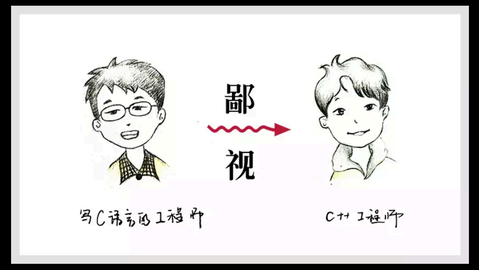
PHP常驻进程编程须知

—郭新华

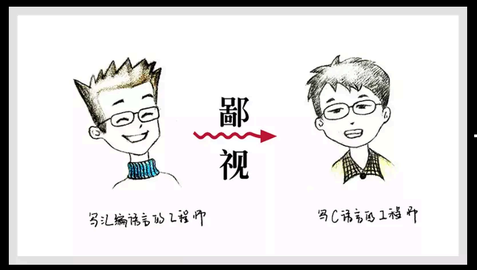
PHPCon 历年完整 PPT 下载站：

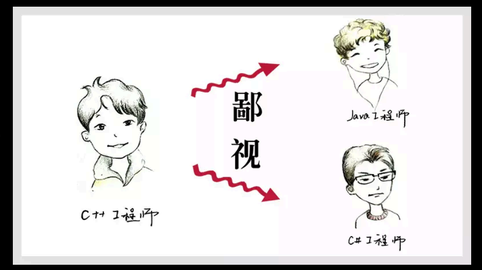
https://github.com/ThinkDevelopers/PHPConChina

程序员鄙视链



PHP是最好的编程语言！！！！！



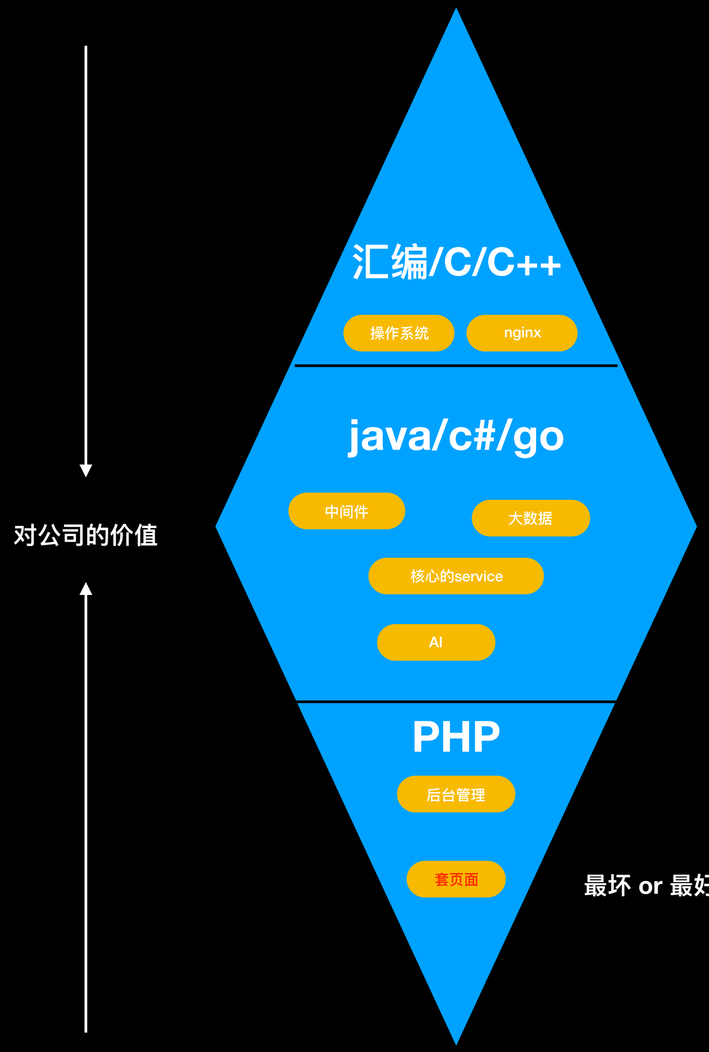


图片引用：https://segmentfault.com/a/1190000012028295

PHP的官方定位

PHP is a popular general-purpose scripting language that is especially suited to web development.

