

Apache, Mod_Proxy, & 4D

By Timothy Penner, Technical Services Team Member, 4D Inc.

Technical Note 09-38

概要

このテクニカルノートでは 4D Web サーバのフロントエンドとして Apache HTTP サーバを使用する方法について論じます。このために mod_proxy モジュールを使用します。いくつかのサンプル設定やオプションを議論します。この情報を使用すれば、顧客が望む際に、4D Web サーバのフロントエンドとして Apache を使用するための知識を得ることができます。

はじめに

mod_proxy モジュールを使用すれば、Apache をとても簡単に 4D Web サーバのフロントエンドとして利用できます。このテクニカルノートではさまざまな mod_proxy の設定やオプションについて検証します。この情報を得ることで 4D 開発者は、顧客が Apache を 4D Web サーバのフロントエンドとするよう要求してきたときに対応できる十分な知識を持つことができます。

Apache と Mod_Proxy について

Apache はオープンソースの HTTP Web サーバプロジェクトであり、The Apache Software Foundation により管理されています。Apache は広く利用されている HTTP Web サーバです。ライセンスは以下で確認できます:

<http://www.apache.org/licenses/>

Mod_proxy は Apache のモジュールで、The Apache Software Foundation により作成されました。Apache HTTP 上で特定のリクエストをキャッチし、(4DWeb サーバなど) 他のサーバに転送、そしてレスポンスをプロキシして返信できます。The Apache Foundation は Mod_proxy を以下のように説明しています:

このモジュールは Apache のプロキシ/ゲートウェイ機能を実装しています。AJP13 (Apache JServe Protocol version 1.3), FTP, CONNECT (SSL 用), HTTP/0.9, HTTP/1.0, HTTP/1.1 のプロキシ機能を実装しています。これらのプロトコルやその他のプロトコル用のプロキシ機能を持った、他のモジュールに接続するようにも設定できます。

Apache のプロキシ機能は mod_proxy の他に、いくつかのモジュールに分割されています: mod_proxy_http, mod_proxy_ftp, mod_proxy_ajp, mod_proxy_balancer, mod_proxy_connect です。ですから、特定のプロキシの機能を使いたい場合は、mod_proxy と該当するモジュールを

サーバに (コンパイル時に静的に行なうか LoadModule で動的に読み込むかして) 組み込む必要があります。

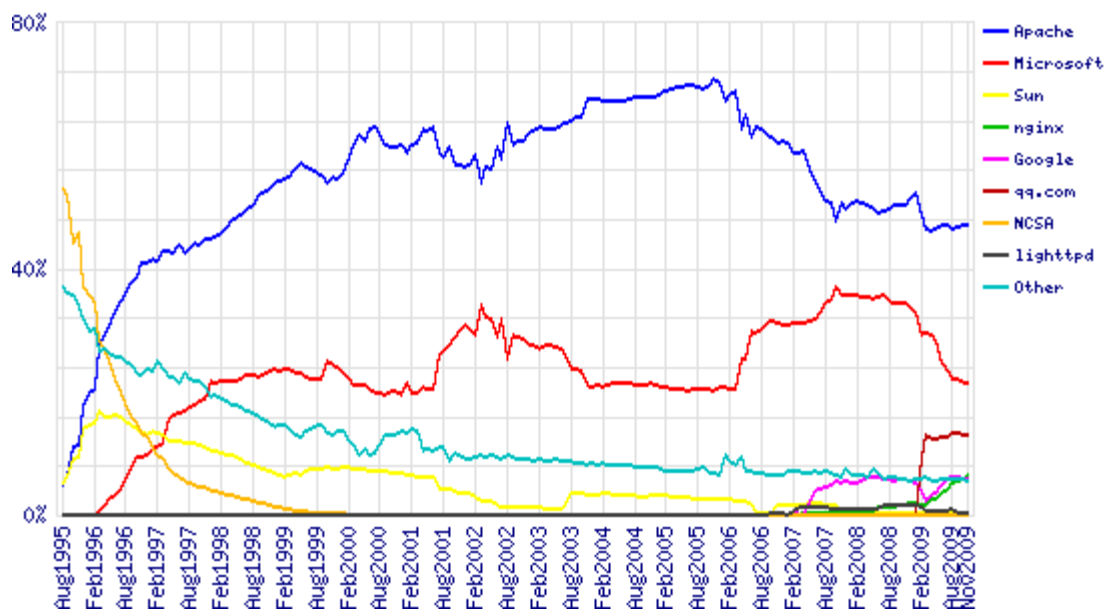
これに加えて、他のモジュールによって拡張機能が提供されています。キャッシュは mod_cache と関連モジュールで 提供されています。SSL/TLS で遠隔サーバに接続する機能は mod_ssl の SSLProxy* ディレクティブで 提供されています。これらの機能を利用するためには、該当するモジュールを 組み込んで設定しなければなりません。

