

php中fastcgi和php-fpm是什么东西

php

最近在研究和学习php的性能方面的知识，看到了fastcgi以及php-fpm，发现我对他们是少之又少的理解，可以说几乎是一无所知，想想还是蛮可怕的。决定仔细的学习一下关于这方面的知识。

参考和学习了以下文章：

1. [mod_php和mod_fastcgi和php-fpm的介绍,对比,和性能数据](#)
2. [实战Nginx_取代](#)

为了如何一步步的引出fastcgi和php-fpm，我先一点一点的说说关于php的周边。哎。突然觉得人活着好累！

先说说web服务器

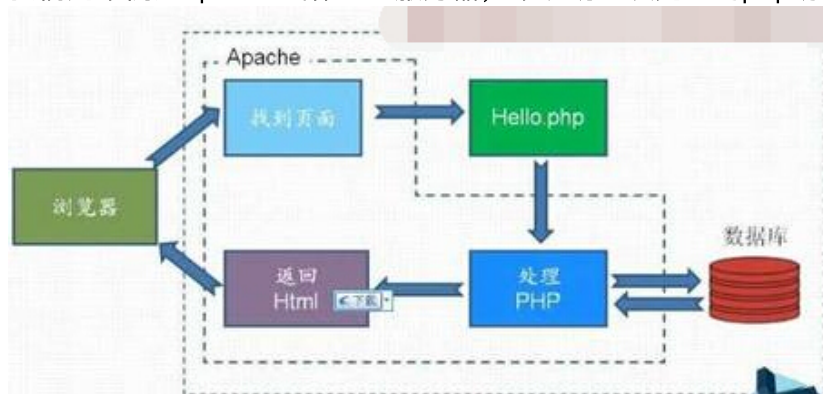
php是为web而生的一门后端语言，我们php狗当然是最清楚的啦。所以php仅仅是一门后端语言，那么它就必须借助于web服务器，才能提供web功能。当然其他的后端语言如果做web应用，也必须借助于web服务器。好，由php引出了web服务器，不错！

那么常见的web服务器有哪些呢？php狗用的最多的就是Apache了，还有其他的：

- apache
- nginx
- IIS
- lighttpd
- tomcat

基本上就是上面几种，与php相关联起来用的最多的就是Apache和Nginx了。

我们先举例用apache当作web服务器，来说明一次完整的php访问的情况：



图片中就很好的解释了php与Apache结合mysql数据库的一次完成的web访问流程图

mod_php模式

上面讲清楚了php必须借助于web服务器才能提供web的功能服务，现在看下他俩是怎么成为基友的。

我们用到的最多的就是Apache了。那么回忆一下，如何使apache是怎么能够识别php代码的？是不是apache的配置文件httpd.conf中加上或者修改这样几句：

```
1. //加入以下2句
2. LoadModule php5_module D:/php/php5apache2_2.dll
3. AddType application/x-httpd-php .php
4.
5. //将下面的
6. <IfModule dir_module>
7.     DirectoryIndex index.html
8. </IfModule>
9. //将其修改为:
10. <IfModule dir_module>
11.     DirectoryIndex index.html index.htm index.php index.phtml
12. </IfModule>
```

