

Apache HTTP Server Version 2.4

Apache MPM prefork

Description:	Implements a non-threaded, pre-forking web server
Status:	MPM
Module Identifier:	mpm_prefork_module
Source File:	prefork.c

Summary

This Multi-Processing Module (MPM) implements a non-threaded, pre-forking web server. Each server process may answer incoming requests, and a parent process manages the size of the server pool. It is appropriate for sites that need to avoid threading for compatibility with non-thread-safe libraries. It is also the best MPM for isolating each request, so that a problem with a single request will not affect any other.

This MPM is very self-regulating, so it is rarely necessary to adjust its configuration directives. Most important is that `MaxRequestWorkers` be big enough to handle as many simultaneous requests as you expect to receive, but small enough to assure that there is enough physical RAM for all processes.



Topics

- How it Works

Directives

- CoreDumpDirectory
- EnableExceptionHook
- Group
- Listen
- ListenBacklog
- MaxConnectionsPerChild
- MaxMemFree
- MaxRequestWorkers
- MaxSpareServers
- MinSpareServers
- PidFile
- ReceiveBufferSize
- ScoreBoardFile
- SendBufferSize
- ServerLimit
- StartServers
- User

Bugfix checklist

- httpd changelog
- Known issues

- [Report a bug](#)

See also

- [Setting which addresses and ports Apache HTTP Server uses](#)
- [Comments](#)

How it Works

A single control process is responsible for launching child processes which listen for connections and serve them when they arrive. Apache httpd always tries to maintain several *spare* or idle server processes, which stand ready to serve incoming requests. In this way, clients do not need to wait for a new child processes to be forked before their requests can be served.

The `StartServers`, `MinSpareServers`, `MaxSpareServers`, and `MaxRequestWorkers` regulate how the parent process creates children to serve requests. In general, Apache httpd is very self-regulating, so most sites do not need to adjust these directives from their default values. Sites which need to serve more than 256 simultaneous requests may need to increase `MaxRequestWorkers`, while sites with limited memory may need to decrease `MaxRequestWorkers` to keep the server from thrashing (swapping memory to disk and back). More information about tuning process creation is provided in the performance hints ([↗ ../misc/perf-tuning.html](#)) documentation.

While the parent process is usually started as `root` under Unix in order to bind to port 80, the child processes are launched by Apache httpd as a less-privileged user. The `User` and `Group` directives are used to set the privileges of the Apache httpd 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.

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

This MPM uses the `mpm-accept` mutex to serialize access to incoming connections when subject to the thundering herd problem (generally, when there are multiple listening sockets). The implementation aspects of this mutex can be configured with the `Mutex` directive. The performance hints ([↗ ../misc/perf-tuning.html](#)) documentation has additional information about this mutex.

MaxSpareServers Directive

Description:	Maximum number of idle child server processes
Syntax:	<code>MaxSpareServers</code> <i>number</i>
Default:	<code>MaxSpareServers</code> 10
Context:	server config
Status:	MPM
Module:	prefork

The `MaxSpareServers` directive sets the desired maximum number of *idle* child server processes. An idle process is one which is not handling a request. If there are more than `MaxSpareServers` idle, then the parent process will kill off the excess processes.

Tuning of this parameter should only be necessary on very busy sites. Setting this parameter to a large number is almost always a bad idea. If you are trying to set the value equal to or lower than `MinSpareServers`, Apache HTTP Server will automatically adjust it to `MinSpareServers` + 1.

See also

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

MinSpareServers Directive

Description:	Minimum number of idle child server processes
Syntax:	<code>MinSpareServers <i>number</i></code>
Default:	<code>MinSpareServers 5</code>
Context:	server config
Status:	MPM
Module:	prefork

The [MinSpareServers](#) directive sets the desired minimum number of *idle* child server processes. An idle process is one which is not handling a request. If there are fewer than [MinSpareServers](#) idle, then the parent process creates new children: It will spawn one, wait a second, then spawn two, wait a second, then spawn four, and it will continue exponentially until it is spawning 32 children per second. It will stop whenever it satisfies the [MinSpareServers](#) setting.

Tuning of this parameter should only be necessary on very busy sites. Setting this parameter to a large number is almost always a bad idea.

See also

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

Comments

Notice:

This is not a Q&A section. Comments placed here should be pointed towards suggestions on improving the documentation or server, and may be removed again by our moderators if they are either implemented or considered invalid/off-topic. Questions on how to manage the Apache HTTP Server should be directed at either our IRC channel, #httpd, on Freenode, or sent to our mailing lists.

**Copyright 2019 The Apache Software Foundation.
Licensed under the Apache License, Version 2.0.**