**nginx如何调用php**

 转载

采用nginx+php作为webserver的架构模式，在现如今运用相当广泛。然而第一步需要实现的是如何让nginx正确的调用php。由于nginx调用php并不是如同调用一个静态文件那么直接简单，是需要动态执行php脚本。所以涉及到了对nginx.conf文件的配置。这一步对新手而言一般需要网上查资料，对于一般的熟手而言，也有不少同学并没有搞透彻为何要如此这般配置。本文的主要内容为如何在nginx server中正确配置php调用方法，以及配置的基本原理。

**一、nginx+php运行原理：**

首先简单的讲一讲原理，目前主流的nginx+php的运行原理如下：   
1、nginx的worker进程直接管理每一个请求到nginx的网络请求。   
2、对于php而言，由于在整个网络请求的过程中php是一个cgi程序的角色，所以采用名为php-fpm的进程管理程序来对这些被请求的php程序进行管理。php-fpm程序也如同nginx一样，需要监听端口，并且有master和worker进程。worker进程直接管理每一个php进程。   
3、关于fastcgi：fastcgi是一种进程管理器，管理cgi进程。市面上有多种实现了fastcgi功能的进程管理器，php-fpm就是其中的一种。再提一点，php-fpm作为一种fast-cgi进程管理服务，会监听端口，一般默认监听9000端口，并且是监听本机，也就是只接收来自本机的端口请求，所以我们通常输入命令 netstat -nlpt|grep php-fpm 会得到：

|  |  |
| --- | --- |
| 1 | tcp        0      0 127.0.0.1:9000              0.0.0.0:\*                   LISTEN      1057/php-fpm |

这里的127.0.0.1:9000 就是监听本机9000端口的意思。   
4、关于fastcgi的配置文件，目前fastcgi的配置文件一般放在nginx.conf同级目录下，配置文件形式，一般有两种：

fastcgi.conf 和 fastcgi\_params。不同的nginx版本会有不同的配置文件，这两个配置文件有一个非常重要的区别：fastcgi\_parames文件中缺少下列配置：

fastcgi\_param SCRIPT\_FILENAME $document\_root$fastcgi\_script\_name;

我们可以打开fastcgi\_params文件加上上述行，也可以在要使用配置的地方动态添加。使得该配置生效。   
5、当需要处理php请求时，nginx的worker进程会将请求移交给php-fpm的worker进程进行处理，也就是最开头所说的nginx调用了php，其实严格得讲是nginx间接调用php。   
了解了上面的这五个简单原理，在nginx中配置php调用方法就变得易如反掌。

**二、配置文件：**

直接贴上代码逐行进行讲解，此处贴出一个能正常启动php脚本的最简nginx vhost配置：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | server {      listen       8011;      server\_name  test.cn;      location ~ \.php?.\*$ {          root           /share/test;          fastcgi\_pass   127.0.0.1:9000;          fastcgi\_param  SCRIPT\_FILENAME  $document\_root$fastcgi\_script\_name;          include        fastcgi\_params;      }  } |

1、第一个大括号 server{ }：不必多说，代表一个独立的server，   
2、listen 8011：代表该server监听8011端口   
3、location ~ .php?.\*${ }：代表一个能匹配对应uri的location，用于匹配一类uri，并对所匹配的uri请求做自定义的逻辑、配置。这里的location，匹配了所有带.php的uri请求，例如：http://192.168.244.128:8011/test.php/asdasd http://192.168.244.128:8011/index.php等   
4、root /share/test：请求资源根目录，告诉匹配到该location下的uri到/share/teset文件夹下去寻找同名资源。   
5、fastcgi\_pass 127.0.0.1:9000：这行开始是本文的重点：这行代码的意思是，将进入到该location内的uri请求看做是cgi程序，并将请求发送到9000端口，交由php-fpm处理。   
6、fastcgi\_param SCRIPT\_FILENAME

fastcgi\_script\_name; ：这行配置意思是：动态添加了一行fastcgi配置，配置内容为SCRIPT\_FILENAME，告知管理进程，cgi脚本名称。由于我的nginx中只有fastcgi\_params文件，没有fastcgi.conf文件，所以要使php-fpm知道SCRIPT\_FILENAME的具体值，就必须要动态的添加这行配置。   
7、include fastcgi\_params; 引入fastcgi配置文件   
以上就是最简洁版的nginx启动php脚本的最简配置，当重启nginx之后，在/share/test目录下创建一个xx.php文件，输入

**三、总结：**

其实对于调用php这类cgi脚本程序，只要理解了我开头提到的5点原理，然后结合5-7行配置讲解，完全可以较清晰的明白为什么需要这样配置了。对于新手而言，往往被fastcgi,php-fpm cgi程序搞得一头雾水，胡乱配置已通，跑通上线，也不去深究其原理。所以希望写在这里的东西能对读者带来一点点的帮助。