Source: http://httpd.apache.org/docs/2.2/ja/mod/mod_proxy.html

Mod_proxy モジュールは Apache インストール時にインストールされます。

それらを使用する理由

Apache は多くの団体で使用されています。Apache Software Foundation によれば 1996 年 4 月よりインターネットでもっとも人気のある Web サーバです (このドキュメントが公開された当時で最新の情報による)。



Developer	October 2009	Percent	November 2009	Percent	Change
Apache	108,078,535	46.90%	110,201,883	47.17%	0.27
Microsoft	49,723,999	21.58%	49,691,412	21.27%	-0.31
qq.com	30,069,136	13.05%	30,069,189	12.87%	-0.18
nginx	13,813,997	5.99%	14,988,610	6.42%	0.42
Google	13,819,947	6.00%	13,771,004	5.89%	-0.10
lighttpd	1,020,227	0.44%	1,113,605	0.48%	0.03

出所: http://news.netcraft.com/archives/web_server_survey.html

多くの企業の IT 部署が Apache を気に入っており、展開にも採用しています。これはときに 4D Web サーバを使用したアイデアの販売に困難をもたらします。大企業のプロジェクトでは IT 担当者が自分の知っているものを採用したがるからです。

mod_proxy を使用すれば、Apache を 4D Web サーバのフロントエンドとして使用することができます。それで IT 担当者は引き続きかれらが気に入っているものを使うことができるというわけです。同時に 4D Web サーバからダイナミックなコンテンツを提供することができます。

mod_proxy はいくつかの方法で設定できますが、このテクニカルノートでは Apache を 4D Web サーバのフロントエンドとして使用 (リバースプロキシ) することに主眼を置きます。

どのように動作するか

Mod_proxy は Apache のモジュールであり、4D のものではありません。すなわちモジュールは Apache マシンにインストールされ、ロードされなければなりません。4D と Apache が同じマシンにインストールされている場合を除き、モジュールが 4D Web サーバと同じマシンにインストールされている必要はありません。もちろん、4D Web サーバと Apache Web サーバが異なるマシンにインストールされているときは、設定で指定されたポートを使用してお互いに通信できなければなりません。

Note: 4D 側で特別な設定を行う必要はありません。Apache 設定を使用して mod_proxy をロードし、4D Web サーバと通信を行うよう設定するだけです。

典型的な例では Apache Web サーバが標準ポート(80)、そして 4D Web サーバが (8080 などの) 非標準ポートで動作します。Mod_proxy を特定のリクエストをキャッチし、マッチするリクエストを 4D Web サーバに転送するよう設定されます。

設定は 2 つのカテゴリに大きく分けられます:

- 全サイトに適用されるグローバル設定; httpd.conf – OS やインストールの方法によりこのファイルは下記の場所がないこともありますが、通常はここに配置されています:
Mac OS X = /private/etc/apache2/httpd.conf
Windows = C:\Program Files\Apache\conf\httpd.conf
- 個々のサイトのための個々の設定およびディレクトリに対する設定。それらの設定は.htaccess ファイルとして適用するディレクトリにおかれます。

