

Apache MPM worker

説明:	マルチスレッドとマルチプロセスのハイブリッド型 ウェブサーバを実装したマルチプロセッシングモジュール
ステータス:	MPM
モジュール識別子:	mpm_worker_module
ソースファイル:	worker.c

概要

このマルチプロセッシングモジュール (MPM) は、マルチスレッドとマルチプロセスのハイブリッド型サーバを実装しています。リクエストの応答にスレッドを使うと、プロセスベースのサーバよりも少ないシステム資源で、多くのリクエストに応答することができます。さらに、多くのスレッドを持った複数のプロセスを維持することで、プロセスベースのサーバの持つ安定性を保持しています。

この MPM を制御するのに使われる最も重要なディレクティブは、`ThreadsPerChild` と `MaxClients` です。`ThreadsPerChild` は 各子プロセスで用意されるスレッド数を制御して、`MaxClients` は 起動されるスレッドの総数の最大値を制限します。

トピック

動作方法.....	1
URI References.....	2

ディレクティブ

CoreDumpDirectory.....	⇒	MinSpareThreads.....	⇒
Group.....	⇒	PidFile.....	⇒
Listen.....	⇒	ScoreBoardFile.....	⇒
ListenBacklog.....	⇒	SendBufferSize.....	⇒
LockFile.....	⇒	ServerLimit.....	⇒
MaxClients.....	⇒	StartServers.....	⇒
MaxMemFree.....	⇒	ThreadLimit.....	⇒
MaxRequestsPerChild.....	⇒	ThreadsPerChild.....	⇒
MaxSpareThreads.....	⇒	User.....	⇒

(⇒ This directive is defined elsewhere. See: `mpm_common`)

参照

- Apache の使用するアドレスとポートの設定 ¹

動作方法

プロセスは決まった数のスレッドを持ちます。負荷への適応はプロセスの数を増減により対応します。

一つの制御用プロセスが子プロセスを起動します。子プロセスは `ThreadsPerChild` ディレクティブで指定された一定数のスレッドを作ります。それから、個々のスレッドがコネクションに対して `listen` して、しかるべき時に動作します。

Apache はスペアの、つまりアイドルなサーバスレッドの プールを常に維持していて、それ

Apache MPM worker

らは入ってくるリクエストに 答えられ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の応答が得られるようになるために 新しいスレッドやプロセスが生成されるのを 待たなくてもよいようになっています。起動初期時のプロセス総数は、`StartServers` ディレクティブで設定されます。その後の稼働中に、Apache は全プロセスのアイドルスレッドの合計数を見積もって、`MinSpareThreads` と `MaxSpareThreads` で指定された範囲の中にこの数が収まるように fork したり kill したりします。この操作は非常に自律的なので、これらのディレクティブをデフォルト値から変更する必要は めったにないでしょう。同時に応答することのできるクライアント数の最大数（つまり全プロセス中の総スレッド数の最大値）は `MaxClients` ディレクティブで決定されていて、一方で起動することのできるプロセス数の最大値は `ServerLimit` ディレクティブで設定されています。`ServerLimit` と `ThreadsPerChild` との積は、`MaxClients` と等しいかそれより大きい必要があります。

`worker` MPM の典型的なプロセス・スレッド制御の 設定では、次のようになります。

```
StartServers 2
MaxClients 150
MinSpareThreads 25
MaxSpareThreads 75
ThreadsPerChild 25
ServerLimit 16
```

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。さらに、`suexec`² が使用されていない限り、これらのディレクティブは CGI スクリプトで継承される権限も設定します。

`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。

URI References

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

[2] <http://httpd.apache.org/docs-2.1/suexec.html>