングコードを簡便にするためである。Jetデータベースエンジンを使用するということは、このアプリケーションをインストールするには、インストーラを使用しないといけないということである。ADO.NET自体をデータベースとして使用する場合は、ファイルをコピーするだけでインストールできたのに対して、やや不便になる。

前回までは、クライアント部の要求に対してサーバー部が結果を返し、それをそのまま表示すればよかった。しかし、今回のサンプルでは、まず顧客名で顧客テーブルを検索し、その結果を元に、再度、受注状況を保持するテーブルに対してクエリを実行する。複数の顧客がヒットした場合は、その数だけクエリを実行する。これで2つの結果セットが得られるので、その2つを連結することで最終的な検索結果を取得している。

つまり、連結の元になる2つのテーブルを検索の結果として作成し、それをクライアント側で連結することで、分散したテーブルの連結を実現しているのである。これは何か、不合理な気もするので他にいい手があるかとも思うが、とりあえず実装の容易さと動作の確実性という点でこの方法を選択した。

分散データベースの整合性

別の方法としては、問い合わせ先のサーバーが連結テーブルを探し、連結済みの結果を送り返すという方法も考えられる。この場合、無差別に探す方法の他に、データの作成時に連結テーブルを保有するサーバーを記録しておいて、そこで連結テーブルを探すという方法も考えられる。この方法だとデータは冗長になるが、無益なトラフィックの増加を避けることができる。一方、連結テーブルを持つサーバーが稼働していない場合、他のサーバーが連結可能なテーブルを持っていたとしても、それを見つけれないという問題が発生する。もちろん、指定されたテーブルが見つからなかった場合のみ無差別で探すというハイブリッドな方法もある。