## php-fpm的静态static和动态dynamic执行方式比较

前段时间配置php-fpm的时候，无意间发现原来他还有两种执行方式。与Apache一样，他的进程数也是可以根据设置分为动态和静态的。

而php-fpm也是同样存在两种方式，一种是直接开启指定数量的php-fpm进程，不再增加或者减少；另一种则是开始的时候开启一定数量的php-fpm进程，当请求量变大的时候，动态的增加php-fpm进程数到上限，当空闲的时候自动释放空闲的进程数到一个下限。

这两种不同的执行方式，可以根据服务器的实际需求来进行调整。

这里先说一下涉及到这个的几个参数吧，他们分别是pm、pm.max\_children、pm.start\_servers、pm.min\_spare\_servers和pm.max\_spare\_servers。

pm表示使用那种方式，有两个值可以选择，就是static（静态）或者dynamic（动态）。在更老一些的版本中，dynamic被称作apache-like。这个要注意看配置文件给出的说明了。

下面4个参数的意思分别为：

pm.max\_children：静态方式下开启的php-fpm进程数量。

pm.start\_servers：动态方式下的起始php-fpm进程数量。

pm.min\_spare\_servers：动态方式下的最小php-fpm进程数量。

pm.max\_spare\_servers：动态方式下的最大php-fpm进程数量。

如果dm设置为static，那么其实只有pm.max\_children这个参数生效。系统会开启设置的数量个php-fpm进程。

如果dm设置为dynamic，那么pm.max\_children参数失效，后面3个参数生效。系统会在php-fpm运行开始的时候启动 pm.start\_servers个php-fpm进程，然后根据系统的需求动态在pm.min\_spare\_servers和 pm.max\_spare\_servers之间调整php-fpm进程数。

那么，对于我们的服务器，**选择哪种执行方式比较好呢？**

事实上，跟Apache一样，我们运行的PHP程序在执行完成后，或多或少会有内存泄露的问 题。这也是为什么开始的时候一个php-fpm进程只占用3M左右内存，运行一段时间后就会上升到20-30M的原因了。所以，动态方式因为会结束掉多余 的进程，可以回收释放一些内存，所以推荐在内存较少的服务器或者VPS上使用。具体最大数量根据 内存/20M 得到。比如说512M的VPS，建议pm.max\_spare\_servers设置为20。至于pm.min\_spare\_servers，则建议根据服 务器的负载情况来设置，比较合适的值在5~10之间。

然后对于比较大内存的服务器来说，设置为静态的话会提高效率。因为频繁开关php-fpm进程也会有时滞，所以内存够大的情况下开静态效果会更好。数量也可以根据 内存/30M 得到。比如说2GB内存的服务器，可以设置为50；4GB内存可以设置为100等。

本站是建立在512M的VPS上，因此我设置的参数如下：

pm=dynamic

pm.max\_children=20

pm.start\_servers=5

pm.min\_spare\_servers=5

pm.max\_spare\_servers=20

这样就可以最大的节省内存并提高执行效率。

附各参数说明：

pm string

设置进程管理器如何管理子进程。可用值：static，ondemand，dynamic。必须设置。

static - 子进程的数量是固定的（pm.max\_children）。

ondemand - 进程在有需求时才产生（当请求时，与 dynamic 相反，pm.start\_servers 在服务启动时即启动。

dynamic - 子进程的数量在下面配置的基础上动态设置：pm.max\_children，pm.start\_servers，pm.min\_spare\_servers，pm.max\_spare\_servers。

pm.max\_children int

pm 设置为 static 时表示创建的子进程的数量，pm 设置为 dynamic 时表示最大可创建的子进程的数量。必须设置。

该选项设置可以同时提供服务的请求数限制。类似 Apache 的 mpm\_prefork 中 MaxClients 的设置和 普通PHP FastCGI中的 PHP\_FCGI\_CHILDREN 环境变量。

pm.start\_serversin

设置启动时创建的子进程数目。仅在 pm 设置为 dynamic 时使用。默认值：min\_spare\_servers + (max\_spare\_servers - min\_spare\_servers) / 2。

pm.min\_spare\_servers int

设置空闲服务进程的最低数目。仅在 pm 设置为 dynamic 时使用。必须设置。

pm.max\_spare\_servers int

设置空闲服务进程的最大数目。仅在 pm 设置为 dynamic 时使用。必须设置。

pm.max\_requests int

设置每个子进程重生之前服务的请求数。对于可能存在内存泄漏的第三方模块来说是非常有用的。如果设置为 ‘0’ 则一直接受请求，等同于 PHP\_FCGI\_MAX\_REQUESTS 环境变量。默认值：0。