PHP在中大型公司中的定位



最坏 or 最好的时代

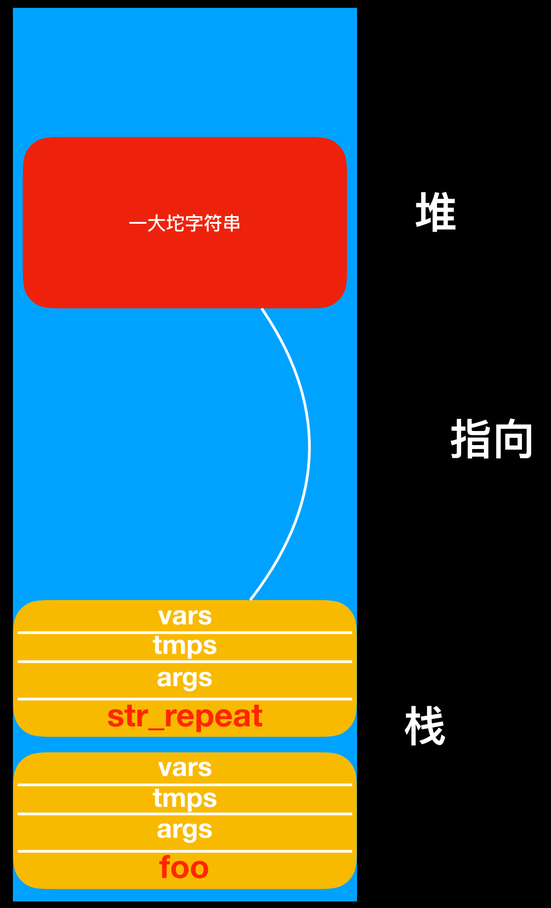
PHPer普遍缺少的技能

1. 不知道什么是内存泄漏
2. 不知道什么是通讯协议
3. 不知道如何进行性能优化
4. 不知道什么是连接池
5. 不知道算法和数据结构

1.内存篇

PHP的堆和栈

```php
function foo($cond){
    if($cond){
        $var = str_repeat("a big string", 1024);
    }
    return 1;
}
foo(1);
```



常见例子

情况1：返回值

```php
function foo($cond){
    if($cond){
        $var = str_repeat("a big string", 1024);
    }
    return $var;
}
```

情况2：全局变量

```php
function foo($cond){
    if($cond){
        $var = str_repeat("a big string", 1024);
        $GLOBALS['key'] = $var;
    }
    return 1;
}
```

情况3：静态属性

```php
function foo($cond){
    if($cond){
        $var = str_repeat("a big string", 1024);
        classB::$pro = $var;
    }
    return 1;
}
```

PHP到底应该怎么释放内存

```php
function foo($cond){
    if($cond){
        $var = str_repeat("a big string", 1024);
    }
    sleep(1000);
    return 1;
}
foo(1);
```

结论： 手动释放 or opcache不用管

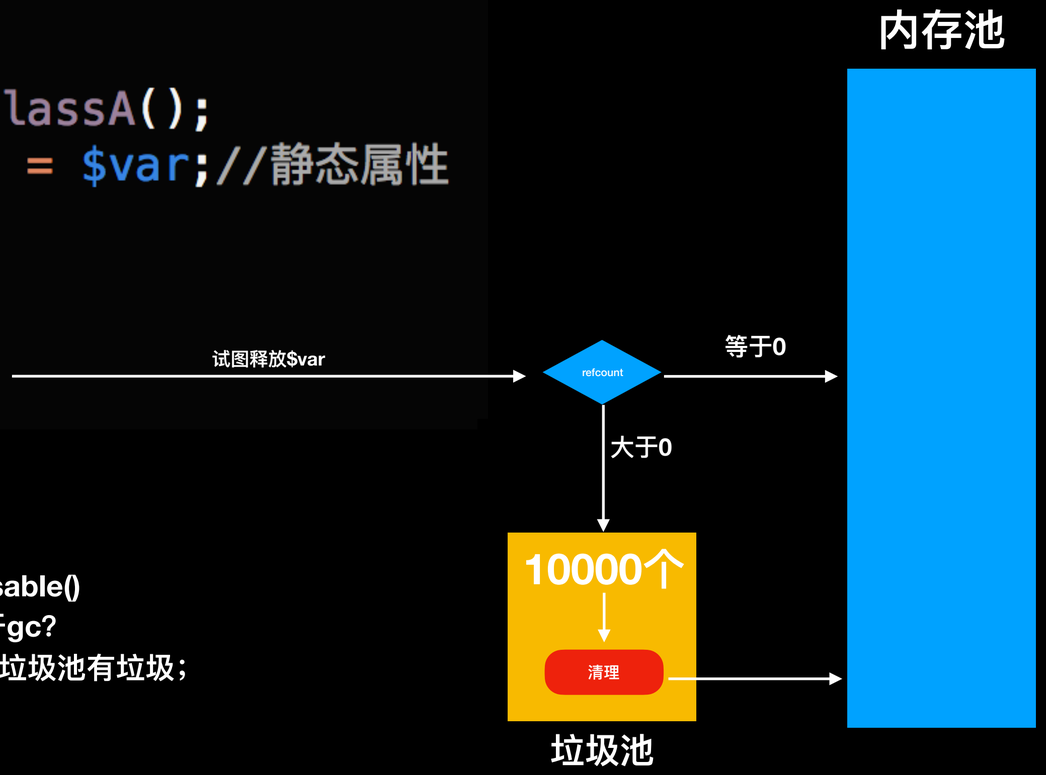
PHP到底应该怎么释放内存

```php
function foo($cond){
    if($cond){
        $var = str_repeat("a big string", 1024);
    }
    unset($var);//unset?
    $var = null;//=null?  还是需要都调用
    sleep(1000);
    return 1;
}
foo(1);
```

