

マルチプロセッシングモジュール (MPM)

この文書ではマルチプロセッシングモジュールがどのようなもので、Apache HTTP サーバでどのように使用されるかについて解説しています。

トピック

はじめに.....	1
MPM を選ぶ.....	1
MPM デフォルト値.....	2
URI References.....	2

はじめに

Apache HTTP サーバは異なる幅広い環境、多種多様なプラットフォームで動作するように、パワフルで柔軟性に富んだ設計になっています。異なるプラットフォーム・異なる環境ではしばしば、異なる機能が必要になったり、同じ機能でも効率のために異なる実装が必要になったりします。Apache ではモジュール化された設計により幅広い環境に適応してきました。この設計のおかげで、管理者は コンパイル時または実行時にどのモジュールをロードするか選ぶことによって、どの機能をサーバに取り込むか選択することができます。

Apache 2.0 では、このモジュール化された設計をサーバの基本機能にまで拡張しました。サーバには精選されたマルチプロセッシングモジュール (MPM) が付いてきて、これらはマシンのネットワークポートをバインドしたり、リクエストを受け付けたり、リクエストを扱うようプロセスに割り当てたり、といった役割を持ちます。

モジュール化された設計をサーバのこのレベルまで拡張することで 二つの重要な利点が生れます:

- Apache は幅広いオペレーティングシステムを より美しく効率的にサポートできます。特に Windows 版の Apache は随分効率的になりました。なぜなら `mpm_winnt` によって、Apache 1.3 で用いられていた POSIX レイヤの代わりにネイティブのネットワーク機能を利用できるからです。特別化された MPM を実装した他のオペレーティングシステムでも、同様にこの利点は生まれます。
- サーバは特定のサイト向けに、より上手にカスタマイズできます。例えば、非常に大きなスケラビリティを必要とするサイトでは、`worker` といったスレッド化された MPM を利用できる一方で、安定性や古いソフトウェアとの互換性を必要とするサイトでは `prefork` が利用できます。また、異なるホストを異なるユーザ ID で動作させる (`perchild`) といった 特別な機能も提供できます。

ユーザレベルでは、MPM は他の Apache モジュールと同等に見えます。主な違いは、いつでも唯一の MPM がロードされなければならないという点です。利用可能な MPM は `module` インデックス¹にあります。

MPM を選ぶ

MPM は設定中に選択して、サーバ内部にコンパイルされなければなりません。コンパイラは、スレッドが使用されていれば様々な機能を最適化できますが、そもそもスレッドが使われているということを知る必要があります。MPM には Unix 上でスレッドを用いるものや、スレッドをまったく使わないものがあるので、Apache は、MPM が設定中に選択されて Apache 内部に組み込まれた場合の方が 常により良いパフォーマンスを発揮します。

マルチプロセッシングモジュール (MPM)

望みの MPM を実際に選ぶためには、./configure スクリプトで --with-mpm= NAME 引数を用いてください。NAME は望みの MPM の名前です。

サーバコンパイル後は、どの MPM が選択されたかを ./httpd -l で確かめることができます。このコマンドは、MPM を含め、サーバにコンパイルで組み込まれたモジュール全てを列挙します。

MPM デフォルト値

次表に様々な OS 向けのデフォルトの MPM 一覧を掲載しています。コンパイル時に意図的に他を選択しなければ、自動的にこれらの MPM が選択されます。

BeOS	beos
Netware	mpm_netware
OS/2	mpmt_os2
Unix	prefork
Windows	mpm_winnt

URI References

[1] <http://httpd.apache.org/docs-2.1/mod/>