

# HTTPD的安装

## 安装

版本: httpd-2.4.6-45.el7.centos.x86\_64

```
1 | yum -y install httpd
```

## 相关文件及目录

```
1 | /etc/logrotate.d/httpd    #日志滚动配置文件
2 | /etc/sysconfig/httpd      #脚本配置文件
3 | /usr/lib/systemd/system/httpd.service    #服务脚本
4 | /var/www                  #默认文档根目录
5 | /etc/httpd                #运行目录
6 | /etc/httpd/log            #默认访问日志及错误日志目录
7 | /etc/httpd/conf           #主配置文件目录
8 | /etc/httpd/modules.d      #模块存放目录
9 | /etc/httpd/conf/httpd.conf #默认httpd配置文件
10 | /etc/httpd/conf.d         #扩展配置文件，每一个配置都可以写入
    | conf.d目录，以.conf为后缀，方便管理配置
11 | /etc/httpd/conf.modules.d  #httpd模块配置文件目录
12 | /etc/httpd/conf.modules.d/00-base.conf  #基础模块配置
    | 文件
13 | /etc/httpd/conf.modules.d/00-mpm.conf   #MPM模块配置
    | 文件
```

## httpd设计特性

1. 高度模块化设计，由 core + modules 组成
2. DSO(Dynamic Share Object) 。动态启动关闭模块
3. MPM：多道处理模块，非一个模块。是一个多路处理模块。与其他类型的模块不同的是，必须在编译配置时进行选择，必须有且仅有一个MPM被静态编译到服务器中。此类型的模块负责基本的对请求的处理和指派。

## 4. 多平台运行

### httpd命令

```
1 httpd -t      #配置文件语法检测
2 httpd -l      #显示httpd编译进内核的模块
3 httpd -v      #显示版本信息、使用MPM模型等
4 httpd -M      #显示所有支持加载的模块
```

## MPM多道处理模块

在Linux平台，apache的MPM模块有 `prefork` `worker` `event`，针对不同需求的站点，可以采用不同的模式。Apache httpd 通过模块化的设计来适应各种环境。这种设计允许网站管理员通过在编译时或运行时，选择哪些模块将会加载在服务器中，来选择服务器特性。任何时间，必须有一个，而且只有一个 MPM 加载到服务器中。要求更高伸缩性的站点可以选择使用线程的 MPM，即 `worker` 或 `event`，需要可靠性或者与旧软件兼容的站点可以使用 `prefork`。

切换不同工作模式，即切换 MPM 模块，可以修改配置文件：

```
1 vim /etc/httpd/modules.d/00-mpm.conf
2 LoadModule mpm_prefork_module
  modules/mod_mpm_prefork.so
3 #LoadModule mpm_worker_module
  modules/mod_mpm_worker.so
4 #LoadModule mpm_event_module modules/mod_mpm_event.so
```