**php-fpm的工作原理**

一、代理与反向代理   
现实生活中的例子   
1、正向代理：访问google.com



如上图，因为google被墙，我们需要vpnFQ才能访问google.com。

vpn对于“我们”来说，是可以感知到的（我们连接vpn）vpn对于”google服务器”来说，是不可感知的(google只知道有http请求过来)。

对于人来说可以感知到，但服务器感知不到的服务器，我们叫他正向代理服务器。

2、反向代理：通过反向代理实现负载均衡



如上图，我们访问baidu.com的时候，baidu有一个代理服务器，通过这个代理服务器，可以做负载均衡，路由到不同的server。(此代理服务器,对于“我们”来说是不可感知的(我们只能感知到访问的是百度的服务器，不知道中间还有代理服务器来做负载均衡)。)

此代理服务器，对于”server1 server2 server3”是可感知的(代理服务器负载均衡路由到不同的server)   
对于人来说不可感知，但对于服务器来说是可以感知的，我们叫他反向代理服务器

总结   
说白了：“正向”、“反向”是相对于人的感知来说的。人能感受到的代理就是正向代理，人感受不到的代理就是反向代理。

二、初识Nginx与Php-fpm   
**Nginx是什么:**   
Nginx (“engine x”) 是一个高性能的HTTP和反向代理服务器，也是一个IMAP/POP3/SMTP服务器。

**Php-fpm是什么:**   
1、cgi、fast-cgi协议

**cgi的历史**   
早期的webserver只处理html等静态文件，但是随着技术的发展，出现了像php等动态语言。   
webserver处理不了了，怎么办呢？那就交给**php解释器**来处理吧！   
交给php解释器处理很好，但是，php解释器如何与webserver进行通信呢？   
为了解决不同的**语言解释器(**如php、python解释器)与webserver的通信，于是出现了**cgi协议**。只要你按照cgi协议去编写程序，就能实现语言解释器与webwerver的通信。如php-cgi程序。

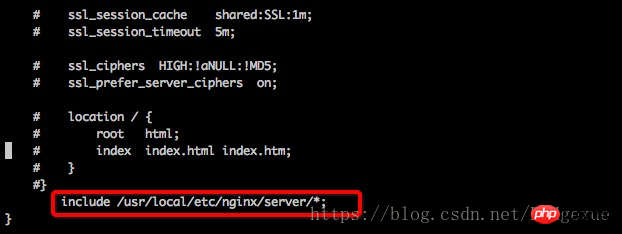
**fast-cgi的改进**   
有了cgi协议，解决了php解释器与webserver通信的问题，webserver终于可以处理动态语言了。但是，**webserver每收到一个请求，都会去fork一个cgi进程**，请求结束再kill掉这个进程。这样有10000个请求，就需要fork、kill php-cgi进程10000次。

有没有发现很浪费资源？   
于是，出现了cgi的改良版本，fast-cgi。fast-cgi每次处理完请求后，不会kill掉这个进程，而是保留这个进程，使这个进程可以一次处理多个请求。这样每次就不用重新fork一个进程了，大大提高了效率。

2、php-fpm是什么   
php-fpm即php-Fastcgi Process Manager.   
**php-fpm是 FastCGI 的实现**，并提供了进程管理的功能。   
进程包含 master 进程和 worker 进程两种进程。   
master 进程只有一个，负责监听端口，接收来自 Web Server 的请求，而 worker 进程则一般有多个(具体数量根据实际需要配置)，每个进程内部都嵌入了一个 PHP 解释器，是 PHP 代码真正执行的地方。

三、Nginx如何与Php-fpm结合   
上面我们说了，Nginx不只有处理http请求的功能，还能做反向代理。Nginx通过反向代理功能将动态请求转向后端Php-fpm。   
下面我们来配置一个全新的**Nginx+Php-fpm**

1、配置nginx.conf文件   
进入nginx目录下，编辑 nginx.conf文件。   
如图，在nginx.conf最后一行，添加include文件



2、添加对应的server   
进入上面include的路径，添加一个server.



下面我们解释一下配置项的含义：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16 | server {      listen       80; #监听80端口，接收http请求      server\_name  www.example.com; #就是网站地址      root /usr/local/etc/nginx/www/huxintong\_admin; # 准备存放代码工程的路径      #路由到网站根目录www.example.com时候的处理      location / {          index index.php; #跳转到www.example.com/index.php          autoindex on;      }        #当请求网站下php文件的时候，反向代理到php-fpm      location ~ \.php$ {        include /usr/local/etc/nginx/fastcgi.conf; #加载nginx的fastcgi模块          fastcgi\_intercept\_errors on;          fastcgi\_pass   127.0.0.1:9000; #nginx fastcgi进程监听的IP地址和端口      }  } |

总而言之：当我们访问www.example.com的时候，处理流程是这样的：

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | www.example.com        |          |        Nginx        |          |路由到www.example.com/index.php        |          |加载nginx的fast-cgi模块        |          |fast-cgi监听127.0.0.1:9000地址        |          |www.example.com/index.php请求到达127.0.0.1:9000          |          |       等待处理... |