unset还是赋值null？？

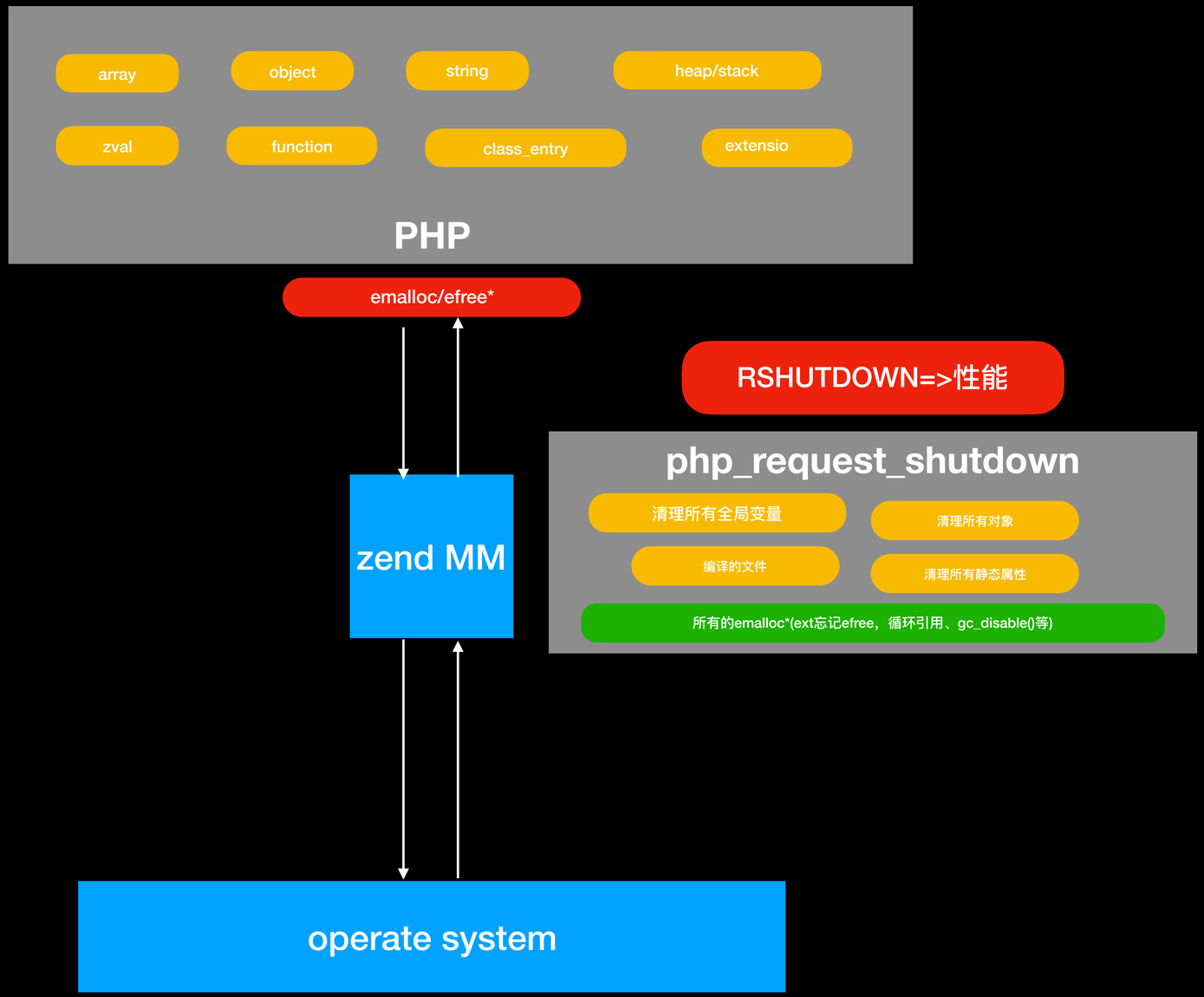
不得不说的循环引用

```php
function foo($cond){
    if($cond){
        $var = new classA();
        classB::$pro = $var;//静态属性
    }
    return 1;
}
echo foo(1);
```