## MPM模块工作原理

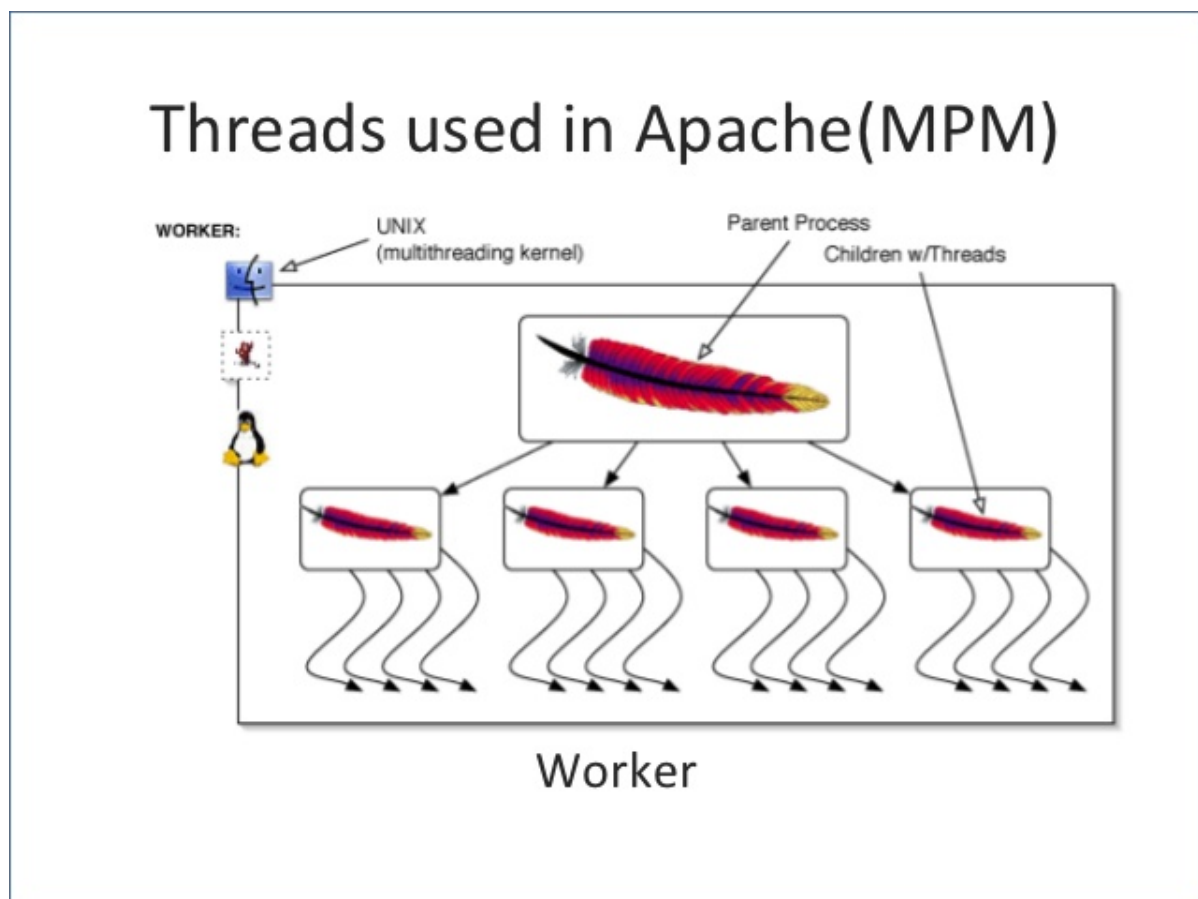
### prefork

由父进程管理多个子进程，一个子进程响应一个请求。当 apache 启动时预先分配出子进程，即预先 fork 出子进程，当有访问请求时可以立刻响应，同时也采用空闲回收机制，即当空闲子进程超过设置的值时，主动停掉多余空闲的子进程。它也是隔离每个请求的最佳 MPM，因为单个的请求不会影响任何其他请求。prefork 模型，不适合高并发的场

景，每个子进程都会占用系统资源。prefork 会预先保留一些空闲子进程，以备有请求的时候快速响应。非线程型、预派生、可以将 MaxClients 调到大值，以便应对潜在请求高峰。

## worker

由父进程生成管理多个子进程，和 prefork 相同也产生多个子进程，worker 和 prefork 不同的是一个进程又会产生多个线程，一个线程响应一个请求，一个进程产生的线程数量是固定的。和 prefork 相同，worker 也会保留空闲进程，以备有请求的时候快速响应。因为线程可以共享父进程的内存空间，所以 worker 模式下会相对节省系统资源。适应要求更有伸缩性的站点。工作模式如下图：



## event

和worker类似。区别是：event模块中，会有一些不负责响应请求的线程，用来释放已经响应完成的工作线程，以用来更好的响应请求，MPM-event 模块为每个进程使用专用的监听线程来检测所有处于 Keep Alive 状态的套接字，当有Keep Alive长期连接，且没有请求甚至超时，用来监听的线程将他释放。能够承受更高的并发负载，节省系统资源。