上面的windows下安装php和apache环境后的手动配置，在linux下源码安装大致是这样配置的：

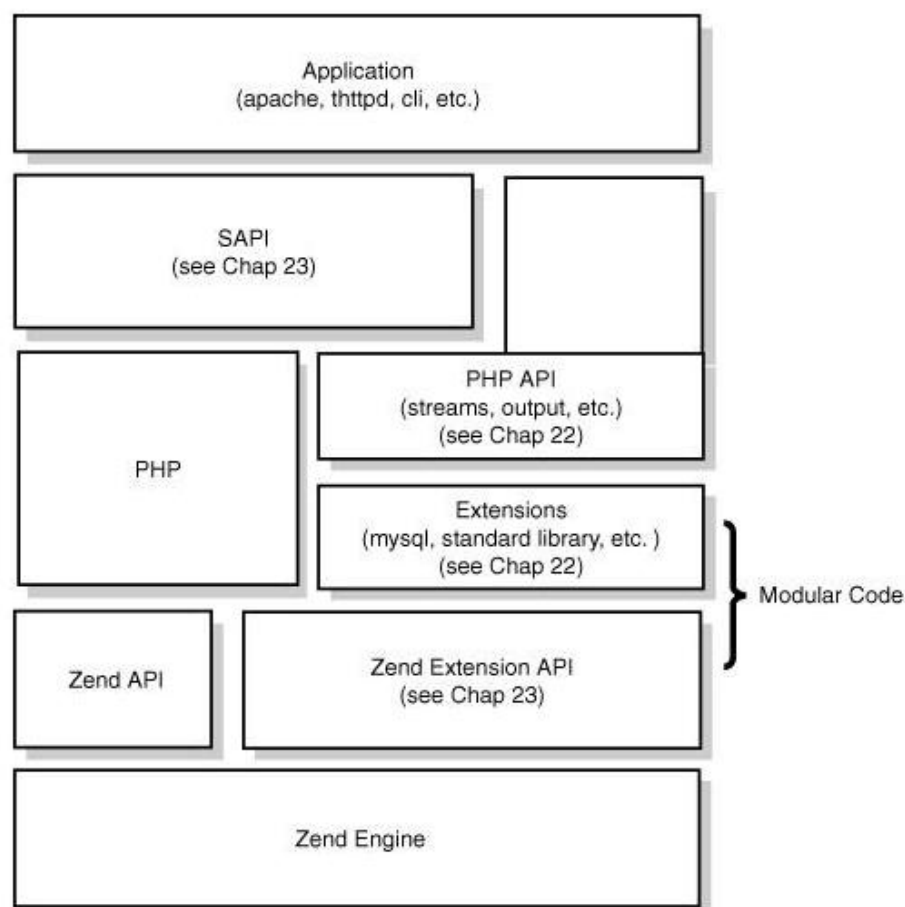
```
1. ./configure --with-mysql=/usr/local --with-apache=/usr/local/apache --enable-track-vars
```

所以，这种方式，他们的共同本质都是用LoadModule来加载php5_module，就是把php作为apache的一个子模块来运行。当通过web访问php文件时，apache就会调用php5_module来解析php代码。

那么php5_module是怎么来将数据传给php解析器来解析php代码的呢？

答案是通过sapi

我们再来看一张图，详细的说说apache 与 php 与 sapi的关系：



从上面图中，我们看出了sapi就是这样的一个中间过程，SAPI提供了一个和外部通信的接口，有点类似于socket，使得PHP可以和其他应用进行交互数据（apache,nginx,cli等）。php默认提供了很

多种SAPI，常见的给apache和nginx的php5_module，CGI，给IIS的ISAPI，还有Shell的CLI。

所以，以上的apache调用php执行的过程如下：

```
apache -> httpd -> php5_module -> sapi -> php
```

好了。apache与php通过php5_module的方式就搞清楚了吧！

我们把这种运行方式叫做**mod_php**模式

mod_fastcgi模式

上面我们仔细说了php与apache通过php5_module,php5_module通过sapi的方式访问php，来达到php web的整个流程。

上面也说到sapi，sapi是php提供的统一接口，它提供给了php5_module和cgi等方式供web服务器来链接和解析php代码。上面讲到的php5_module加载模式，我们称之为mod_php模式。

那么！当当当当！马上就要说出fastcgi模式了。哈哈哈哈哈，太不容了。

那么php的sapi的另一种方式就是提供cgi模式，由于cgi比较老所以就出现了fastcgi来取代它。

所以，哎。没办法，又要说什么是CGI了？

CGI(Common Gateway Interface)。CGI是外部应用程序（CGI程序）与Web服务器之间的接口标准，是在CGI程序和Web服务器之间传递信息的规程。CGI规范允许Web服务器执行外部程序，并将它们的输出发送给Web浏览器，CGI将Web的一组简单的静态超媒体文档变成一个完整的新的交互式媒体。

看官方的解释就蛋疼，简单的说，就是：cgi就是专门用来和web 服务器打交道的。web服务器收到用户请求，就会把请求提交给cgi程序（php的fastcgi），cgi程序根据请求提交的参数作应处理（解析php），然后输出标准的html语句返回给web服务器，再返回给客户端，这就是普通cgi的工作原理。

cgi的好处就是完全独立于任何服务器，仅仅是做为中间分子。提供接口给apache和php。他们通过cgi搭线来完成搞基动作。这样做的好处了尽量减少2个的关联，使他们2变得更独立。