設定ファイルはプレーテキストで標準のテキストエディタを使用して編集できます。

設定ファイルの編集

Mod_proxy モジュールは標準の Apache 設定ファイルを使用します。つまりグローバル設定を httpd.conf ファイルに、個々の設定を.htaccess ファイルに記述することができます。

個々の.htaccess ファイルは通常設定を適用するディレクトリに置かれます。グローバル設定ファイルは以下の場所で見つかります:

Mac OS X = /private/etc/apache2/httpd.conf
Window = C:\Program Files\Apache\conf\httpd.conf

これらのファイルは標準のプレーンテキストエディタで編集できます。

このテクニカルノートではすべての設定をグローバル設定ファイルに対して行います。

Note: 設定ファイルに変更を行った後は、変更内容を有効にするために Apache サービスを再起動しなければなりません。

このテクニカルノートで使用するディレクティブ

このセクションではこのテクニカルノートで使用するディレクティブについて説明します。利用可能な全ディレクティブのリストとその利用方法については Mod_proxy モジュールの Web ページをご覧ください:

http://httpd.apache.org/docs/2.0/mod/mod_proxy.html

ProxyRequests

説明: フォワード (標準) プロキシを有効にする

シンタックス: ProxyRequests On|Off

デフォルト: ProxyRequests Off

コンテキスト: server config, virtual host

ステータス: Extension

モジュール: mod_proxy

このディレクティブは Apache をフォワードプロキシサーバとして動作させるか、させないかを指定します (ProxyRequests を Off に設定しても ProxyPass ディレクティブは利用できます)。

典型的なリバースプロキシの設定では、このオプションを Off に設定します。

HTTP や FTP サイトのプロキシ機能を得るためには、mod_proxy_http または mod_proxy_ftp (あるいは両方) がサーバに存在しなければなりません。

警告: サーバを安全な状態に設定できるまで *ProxyRequests* を有効にしないでください。オープンプロキシサーバはあなたのネットワークおよびインターネットに対し危険です。

ProxyPass

説明: リモートサーバをローカルサーバの URL 空間にマップする

シンタックス: ProxyPass [path] !|url

コンテキスト: server config, virtual host, directory

ステータス: Extension

モジュール: mod_proxy

このディレクティブはリモートサーバをローカルサーバの空間にマップするために使用します。ローカルサーバは従来の意味においてプロキシとしては動作しませんが、リモートサーバのミラーのように見えます。“[path]”はローカルバーチャルパスの名前です。“url”はリモートサーバの URL の一部でクエリ文字列を含むことはできません。

ローカルサーバが以下のアドレスを持つ時 `http://example.com/`;

```
ProxyPass /mirror/foo/ http://backend.example.com/
```

は `http://example.com/mirror/foo/bar` へのローカルリクエストを内部的にプロキシリクエスト `http://backend.example.com/bar` に変換します。

<Location>セクション内で使用される場合第一引数は省略され、ローカルディレクトリは<Location>から取得されます。先の例は以下のように書けます:

```
<Location /mirror/foo/>
ProxyPass http://backend.example.com/
</Location>
```

サブディレクトリをリバースプロキシしたくない場合、!ディレクティブを使用できます。例えば

```
ProxyPass /mirror/foo/i !
ProxyPass /mirror/foo http://backend.example.com
```

は/mirror/foo へのすべてのリクエストを backend.example.com にプロキシしますが、/mirror/foo/i は除きます。

Note: 順番は重要です。例外は通常の ProxyPass ディレクティブの前に書きます。

ProxyPass 利用時には通常 ProxyRequests ディレクティブを Off に設定します。

より自由度の高いリバースプロキシ設定を行うには、RewriteRule ディレクティブの[P]フラグを参照してください。

ProxyPassReverse

説明: リバースプロキシサーバから送信された HTTP レスポンスヘッダ内の URL を調整する

シンタックス: ProxyPassReverse [path] url

コンテキスト: server config, virtual host, directory

ステータス: Extension

モジュール: mod_proxy

このディレクティブは Apache が Location の URL、Content-Location、そして HTTP リダイレクトレスポンスの URI ヘッダを変更できるようにします。これは Apache をリバースプロキシとして使用する場合に不可欠のものです。これによりバックエンドサーバ上での HTTP リダイレクトを調整し、リバースプロキシがバイパスされることを防ぎます。

