

Linux平台PHP服务端开发——

第三讲 LNMP初步

目录

LNMP环境基本介绍

Nginx源码编译

管理Nginx和PHP

1

LNMP环境基本介绍

LNMP

- Linux + Nginx + MySQL/MariaDB + PHP/Python
- 简称为LNMP
- 本课程使用PHP语言
- Linux平台使用Ubuntu
- PHP和数据库之间的操作方式没有变化，这里主要讲述LNMP中的Nginx+PHP-FPM的模式

Nginx简介

- Nginx是俄罗斯程序员伊戈尔·赛索耶夫发明的HTTP/HTTPS和反向代理服务器。
- 目前Nginx也支持HTTP2, SMTP/POP3等协议。
- Nginx使用异步的方式处理请求，所以并发性能会很好。
- Nginx轻量级，高效，功能强大。
- Nginx的高性能依赖于系统异步接口的实现，在Linux, Unix上才能够体现。

Nginx for Windows

- Nginx在Windows只能用select接口，并且是作为Beta版本。

nginx for Windows

[Known issues](#)

[Possible future enhancements](#)

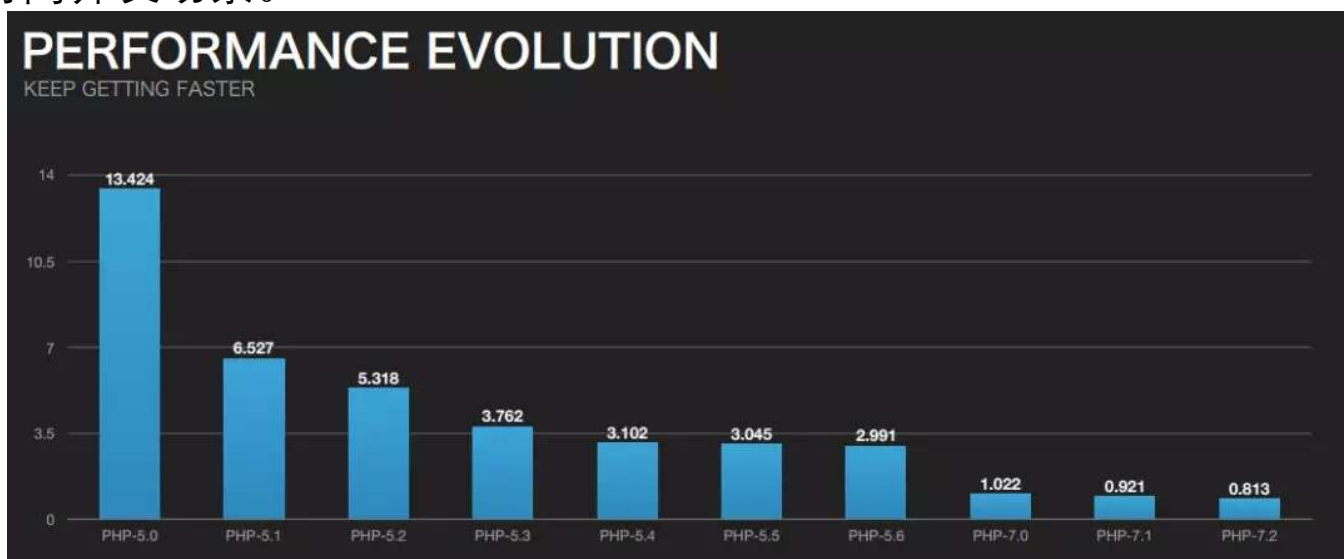
Version of nginx for Windows uses the native Win32 API (not the Cygwin emulation layer). Only the `select()` connection processing method is currently used, so high performance and scalability should not be expected. Due to this and some other known issues version of nginx for Windows is considered to be a *beta* version. At this time, it provides almost the same functionality as a UNIX version of nginx except for XSLT filter, image filter, GeoIP module, and embedded Perl language.

