

Apache MPM prefork

説明:	スレッドを使わず、先行して fork を行なうウェブサーバを実装
ステータス:	MPM
モジュール識別子	mpm_prefork_module
:	
ソースファイル:	prefork.c

概要

このマルチプロセッシングモジュール (MPM) は、Unix 上での Apache 1.3 のデフォルトの挙動と非常によく似た方法で リクエストを処理する、スレッドを使わず、先行して fork を行なう ウェブサーバを実装しています。スレッドセーフでないライブラリとの互換性をとるために、スレッドを避ける必要のあるサイトでは、このモジュールの使用が適切でしょう。あるリクエストで発生した問題が他のリクエストに影響しないように、個々のリクエストを単離するのにも、最適な MPM です。

この MPM は非常に自律的なので、この MPM の設定ディレクティブを調整する必要はほとんどないでしょう。もっとも重要なことは、`MaxClients` が、予想される同時リクエスト数を十分扱えるぐらいは大きいけれども、全プロセスに十分な物理メモリが確実に行き渡る程度には小さい値にする、ということです。

トピック

動作方法.....	1
URI References.....	4

ディレクティブ

AcceptMutex.....	2	MaxSpareServers.....	3
CoreDumpDirectory.....	➡	MinSpareServers.....	3
Group.....	➡	PidFile.....	➡
Listen.....	➡	ScoreBoardFile.....	➡
ListenBacklog.....	➡	SendBufferSize.....	➡
LockFile.....	➡	ServerLimit.....	➡
MaxClients.....	➡	StartServers.....	➡
MaxMemFree.....	➡	User.....	➡
MaxRequestsPerChild.....	➡		

(➡ This directive is defined elsewhere. See: [mpm_common](#))

参照

- Apache が使用するアドレスとポートの設定¹

動作方法

一つのコントロールプロセスが、コネクションに対して listen して、しかるべき時に応答する子プロセスを起動します。Apache は常に幾つかのスペア かアイドルなサーバプロセスを維持していて、それらは入ってきた リクエストに応答できるように待機しています。このようにしてクライアントは、リクエストが応答される前に、新しい子プロセスが fork されるのを待たなくてもよいようになります。

Apache MPM prefork

親プロセスがリクエストに回答するのの子プロセスをどのように生成するかは、`StartServers`、`MinSpareServers`、`MaxSpareServers`、`MaxClients` で調整します。一般的に、Apache は非常に自律的なので、大抵のサイトではこれらのディレクティブをデフォルト値から調整する必要はないでしょう。同時に 256 を超えるリクエストに回答しないといけないサイトでは、`MaxClients` を増やす必要があるでしょう。一方、メモリの限られているサイトでは、スラッシング（メモリとディスク間で何度もスワップ）が起こるのを防ぐために `MaxClients` を減らす必要があるでしょう。プロセス生成のチューニングに関する詳しい情報は、性能に関するヒント² にあります。

通常 Unix では親プロセスは 80 番ポートにバインドするために root で起動されますが、子プロセスやスレッドはもっと低い権限のユーザで Apache によって起動されます。`User` と `Group` ディレクティブは Apache の子プロセスの権限を設定するのに用いられます。子プロセスはクライアントに送るコンテンツ全てを読めないといけません、可能な限り必要最小限の権限のみを持っているようにすべきです。

`MaxRequestsPerChild` は、古いプロセスを停止して新しいプロセスを起動することによって、どの程度の頻度でサーバがプロセスをリサイクルするかを制御します。

AcceptMutex ディレクティブ

説明:	複数の子プロセスがネットワークソケットでリクエストを accept しようとしているときに、Apache がそれらの子プロセスを直列化するために使う方法
構文:	<code>AcceptMutex default method</code>
デフォルト:	<code>AcceptMutex default</code>
コンテキスト:	サーバ設定ファイル
ステータス:	MPM
モジュール:	<code>prefork</code>

`AcceptMutex` ディレクティブは、ネットワークソケットのリクエストを accept しようとしている複数の子プロセスを Apache が直列化するために使う方法を設定します。Apache 2.0 より前では、このメソッドはコンパイル時にのみ選択できました。最適な方法は、アーキテクチャやプラットフォームに大きく依存します。詳細に関しては、性能のチューニング² ドキュメントをご覧ください。

このディレクティブが `default` に設定されていれば、コンパイル時に選択されたデフォルト値が使われます。他の使用可能なメソッドの一覧は下にあります。全てのメソッドが全てのプラットフォームで使用可能であるわけではない、ということに注意してください。使用可能でないメソッドが指定された場合は、使用可能なメソッドの一覧を含んだメッセージがエラーログに出力されます。

flock

`LockFile` ディレクティブで定義したファイルのロックに、`flock(2)` システムコールを使います。

fcntl

`LockFile` ディレクティブで定義したファイルのロックに、`fcntl(2)` システムコールを使います。

sysvsem

Apache MPM prefork

排他処理の実装に SysV 形式のセマフォを使います。

pthread

POSIX Threads (PThreads) 規格で実装されている POSIX 排他処理を使います。

MaxSpareServers ディレクティブ

説明:	アイドルな子サーバプロセスの最大個数
構文:	MaxSpareServers number
デフォルト:	MaxSpareServers 10
コンテキスト:	サーバ設定ファイル
ステータス:	MPM
モジュール:	prefork

MaxSpareServers ディレクティブは、アイドルな子サーバプロセスの希望最大個数を設定します。アイドルプロセスとは、リクエストを扱っていないプロセスです。MaxSpareServers よりも多い数がアイドルであれば、親プロセスは超過プロセスを kill します。

非常に混んでいるサイトでのみ、このパラメータをチューニングする必要があります。このパラメータを大きくすることは、大抵の場合は悪い発想です。

参照

- [MinSpareServers](#)
- [StartServers](#)

MinSpareServers ディレクティブ

説明:	アイドルな子サーバプロセスの最小個数
構文:	MinSpareServers number
デフォルト:	MinSpareServers 5
コンテキスト:	サーバ設定ファイル
ステータス:	MPM
モジュール:	prefork

MaxSpareServers ディレクティブは、アイドルな子サーバプロセスの希望最小個数を設定します。アイドルプロセスとは、リクエストを扱っていないプロセスです。MinSpareServers よりも少ない数がアイドルであれば、親プロセスは最高で 1 秒につき 1 個の割合で新しい子プロセスを生成します。

非常に混んでいるサイトでのみ、このパラメータをチューニングする必要があります。このパラメータを大きくすることは、大抵の場合は悪い発想です。

参照

- [MaxSpareServers](#)
- [StartServers](#)

URI References

- [1] <http://httpd.apache.org/docs-2.1/bind.html>
- [2] <http://httpd.apache.org/docs-2.1/misc/perf-tuning.html>