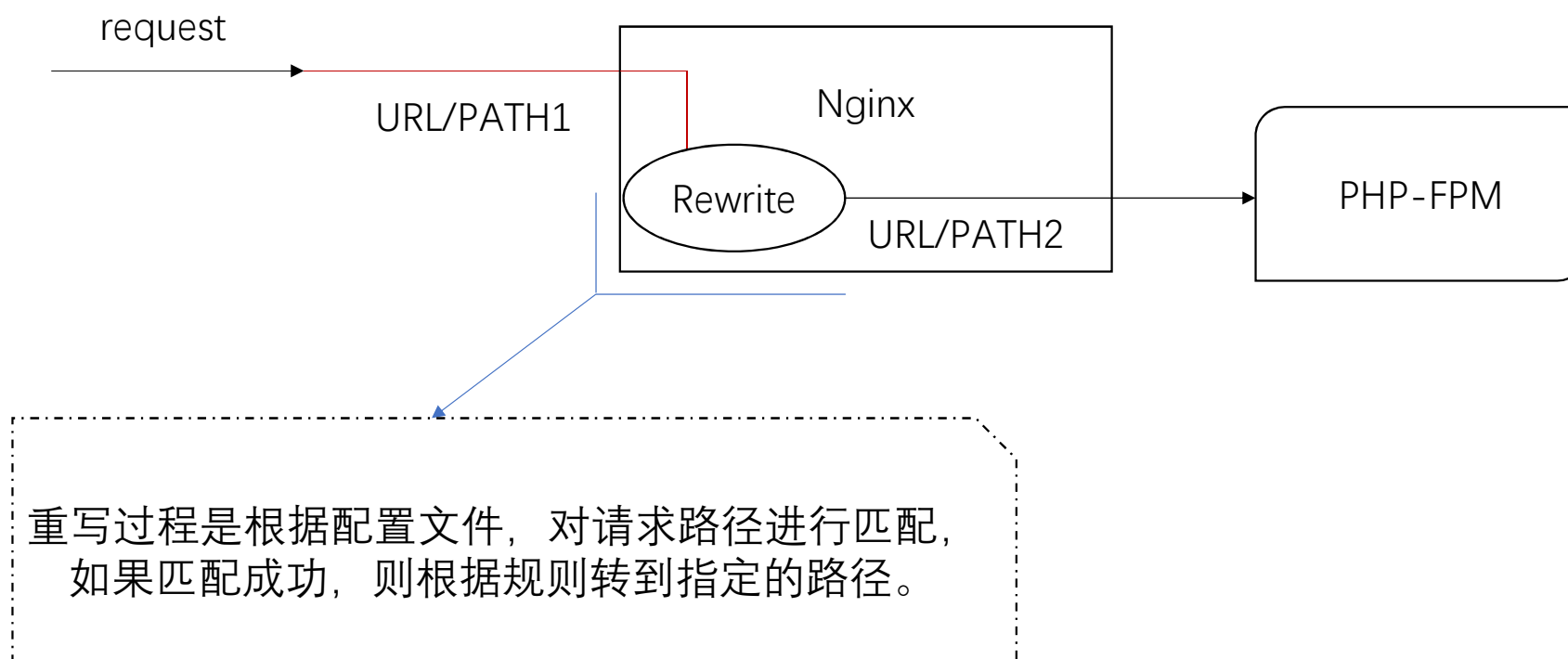


Linux平台PHP服务端开发——

第十六讲 LNMP更多细节以及系统服务

Nginx路由重写



Nginx路由重写：rewrite

- Nginx的rewrite主要实现路由重写功能，需要PCRE支持：用到perl的正则表达式语法进行规则匹配。

- rewrite语法结构：

rewrite [REGEX] [REPLACEMENT] [FLAG]
正则表达式 要替换的内容 标志

flag：

last	URL重写后后，再次发起新的请求匹配location块，地址栏URL不变
break	直接使用当前资源，不再匹配location，地址栏URL不变
redirect	302临时重定向，地址栏显示重定向的URL，爬虫不更新数据
prement	301永久重定向，地址栏URL显示重定向后的URL

Ngixn : 检测参数

- ~ 区分大小写的正则匹配, ~* 不区分大小写的正则匹配。!~表示正则不匹配。
 - -f and !-f 检测文件是否存在。
 - -d and !-d 检测目录是否存在。
 - -e and !-e 检测文件, 目录, 符号链接是否存在。
 - -x and !-x 检测是否是可执行文件。
-
- 使用正则表达式, 如果使用(), 小括号匹配的内容可以通过\$1~\$9获取。

rewrite示例

- rewrite只能放在server, location, if的配置块中。

- rewrite配置：

```
server{
    .....;
    location / {
        index index.php index.html;
        if (!-e $request_filename) {
            rewrite ^(.*)$ /index.php$1 last;
            break;
        }
    }
    location ~ ^\.php(.*)$ {
        .....;
    }
}
```