下面我们启用php的php-fpm来处理这个请求

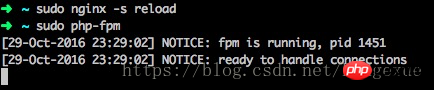
打开php-fpm.conf文件，我们看到如下配置：



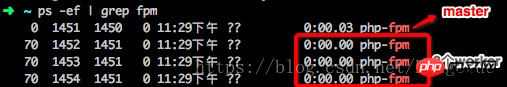
即:php-fpm模块监听127.0.0.1:9000端口，等待请求到来去处理。

四、总结   
nginx与php-fpm的结合，完整的流程是这样的。

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13 | www.example.com        |         |       Nginx        |         |路由到www.example.com/index.php        |         |加载nginx的fast-cgi模块        |         |fast-cgi监听127.0.0.1:9000地址        |         |www.example.com/index.php请求到达127.0.0.1:9000         |         |php-fpm 监听127.0.0.1:9000         |         |php-fpm 接收到请求，启用worker进程处理请求        |         |php-fpm 处理完请求，返回给nginx        |         |nginx将结果通过http返回给浏览器 |

五、效果展示   
1、启动nginx与php-fpm模块   


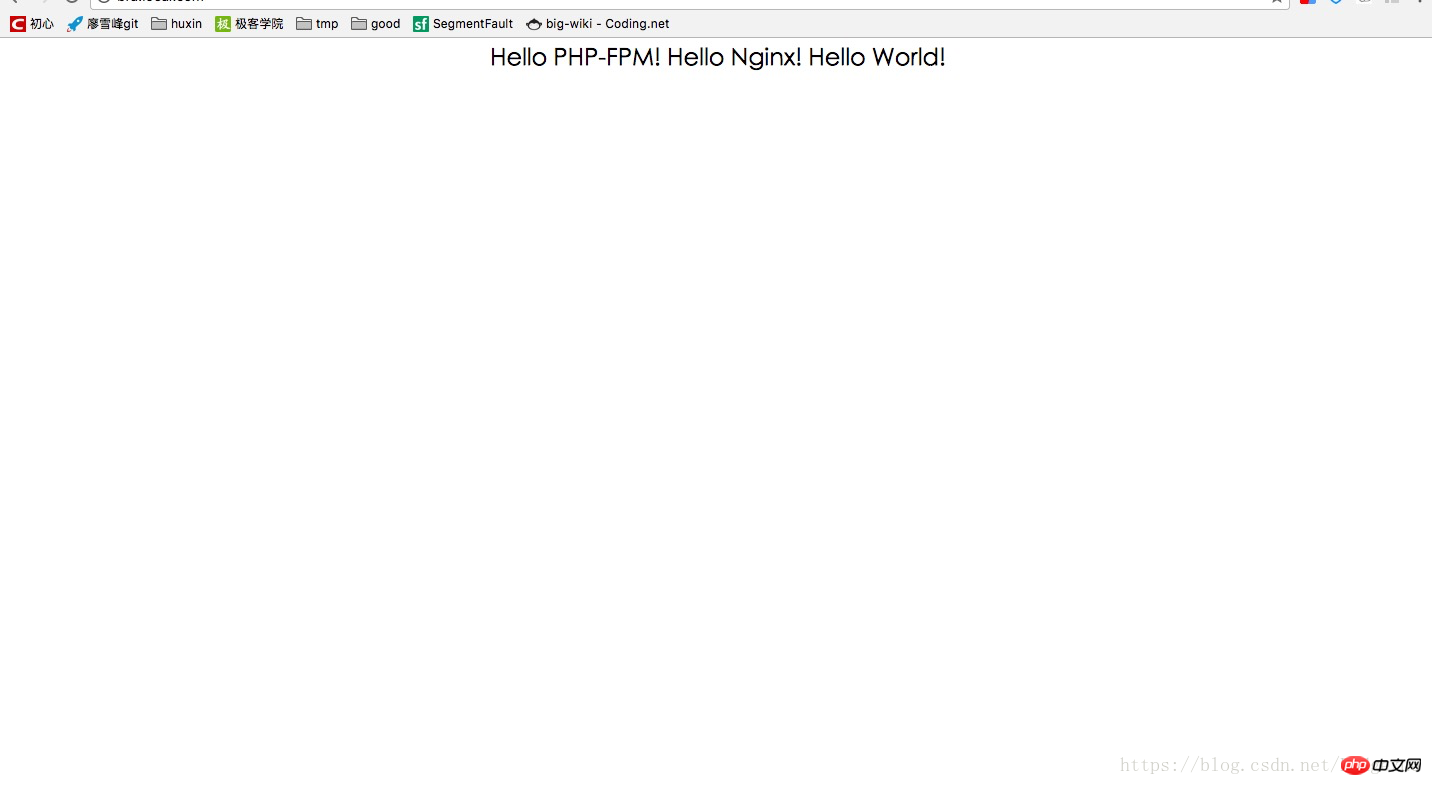
启动成功，我们查看php-fpm进程



如上图，有一个master进程，3个worker进程。

2、在网站目录下建立文件   
我们编辑文件如下图：

这里写图片描述

3、访问网站   


曾经在幽幽暗暗反反复复中追问 才知道平平淡淡从从容容才是真