先に示した HTTP レスポンスヘッダのみが書き直されます。Apache 他のレスポンスヘッダや、HTML 内の URL 参照を変更しません。これはプロキシされたコンテンツが全体 URL 参照を含んでいる場合、これはプロキシをバイパスすることを意味します。サードパーティーのモジュールである Nick Kew の `mod_proxy_html` は HTML の中身を解析し URL 参照の書き換えを行います。

“`[path]`”はローカルバーチャルパスの名前です。“`url`”はリモートサーバの URL の一部で、ProxyPass ディレクティブと同じように使用されます。

例えばローカルサーバのアドレスが以下のとき: `http://example.com/`

```
ProxyPass /mirror/foo/ http://backend.example.com/
ProxyPassReverse /mirror/foo/ http://backend.example.com/
```

は `http://example.com/mirror/foo/bar` へのローカルリクエストを (ProxyPass が提供する機能により) 内部的に変換して `http://backend.example.com/bar` へのプロキシリクエストにするだけでなく、サーバ `backend.example.com` が送信するリダイレクトも調整します。例えば `http://backend.example.com/bar` が `http://backend.example.com/quux` にリダイレクトされる際、Apache は HTTP リダイレクトレスポンスをクライアントにフォワードする前にこのリダイレクト先を `http://example.com/mirror/foo/quux` に変更します。URL を構築するために使用されるホスト名は `UseCanonicalName` ディレクティブの設定に基づき決められます。

<Location>セクション内で使用される場合第一引数は省略され、ローカルディレクトリは<Location>から取得されます。先の例は以下のように書けます:

```
<Location /mirror/foo/>
ProxyPass http://backend.example.com/
ProxyPassReverse http://backend.example.com/
</Location>
```

Note: ProxyPassReverse ディレクティブは `mod_rewrite (RewriteRule ... [P])` のプロキシパススルー機能とともに使用することもできます。ProxyPass ディレクティブ対応に依存するものではありません。

例題

`mod_proxy` モジュールを使用すれば、開発者は独自の設定ファイルを記述し、完全動作をカスタマイズできます。この節では 4D 開発者が最も興味があり、必要とするであろう設定にフォーカスします。

特定のディレクトリのみを 4D にプロキシする

この節では Apache の Mod_proxy モジュールを使用して、Apache への特定のリクエストを 4D Web サーバにプロキシする方法を示します。

```
# start
<IfModule mod_proxy.c>
  ProxyRequests Off
  <Location /4D/>
    ProxyPass http://127.0.0.1:8080/
    ProxyPassReverse http://127.0.0.1:8080/
  </Location>
</IfModule>
# end
```

この設定ファイルは /4D/ で始まるすべての入力 Apache リクエストをキャッチし、そのリクエストを 4D Web サーバが実行されている 127.0.0.1:8080 にプロキシします。このようにして、4D は直接受け取ったかのように Web リクエストを受け取ることができます。この設定において、http://mysite.com/4D/ を 4D Web サーバのルートとみなすことができます。例えば http://mysite.com/4D/reports/sales.shtml は 4D Web サーバに http://127.0.0.1:8080/reports/sales.shtml へのリクエストとして送信されます。

Note: この例題の URL 中の /4D/ は Apache が使用するエントリーポイントであり、4D に送信される HTTP リクエストからは取り除かれます。

SOAP リクエストのみを 4D にプロキシする

この節では Apache の Mod_proxy モジュールを使用して、SOAP リクエストを 4D Web サーバにプロキシする方法を示します。

```
# start
<IfModule mod_proxy.c>
  ProxyRequests Off
  <Location /4DSOAP>
    ProxyPass http://127.0.0.1:8080/4DSOAP
    ProxyPassReverse http://127.0.0.1:8080/4DSOAP
  </Location>
</IfModule>
# end
```