但是cgi有个蛋疼的地方，就是每一次web请求都会有启动和退出过程，也就是最为人诟病的fork-and-execute模式，这样一在大规模并发下，就死翘翘了。

所以。这个时候fastcgi运用而生了。它事先就早早的启动好了，而且可以启动多个cgi模块，在那里一直运行着等着，等着web发过来的请求，然后再给php解析运算完成生成html给web后，也不会退出，而且继续等着下一个web请求。而且这些cgi的模块启动是可控的，可监测的。这种技术还允许把web server和php运行在不同的主机上，以大规模扩展和改进安全性而不损失生产效率。

所以现在一般操作系统都是fastcgi模式。cig模式也慢慢退出了历史舞台！我们文章中说cgi一般也就指fastcgi。

所以把这种运行方式叫做**mod_fastcgi**模式

我会在接下来的段落讲如何使用fastcgi模式来连接php和apache(或者nginx)

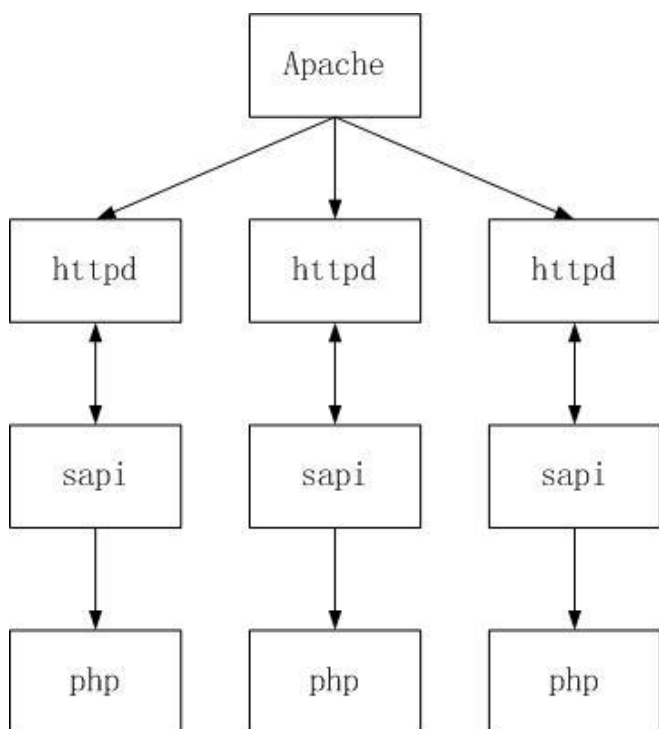
总结一下：php 与 apache 或者 ngx 结合，会用sapi 提供2种连接方法:mod_php和mod_fastcgi。

mod_php 模式会将php模块安装到apache下面来运行，2者结合度较大。mod_fastcgi模式则是作为一个中间过程，apache介绍用户请求后，就发送给fastcgi, 再连接php来完成访问。

