**Launcher 高并发处理**

**现象：**

查看php-fpm进程ps -ef | grep php-fpm |wc -l

发现php-fpm进程不断上涨，很快接近设置的最大值，然后服务器就会一直报错。

**解决步骤：**

**步骤1**. http原则是少请求，少返回。这里控制返回的数据：减少/zyp/api/datas接口的返回数据，每次6条

此时服务器可以坚持2小时

**步骤2**.

设置php-fpm进程的数量

修改文件php-fpm.conf文件

pm = dynamic

pm.max\_children = 200

pm.start\_servers = 18

pm.min\_spare\_servers = 6

pm.max\_spare\_servers = 30

其中

max\_spare\_servers< max\_children,

start\_servers = min\_spare\_servers +( max\_spare\_servers- min\_spare\_servers) /2

严格遵守公式

可以坚持一个晚上。

**步骤3**.

对访问慢的接口设置超时限制，在发送请求接口的地方加上

curl\_setopt($ch, CURLOPT\_TIMEOUT,1);

如果超过1秒，请求终止，返回空。

这时 php-fpm 观察4个小时没有增加进程，基本达到控制目的。

**步骤4**.

针对访问慢的接口单独处理，单独开php-fpm进程处理。

**A．设置nginx配置**。如下：

文件launcher.szprize.cn.conf增加

location = /zyp/api/datas

{

if ( !-e $request\_filename){

rewrite ^/index.php(.\*)$ /index.php?s=$1 last;

rewrite ^(.\*)$ /index.php?s=$1 last;

break;

}

fastcgi\_pass 127.0.0.1:9001;

fastcgi\_param SCRIPT\_FILENAME /mnt/www/launcher.szprize.cn$fastcgi\_script\_name;

fastcgi\_param RUN\_ENV\_TEST 0;

include fastcgi\_params;

}

**B：设置php**

php-fpm.conf文件中增加

[curl]

listen = 127.0.0.1:9001

user=www

group=www

pm = dynamic

pm.max\_children = 200

pm.start\_servers = 18

pm.min\_spare\_servers = 6

pm.max\_spare\_servers = 30

php-fpm进程数不增加，达到控制目的！