To install nginx/Windows, [download](#) the latest mainline version distribution (1.13.8), since the mainline branch of nginx contains all known fixes. Then unpack the distribution, go to the nginx-1.13.8 directory, and run `nginx`. Here is an example for the drive C: root directory:

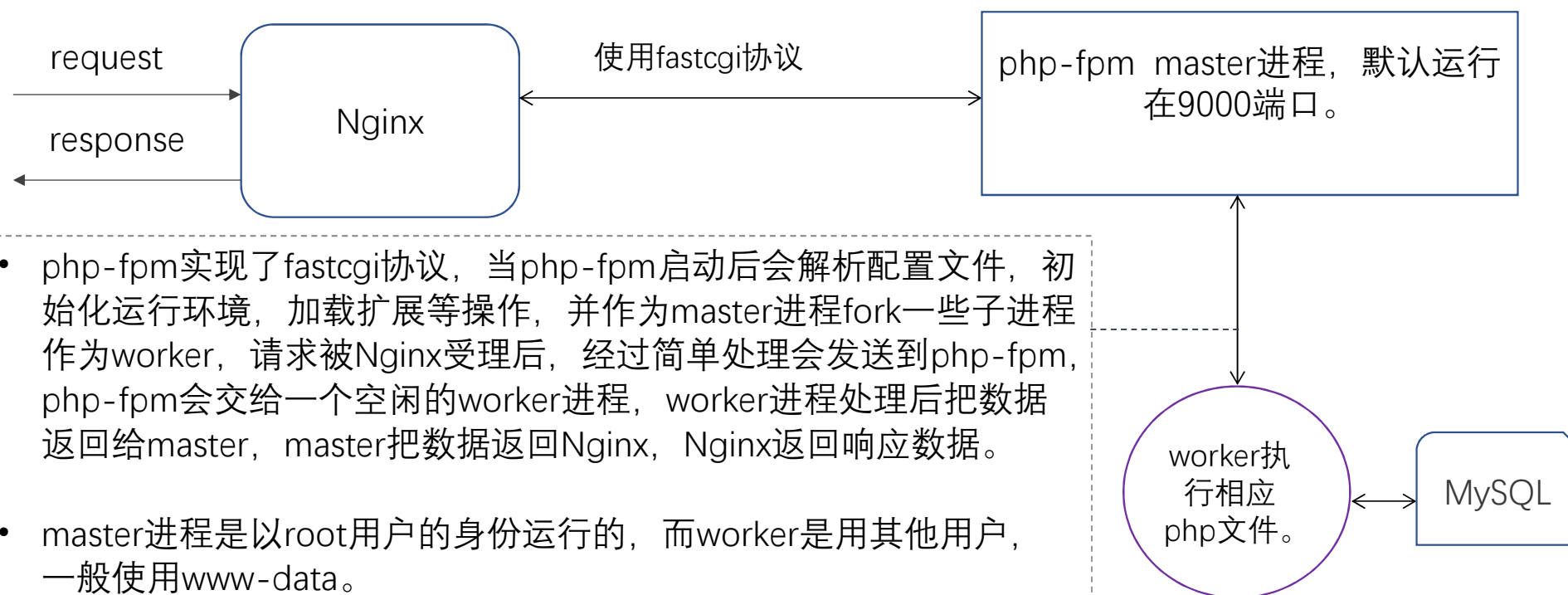
LNMP优势

- Nginx在Linux上基于epoll实现了异步IO模式，相比于Apache传统的方式，性能要高很多。同时，Nginx占用资源更少。
- PHP7性能极大提高，并且PHP支持热部署，PHP成熟的系统，框架等非常多，有大量的方案可以选择。
- PHP对于数据库的支持更好，主流数据库都支持，并且有大量的扩展可直接使用。
- 配置良好的LNMP环境能很好应对高并发场景。

！注意：虽然Nginx是异步的，但是PHP-FPM是用多进程的方式同步处理请求的。Nginx与PHP-FPM的配置方式简单灵活，并且Nginx对静态文件并不需要经过PHP-FPM处理。而PHP-FPM的可以通过配置文件控制进程数量，动态调整还是使用静态进程池。所以整体性能还是很高的。



Nginx+PHP的运行模式



- php-fpm实现了fastcgi协议，当php-fpm启动后会解析配置文件，初始化运行环境，加载扩展等操作，并作为master进程fork一些子进程作为worker，请求被Nginx受理后，经过简单处理会发送到php-fpm，php-fpm会交给一个空闲的worker进程，worker进程处理后把数据返回给master，master把数据返回Nginx，Nginx返回响应数据。
- master进程是以root用户的身份运行的，而worker是用其他用户，一般使用www-data。
- 注意：php-cgi和php-fpm是不同的，二者不相关，php-fpm可以根据配置启动时运行一些worker进程，php-fpm会根据配置设置动态调整，繁忙时创建多个进程，空闲时关闭一些进程，释放系统资源。