# MPM模块详细配置

## MPM/Prefork配置

### 1.配置格式：

```
1 <IfModule mpm_prefork_module>
2     [Run-time Configuration Directives] VALUE
3     .
4     .
5     .
6 </IfModule>
```

### 2.配置指令解释：

默认为5。初始启动httpd时，默认启动几个工作进程，不包括主进程，通常也没有必要修改此参数。

```
1 StartServers 5
```

默认是5。最少空闲进程数，如果空闲进程数小于这个数，则父进程以分裂的方式创建子进程。如果不是非常繁忙的服务器，不建议修改此值，因为开启过多的子进程会占用系统资源，反而会影响性能。

```
1 MinSpareServers 5
```

默认值为10。最多空闲进程数，如果空闲子进程超过这个数，则父进程会管理终止多余的子进程。如果您尝试将值设置为等于或低于MinSpareServers，Apache HTTP Server 将自动将其调整为MinSpareServers + 1。

```
1 MaxSpareServers 10
```

默认值256。并发请求最大数，设置同时请求数的限制，即可用于响应请求的最大子进程数如果超过这个值，则通过尝试将请求放入队列等待。在 httpd/2.3.13 版本之前，此配置指令是：MaxClients，但 httpd/2.4 依然兼容这个指令。

## 1 | MaxRequestWorkers 256

当前最大活动进程数。ServerLimit一般要大于MaxRequestWorkers。20000是最大值，如果需要更大值，则需要重新编译进行定制。

## 1 | ServerLimit 256

默认为0。配置进程将处理的[连接数]的限制。即限制单个子进程的生命周期，处理多少个连接后，这个进程被杀死。如果值为0，表示子进程永远不会过期。可用于httpd/2.3.9以上的版本，之前的版本配置指令是：MaxRequestsPerChild。

设置为非0值的好处：

- 1.可以防止(偶然的)内存泄漏无限进行，从而耗尽内存。
- 2.给进程一个有限寿命，有助于当服务器负载减轻的时候减少活动进程的数量

## 1 | MaxConnectionsPerChild 0

3.配置示例：

```
1 <IfModule mpm_prefork_module>
2     StartServers 5
3     MinSpareServers 5
4     MaxSpareServers 10
5     MaxRequestWorkers 256
6     ServerLimit 256
7     MaxConnectionsPerChild 0
8 </IfModule>
```

## MPM/Worker配置

1.配置格式：

```
1 <IfModule mpm_worker_module>
2     [Run-time Configuration Directives] VALUE
3     .
4     .
5     .
6 </IfModule>
```

## 2.配置指令解释：

默认值是3。初始启动 httpd 时，默认启动几个工作子进程。

```
1 StartServers 3
```

同时处理的最大连接数。ServerLimit 乘以 ThreadsPerChild 的结果。因此要增加 MaxRequestWorkers 的时候，你必须同时增加 ServerLimit 的值。

```
1 MaxRequestWorkers 256
```

默认75。最小空闲进程数。

```
1 MinSpareThreads 75
```

默认250。最大空闲进程数。

```
1 MaxSpareThreads 250
```

默认值是64。每个子进程创建的线程数。

```
1 ThreadsPerChild 64
```

## 3.配置示例：

```
1 <IfModule mpm_worker_module>
2     StartServers 3
3     MaxRequestWorkers 1024
4     ServerLimit 16
5     MinSpareThreads 75
6     MaxSpareThreads 250
7     ThreadsPerChild 64
8     MaxConnectionsPerChild 0
9 </IfModule>
```

## MPM/Event配置

event模块配置指令和worker相同，只添加了  
`AsyncRequestWorkerFactor`指令。

### 1.配置示例：

```
1 <IfModule mpm_event_module>
2     StartServers 3
3     MaxRequestWorkers 1024
4     ServerLimit 16
5     MinSpareThreads 75
6     MaxSpareThreads 250
7     ThreadsPerChild 64
8     MaxConnectionsPerChild 0
9 </IfModule>
```

## 参考

<http://httpd.apache.org/docs/>