Apache MPM worker

Description:	Multi-Processing Module implementing a hybrid multi-threaded multi-process web server
Status:	MPM
Module Identifier:	mpm_worker_module
Source File:	worker.c

Summary

This Multi-Processing Module (MPM) implements a hybrid multi-process multi-threaded server. By using threads to serve requests, it is able to serve a large number of requests with less system resources than a process-based server. Yet it retains much of the stability of a process-based server by keeping multiple processes available, each with many threads.

The most important directives used to control this MPM are `ThreadsPerChild`, which controls the number of threads deployed by each child process and `MaxClients`, which controls the maximum total number of threads that may be launched.

Topics

Directives

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

See also

- Setting which addresses and ports Apache uses[1]

How it Works

Each process has a fixed number of threads. The server adjusts to handle load by increasing or decreasing the number of processes.

A single control process is responsible for launching child processes. Each child process creates a fixed number of threads as specified in the `ThreadsPerChild` directive. The individual threads then listen for connections and serve them when they arrive.

Apache always tries to maintain a pool of *spare* or idle server threads, which stand ready to serve incoming requests. In this way, clients do not need to wait for a new threads or processes to be created before their requests can be served. The number of processes that will initially launched is set by the `StartServers` directive. Then during operation, Apache assesses the total number of idle threads in all processes, and forks or kills processes to keep this number within the boundaries specified by

MinSpareThreads and MaxSpareThreads. Since this process is very self-regulating, it is rarely necessary to modify these directives from their default values. The maximum number of clients that may be served simultaneously (i.e., the maximum total number of threads in all processes) is determined by the MaxClients directive, while the maximum number of processes that can be launched is set by the ServerLimit directive. ServerLimit multiplied by ThreadsPerChild must be greater than or equal to MaxClients

A typical configuration of the process-thread controls in the worker MPM could look as follows:

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

While the parent process is usually started as root under Unix in order to bind to port 80, the child processes and threads are launched by Apache as a less-privileged user. The User and Group directives are used to set the privileges of the Apache child processes. The child processes must be able to read all the content that will be served, but should have as few privileges beyond that as possible. In addition, unless suexec [2] is used, these directives also set the privileges which will be inherited by CGI scripts.

MaxRequestsPerChild controls how frequently the server recycles processes by killing old ones and launching new ones.

URI References

[1] http://httpd.apache.org/docs-2.1/bind.html
[2] http://httpd.apache.org/docs-2.1/suexec.html