

# .NET Frameworkで作る

# Windowsサーバー

作ればわかるアプリケーション  
の動作とメカニズム

第13回

## .NET Framework 2.0で作る プロキシサーバー

秋月巖ソリューション事務所

秋月 巖 AKIZUKI, Iwao

<http://www.akizuki.co.jp/>



### プロキシサーバー

前回、紹介したAST (Asynchronous Single Threading) サーバーをプログラミングテンプレートとして用いて、今回はプロキシサーバーを作成する。作る側から言ってしまうと、Web用のプロキシサーバーとは、Webサーバーがファイ

ルアクセスする代わりに他のWebサーバーにファイルを要求するものということができる。では、SMTP用のプロキシサーバーの場合、どういまいいのか考えたのだが、SMTPサーバーは、それ自身プロキシサーバーのようなものである。この連載を読んできた方ならば、SMTPサーバーがSMTPクライアントを内蔵していたことを覚えているだろう。SMTPサーバーとしてメールクライアントから受け取ったデータを他のSMTPサーバーに転送するのがSMTPサーバーの仕事なわけだから、これはまさにプロキシサーバーの仕事である。

今回、作成するのはWeb用のプロキシサーバーである(図1)。POSTメソッドのパラメータの扱いに問題があり、データ送信をするWebページをうまく処理できないことがある(うまくできることもある)。まあ、普通のWebページは見るので、一応、プロキシサーバーとして動作しているといえるだろう。



### プロキシサーバー の動作原理

プロキシサーバーが使用できるように設定されたWebブラウザは、Webページ表示の指示が出されたとき、常にプロキシサーバーに対して要求を行なう。プロキシサーバーを使わない通常のWebページの取得の場合、Webブラウザは対象コンピュータに取得するファイルのパス名付きのファイル名を送信するだけだが、プロキシサーバーに対し

図1: サンプルプロキシサーバー



レベル >>> Level

1 2 3 4 5

言語 >>> Language

Visual Basic

ツール >>> Tool

Visual Studio 2005 Professional

サンプル >>> Sample

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、<http://www.shoeisha.com/mag/windev/>からダウンロード可能です。



て要求を行なう場合「http://…」から始まる完全なURLを送信する。通常のWebサーバーへのアクセスの場合、ファイルは接続先のサーバーにあることが明白だが、プロキシサーバーの場合、ファイル名だけでは、どのサーバーにあるのかわからないから当然である。

Webページ要求を受け取ったプロキシサーバーは、Webブラウザのふりをして対象のWebサーバーに接続して指定されたファイルを要求する。今日ではWebサーバーもプロキシサーバーからもアクセスされることが前提になっているが、当初WebサーバーはWebブラウザからのみアクセスされるという前提だったはずなので、まさにWebブラウザを偽装していたわけである。

相手側のWebサーバーは当然のように要求されたファイルの内容を返信するので、プロキシサーバーはそれを受信して、今度はWebブラウザに対して、その内容を送信する。これでWebブラウザはターゲットのファイルを取得することができる。

Web用のプロキシサーバーならばWebブラウザからの接続を受け付けてWebサーバーにアクセスするし、POP3用のプロキシサーバーならばメールクライアントからの接続を受け付けてメールサーバーに接続する。



### プロキシサーバーの使用目的

プロキシサーバーの動作はシンプルだが、使用することによって通信経路が変更される。それにより、複数のメリットとデメリットが発生するので、使用する目的について簡単に説明しておいたほうがいだろう。

たとえば、ひとつのインターネット接続回線しかない環境で、2つ以上のコンピュータをインターネットに接続したい場合、プロキシサーバーを用いて実現することができる。それならば、ルーターと同じではないかと考える人もいるかもしれないが、その通りである。Web用のプロキシサーバーが稼動するコンピュータは、Web用のルーターと考えることができる。このようなケースでは、プロキシサーバーが稼動するコンピュータは、インターネットとローカルネットワークの両方に接続されている。ローカルネットワークに接続された他のコンピュータは、このプロキシサーバーが稼動するコンピュータを経由してインターネットに接続される。これ

はルーティングと同じである。ただし、あくまでアプリケーションレベルでの実装なので、対応するアプリケーション（この場合、WebブラウザとWebサーバー）でしか利用することができない。

また、プロキシサーバーを用いることで、Webブラウザが直接インターネットにアクセスしなくなるので、セキュリティを高めることが可能となる。プロキシサーバーを経由してアクセスした場合、相手側のWebサーバーには、プロキシサーバーが稼動するコンピュータがアクセスしたように見える。プロキシサーバーを経由しないで直接他のWebサーバーにアクセスした場合、IPアドレスばかりでなく、使用しているWebブラウザやOSの種類などのプライベート情報までが、相手先のWebサーバーに伝わってしまう。とはいえ、これらの情報を相手先に伝えるプロキシサーバーもあるので、単にプロキシサーバーを使用したから安心だというわけでもない。サンプルのプロキシサーバーはWebClientクラスを使用しているため、これらの情報を他のWebサーバーに送ることはない。

プロキシサーバーはWebブラウザとWebサーバーの通信経路の中間に位置するため、行き来するデータを制限したり加工したりする機能を付加することも可能である。サンプルのプロキシサーバーは、何も処理を行っていないが、そのような機能を実装することでキャッシュやフィルタリングの目的に使うこともできる。

キャッシュとは、一度、取得したデータを保存しておき、再度、要求があったときには、それを利用するというものである。Webブラウザにもキャッシュ機能があるが、それと同じものと思えばいい。Webブラウザのキャッシュとの違いは、Webブラウザのキャッシュがそのコンピュータでのみ再利用されるのに対して、プロキシサーバーのキャッシュデータはプロキシサーバーにアクセスするすべてのユーザーで共有できることである。キャッシュを使用することで、アクセス回数を減らすことができ、結果的にWebページの表示速度を上げることができる他、ネットワークトラフィックを減らすことも可能となる。

フィルタリングでは、次のようなことを行なうことができる。

①特定のユーザーからのアクセスだけを受け付ける。