- 此配置会检测如果请求的路径不是一个存在的文件则进行路由重写至/index.php，而rewrite会匹配新的location，匹配到则执行。

PHP-FPM配置文件

- 在php-fpm的配置文件www.conf中 (PHP_INSTALL/etc/php-fpm.d/www.conf) , 有几行配置用于控制php-fpm的进程：

```
pm = dynamic
pm.max_children = 10
pm.start_servers = 3          //when pm is set to dynamic
pm.min_spare_servers = 3     //when pm is set to dynamic
pm.max_spare_servers = 10    //when pm is set to dynamic
```
- 当pm = dynamic 则会在启动时根据start_servers设置值启动子进程，而min_spare_servers和max_spare_servers分别限制了最小空闲子进程和最大空闲子进程，max_spare_servers不能超过max_children的设置值。
- 当pm = static 则直接启动pm.max_children设置值的子进程处理请求。

并发压力测试监控LNMP负载变化

- 本地服务器IP地址为192.168.56.104，使用ab进行并发测试：
`ab -c 100 -n 10000 '192.168.56.104:9009/'`
- 观察并发测试前后的负载情况：

并发压力测试前负载情况

```
root      13102      1 ?          php-fpm    php-fpm: master process (/web
/etc/php-fpm.conf)
www-data  13103  13102 ?          php-fpm    php-fpm: pool www
www-data  13104  13102 ?          php-fpm    php-fpm: pool www
www-data  13105  13102 ?          php-fpm    php-fpm: pool www
```

```

1      ||                               1.3%    Tasks: 54, 63 thr; 1 running
2      |                               2.0%    Load average:      0.01 0.00
Mem    ||||||| ||||||| ||||||| ||||||| 229M/985M    Uptime: 04:25:37
Swp    |                               0K/1.92G

```


并发压力测试中负载情况

```
/usr/sbin/php-fpm: pool www:
www-data 13103 13102 ?      php-fpm      php-fpm: pool www
www-data 13104 13102 ?      php-fpm      php-fpm: pool www
www-data 13105 13102 ?      php-fpm      php-fpm: pool www
www-data 14442 13102 ?      php-fpm      php-fpm: pool www
www-data 14450 13102 ?      php-fpm      php-fpm: pool www
www-data 14451 13102 ?      php-fpm      php-fpm: pool www
www-data 14472 13102 ?      php-fpm      php-fpm: pool www
www-data 14488 13102 ?      php-fpm      php-fpm: pool www
www-data 14958 13102 ?      php-fpm      php-fpm: pool www
www-data 14974 13102 ?      php-fpm      php-fpm: pool www
```

```
1  ||||| 97.4%
2  ||||| 92.6%
Mem ||||| 252M/985M
Swp  ||||| 0K/1.92G
```

```
Tasks: 61, 63 thr; 2 running
Load average: 0.15 0.05
Uptime: 04:27:54
```

Linux系统服务

- 让nginx, php-fpm, mysqld开机自动启动。
- 这需要把管理脚本加入到Linux启动脚本。

系统服务管理命令：service

- `service --status-all`显示所有的服务。
- 使用示例：`service ufw restart`
- 也可以直接运行脚本文件：
`sudo /etc/init.d/networking restart`

系统启动初始化脚本 (init scripts)

- /etc/init.d是系统服务的启动程序，一般都是脚本程序。长期运行的服务进程通常是脚本程序去调用指定的程序运行。
- 但是系统启动并不会直接启动这个目录下的服务程序。
- 原因在于Linux系统有运行级别的问题，每个启动级别运行的服务是不同的。Linux在启动时会根据配置文件设置的启动级别查找相对应的目录启动程序。

系统运行级别

- Linux系统有7个运行级别：
 - 0 : 关机，默认级别不能是0，否则不能正常启动
 - 1 : 单用户模式，用于系统维护，不能远程登录
 - 2 : 多用户模式，不支持NFS
 - 3 : 多用户模式，支持NFS
 - 4 : 系统保留，未定义
 - 5 : 界面模式，开机后进入桌面环境
 - 6 : 重启，默认运行级别不能是6否则会不断重启
- 查看运行级别的命令：`runlevel`

系统运行级别所对应的/etc/rc*.d目录

- Linux如何让每个运行级别对应不同的系统服务？
 - /etc/存在rc*.d 目录，*是0-6的数字， /etc/init.d目录下的服务程序只是一个集合，系统启动时，会根据级别运行rc*.d目录的程序，而此目录下是链接到/etc/init.d的符号链接，这样就实现了不同级别的不同服务配置。
 - 注意这里有rc0.d和rc6.d，这两个目录下的符号链接是K开头，而其他的是S开头，K开头的是不启动的，而是在离开此运行级别时要进行的操作，比如关机要进行一些清理工作。
 - 符号链接S/K后面跟的数字是启动顺序，数字小的先运行。
 - Linux在执行时，会识别脚本中的start，stop参数。所以在启动时，会默认执行start参数的操作。

nginx, php-fpm, mysqld加入系统服务

- 对已经写好的LNMP管理脚本，分别把nginx, php-fpm, mysqld加入Linux系统服务：

复制nginx, php-fpm, mysqld脚本到/etc/init.d/

以nginx为例，建立软链接：

```
sudo ln -s /etc/init.d/nginx /etc/rc3.d/S05nginx
```

```
sudo ln -s /etc/init.d/nginx /etc/rc5.d/S05nginx
```

此操作会在用户启动级别为3和5的时候自动启动nginx