この設定ファイルは 4DSOAP で始まるすべての入力 Apache リクエストをキャッチし、そのリクエストを 4D Web サーバが実行されている http://127.0.0.1:8080/4DSOAP/ にプロキシします。このようにして、4D は直接受け取ったかのように SOAP リクエストを受け取ることができます。この設定においては 4DSOAP から始まる SOAP リクエストのみが 4D Web サーバにプロキシされます。

SOAP リクエストと特定のディレクトリを 4D にプロキシする

この節では Apache の Mod_proxy モジュールを使用して、SOAP リクエストおよび特定のリクエストを 4D Web サーバにプロキシする方法を示します。

```
# start
<IfModule mod_proxy.c>
  ProxyRequests Off
  <Location /4D/>
    ProxyPass http://127.0.0.1:8080/
    ProxyPassReverse http://127.0.0.1:8080/
  </Location>
  <Location /4DSOAP>
    ProxyPass http://127.0.0.1:8080/4DSOAP
    ProxyPassReverse http://127.0.0.1:8080/4DSOAP
  </Location>
</IfModule>
# end
```

この設定ファイルは 4D/または 4DSOAP で始まるすべての入力 Apache リクエストをキャッチし、そのリクエストを 4D Web サーバが実行されている 127.0.0.1:8080 にプロキシします –

- リクエストが 4D/で始まっていれば、4D Web サーバのルートにプロキシされ、“4D/”はリクエストから取り除かれます。
- リクエストが 4DSOAP で始まっていれば、そのリクエストが直接 4D Web サーバに対して行われたかのように 4D Web サーバにプロキシされます。

このようにすれば 4DSOAP からはじまる SOAP リクエストも、4D/から始まるリクエストも両方 4D Web サーバにプロキシできます。

すべてのリクエストを 4D にプロキシする

この節では Apache の Mod_proxy モジュールを使用して、すべてのリクエストを 4D Web サーバにプロキシする方法を示します。

```
# start
<IfModule mod_proxy.c>
  ProxyRequests Off
  <Location />
    ProxyPass http://127.0.0.1:8080/
    ProxyPassReverse http://127.0.0.1:8080/
  </Location>
</IfModule>
# end
```

この設定ファイルでは Apache が受け取るすべてのリクエストを、4D Web サーバが実行されている 127.0.0.1:8080 にプロキシします。このようにすれば 4D は Apache からすべてのリクエストを受け取ります。Apache Web サーバのルートは 4D Web サーバのエントリポイントとして考えることができます。例えば <http://mysite.com/reports/sales.shtml> は <http://127.0.0.1:8080/reports/sales.shtml> のように変換されます。

WSDL ファイルに関する注意

4DWSDL ファイルのコンテンツをプロキシすることはできますが、rewrite エンジンでは正しくポート番号を取り除くことができないようです。これは自動で WSDL をベースとした Web サービスプロキシメソッドを作成する際に問題となります。そのため、4DWSDL をファイルとして保存し、データベースが提供する Web サービスごとに "soap:address" 要素の location 属性を編集します。これを行うには 4DWSDL ファイルを Web ブラウザにロードし、ファイル -> 別名で保存を選択します。これによりファイルが XML フォーマットで保存され、編集できます。このファイルを Web サイトに配置し、あなたの Web サービスユーザに新しいアドレスを通知します。また新しいファイルをユーザにメール等で送信することもできます。

まとめ

このテクニカルノートでは mod_proxy モジュールを使用して 4D から Apache にコンテンツをプロキシする一般的なコンセプトについて説明しました。いくつかの設定がテクニカルノート内で議論され、この情報は顧客の求めに応じ 4D Web サーバのフロントエンドに Apache を配置するための知識を提供しました。

Apache HTTP サーバに関する詳細な情報:

<http://httpd.apache.org/>

Apache のライセンスに関する情報:

<http://www.apache.org/licenses/>

Mod_Proxy モジュールに関する情報:

http://httpd.apache.org/docs/2.0/mod/mod_proxy.html

コミュニティ、サポート、メーリングリスト:

<http://httpd.apache.org/lists.html>