

.NET Framework

なにを使うか、どう使えるのか アイデアノート

第10回

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

マルチスレッド TCP/IPサーバーを作成する

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。

¥SAMPLE1

簡単なマルチスレッドTCP/IPサーバー/クライアント

¥SAMPLE2

受信した内容をすべての接続クライアントに送り返すTCP/IPサーバー/クライアント

よし、今度はP2Pアプリケーションだ!

今回は、.NET Frameworkを用いて、P2P (Peer to Peer) アプリケーションを作ってみようと考えた。この連載では、できるだけ.NET Frameworkの長所を活かせるようなサンプルを提供したいと考えている。.NET Frameworkの長所であるデータベースの操作やWebアプリケーション、あるいはXML文書の扱いに関しては、今までに紹介してきた。他に.NET Frameworkが強い分野はなんだろうかと考えたときに、思いついたのがネットワークとマルチスレッドである。

さて、P2Pは「サーバーがないネットワーク形態」と説明されるが、ネットワークプログラミング的には、すべてがサーバーでありクライアントである“ネットワーク”である。サーバーとは「サービスを提供するもの」という意味だが、ネット

ワークプログラミングの文脈でいえば、「サーバーは相手からの接続を待機する側でもある」からである。.NET Frameworkを使えば、簡単にマルチスレッドプログラムを作れることを考えれば、ネットワークサーバーを作るのに適しているということもできる。

P2Pはファイル共有プログラムで有名になった。それによりイメージが悪くなったと嘆く人が多いが、これには技術的にみても理由がある。P2Pアプリケーションがメリットを発揮するのは、「インデックス情報が軽く、実データが重い情報をやりとりする場合」である。ファイル共有アプリケーションのように、ファイル名やアーティストといったテキスト情報は小さく、音声や動画などの本体は大きいデータをやりとりするのに向いているのである。たとえば、インスタントメッセージングソフトの多くは、テキストのやりとりであるチャットをする

ときにはサーバーを介したクライアントサーバーシステムとして動作するが、音声会話をするときには直接会話先に接続してP2Pアプリケーションとなる。



電話機とP2P



このような例もあってP2Pシステムと考えて、私が最初に思いつくのは「電話機」である。すべての電話機が“待ち状態にある”と考えると、接続という意味ではすべての電話機はサーバーである。P2Pアプリケーションで問題になるのは、最初の接続先をどのように特定するかである。電話の場合、家に電話帳があったりするわけで、つまりクライアントが別の文脈でデータベースを取得しているのである。ちなみに、この場合、電話帳はデータベースで、データベースエンジンは人間である。コンピュータシステムとして動作するP2Pの場合、いくらなんでももうすこしエレガントでないと格好悪いので、何か方法を考えなくてはいけない。ちなみに、最初に接続する相手を確定するために、多くのP2Pアプリケーションは相手先情報を取得するための情報をプログラムファイル（もしくはセットアップファイル）に埋め込んでいる。これはあまり洗練された方法ではないが、構造的に仕方ない部分でもある。

P2Pアプリケーションは「クライアントサーバー型とのハイブリッド型P2P」と、「ピュアP2P」に分類することができる。ハイブリッド型P2Pはインデックスデータを別サーバーに持ち、ピュアP2PはP2Pネットワークで完結している。ピュアP2Pでも少なくとも最初の接続先は、何か別の方法によって取得する必要がある。



どこにメリットがあるのか？



さて、P2Pアプリケーションを作ろうと考えてみたが、どのようにデータのやりとりをするプログラムを作るかが問題である。もちろん、これは本末転倒である。プログラムは先に目的があってから、手段が問われるべきで

ある。P2Pという手段が先にあってから、目的を考えるのはおかしい。しかし、ここは、そんなことには目をつぶってもらおう。

これまた、本末転倒な話で申し訳ないのだが、P2Pアプリケーションを作ろうと思いついて、いろいろ考えていたとき、私は分散データベースシステムというのを漠然と考えた。実のところインターネットのWWW自体が、分散データベースシステムであり、イントラネットのWebサーバー群も分散データベースシステムである。しかし、社内においてひとりひとりがWebサーバーを維持するというのは、ユーザーにとっても負担が大きい。

そこでP2Pアプリケーションそれぞれがデスクトップコンピュータにデータを蓄えておき、必要なときにはデータを統合して利用できるようなシステムがあれば、洗練されたものになるのではないかと考えた。だが、データベースシステムにとって、分散されているデメリットは大きい。あるデータの検索を開始する前に、そのデータがどのコンピュータにあるかを取得しないとけないのである。このオーバーヘッドはハイブリッド型のP2Pならばそれほど大きくはないが、ピュアP2P型のシステムでは、データベースエンジンのオーバーヘッドに比べ、とてつもなく大きい。また、今日のコンピュータの性能事情では、社内で負荷いっぱい稼働しているサーバーマシンはそれほどないはずである。分散コンピューティングの最大のメリットである負荷分散の効果もそれほど大きくはないだろう。と、このようにメリットも今ひとつはっきりしないのだが、私が作ったものを応用して、何か、役に立つものを考える人が出るかもしれないので、それに期待することにしよう。といっても、どんなものを作るか、まだ詳しくは考えていない。



.NET Frameworkを使ったマルチスレッドサーバー



今回は、.NET Frameworkを用いて、単純なマルチスレッドネットワークサーバーを作る方法を解説するだけである。だから、P2Pネットワークに興味がない人でも、.NET FrameworkのSocketsクラスのプログラミング記