图形表示一下这2种模式

mod_php 模式

mod_php 模式是将php模块安装到apache中，所以每一次apache结束的请求呢，都会产生一条进程，这个进程就完整的包括php的各种运算计算等操作。

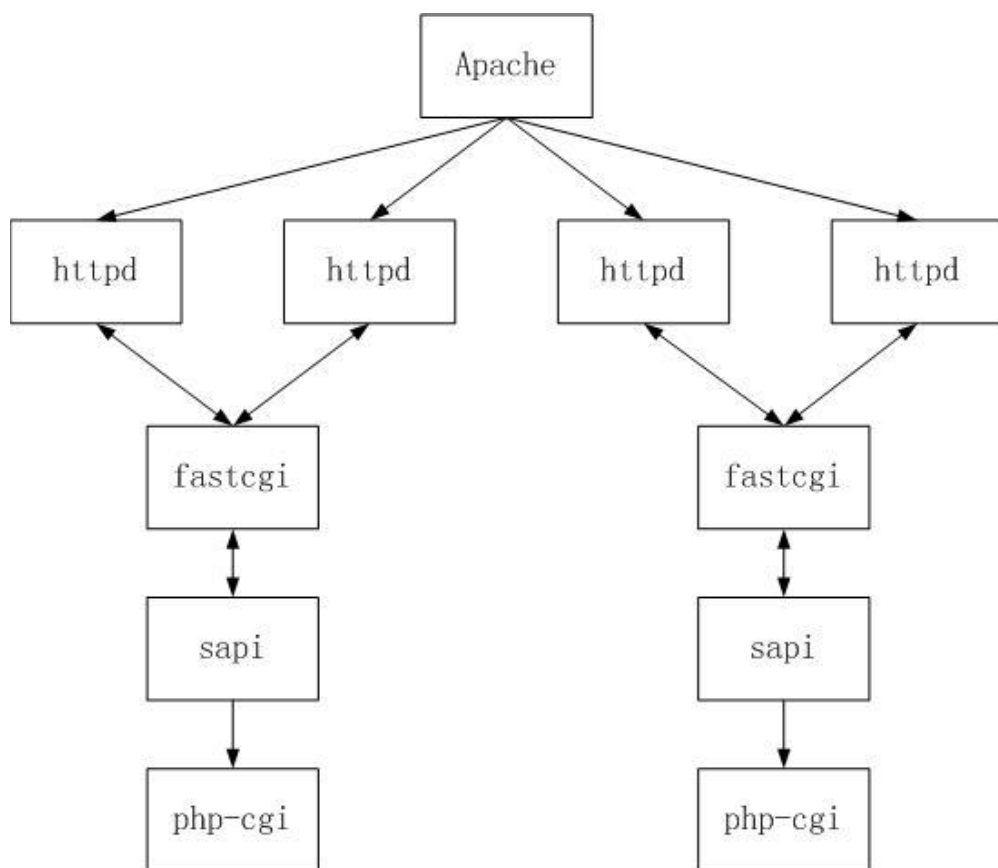


从图中我们很清晰的可以看到，apache每接收一个请求，都会产生一个进程来连接php通过sapi来完成请求，可想而知，如果一旦用户过多，并发数过多，服务器就会承受不住了。

而且，把mod_php编进apache时，出问题时很难定位是php的问题还是apache的问题。

mod_fastcgi 模式

mod_fastcgi模式则刚刚相反，fastcgi是一个独立与apache和php的独立个体，它随着apache一起启动，生成多个cig模块，等着apache的请求：



图中fastcgi早早的启动好了，静静的在哪里等着，已有apache发来的httpd请求就立马接收过来，通过调用sapi给php，完成运算。而且不会退出。这样就能应对大规模的并发请求，因为web server的要做的事情少了，所以就更快的去处理下一个请求，这样并发大大的。

由于apache 与 php 独立了。出问题，很好定位到底是哪里出问题了。这点也是这种模式受欢迎的原因之一。

php-fpm

我了个大操，终于要说到php-fpm了。^....^

先开门见山说php-fpm是干嘛好的了。它就是专门来辅助mode_fastcgi模式的。

嗯。很好，先知道它是干嘛的后，我们再回到mode_fastcgi模式。通过前面的瞎鸡巴一大堆的说明，我已经搞清楚了这种模式是怎么样子的一种状态了。

fastcgi 是一个与平台无关，与语言无关，任何语言只要按照它的接口来实现，就能实现自己语言的fastcgi能力和web server 通讯。

PHP-CGI就是PHP实现的自带的FastCGI管理器。

虽然是php官方出品，自带的，但是这丫的却一点也不给力，性能太差，而且也很麻烦不人性化，主要体现在：

1. php-cgi变更php.ini配置后需重启php-cgi才能让新的php-ini生效，不可以平滑重启。
2. 直接杀死php-cgi进程，php就不能运行了。

上面2个问题，一直让很多人病垢了很久，所以很多人一直还是在用mode_php方式。

直到 2004年(确定是这么早吗?)一个叫 Andrei Nigmatulin的屌丝发明了PHP-FPM，这神器的出现就彻底打破了这种局面，这是一个PHP专用的fastcgi管理器，它很爽的克服了上面2个问题，而