2

Nginx源码编译

获取Nginx源代码

- <http://nginx.org>
- mainline是当前主要开发版本， stable是稳定版本。
- 下载最新的stable版本。

源代码目录结构

目录/文件	说明
configure	编译初始化配置脚本
src	源代码
src/core	Nginx核心功能实现
src/event	Nginx事件支持实现
src/http	http协议实现
src/stream	1.9版本以后加入的模块，实现TCP的代理和负载均衡
src/http/v2	HTTP2协议的实现
src/http/modules	http功能模块实现

编译Nginx

- 脚本初始化：

```
./configure --prefix=/webrun/nginx1122 --with-http_gunzip_module --with-http_ssl_module --with-http_v2_module --with-openssl=../openssl-110g --with-pcre=../pcre-8.41 --with-http_gzip_static_module --with-http_auth_request_module --with-http_secure_link_module --with-http_image_filter_module --with-http_mp4_module --with-stream --with-stream_ssl_module --with-http_xslt_module
```

- 编译安装：

make install

如果是非root用户，使用sudo make install

Nginx安装目录结构

目录	说明
sbin	nginx程序所在目录
conf	配置文件目录
logs	日志目录， nginx.pid也在此目录， 记录nginx进程的PID。 error.log记录错误日志， access.log记录请求访问日志。

3

管理Nginx和PHP

Nginx配置文件

- 在安装目录下找到conf/nginx.conf
- 其中有一行配置pid logs/nginx.pid
这表示在nginx守护进程启动后会在logs目录下的nginx.pid文件记录当前运行的进程ID，通过kill命令可以终止此进程以及所有子进程。编写管理脚本要用到此文件。
- 其他配置可参考nginx官方文档。
- nginx.conf已经给出了基本的配置，在此基础上修改即可。

编写Nginx进程控制脚本

- Nginx要使用root身份运行，运行后，进程ID记录在nginx.pid文件。
- 处理过程就是通过nginx.pid来获取Nginx的PID并进行管理。
- 使用bash脚本实现：
 - 支持参数：start, stop, reload
 - start：检测nginx 是否在运行，没有则启动nginx。
 - stop：检测nginx是否运行，已经运行则获取nginx.pid的PID数据并kill进程ID。
 - reload：调用nginx -s reload即可。
- 具体代码参考脚本文件。

Nginx与PHP如何对接

- 在Nginx安装目录下的conf/nginx.conf文件中，server{……}结构中，配置如下所示。
- #注释的是另一种接入方式，PHP-FPM默认运行在9000端口，而unix:……这一行是使用sock文件的方式，传输性能更好。
- location 后面的是正则表达式，表明以请求URI以xxx.php/xxxx这样的形式都会给php-fpm处理。

```
location ~ /\.php(.*)$ {  
    #fastcgi_pass    127.0.0.1:9000;  
    fastcgi_pass     unix:/webrun/php-721/var/run/php-fpm.sock;  
    fastcgi_index    index.php;  
    fastcgi_split_path_info ^((?U).+\.php)(/?.+)$;  
    fastcgi_param    PATH_INFO    $fastcgi_path_info;  
    fastcgi_param    PATH_TRANSLATED $document_root$fastcgi_path_info;  
    fastcgi_param    SCRIPT_FILENAME $document_root$fastcgi_script_name;  
    include          fastcgi_params;  
}
```

编写控制脚本同时管理Nginx和php-fpm

- 由于已经实现了Nginx和php-fpm的管理脚本，这里使用一个脚本调用两个脚本即可。
- 支持的参数：start, stop, restart, ngx-reload, fpm-reload
 - start：实现方式只需要运行Nginx和php-fpm管理脚本传入start参数
 - stop：和start处理方式相同
 - restart：和start处理方式相同
 - ngx-reload：调用Nginx管理脚本传入reload参数
 - fpm-reload：调用php-fpm管理脚本传入reload参数
- 具体代码不便在ppt展示，参考bash脚本文件。