- composer事件之gc_disable()
- 我们的项目到底要不要开gc?
- gc_disable()；请求结束垃圾池有垃圾；

FPM的黑魔法—php_request_shutdown



常驻进程没有php_request_shutdown我们应该怎么办？？

1.max_request重启

(1)所有客户端重连(base模式)

(2)内存峰值大

(3)无法常驻内存

(4)第一次请求慢(opcache,ext minits)

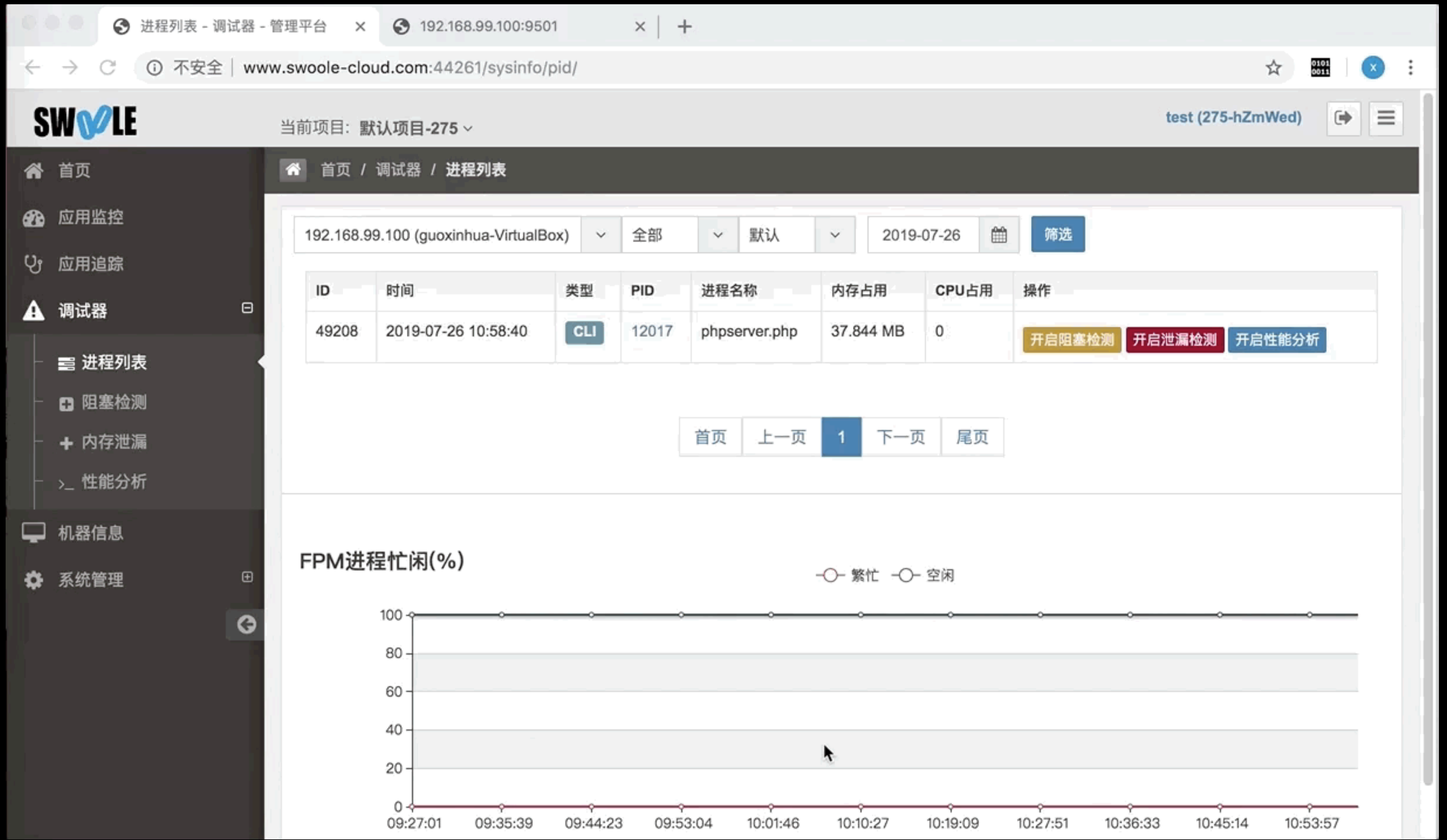
2.手动unset所有栈外引用

我们需要工具

```php
$http = new swoole_http_server("0.0.0.0", 9501, SWOOLE_BASE);

class classA{
    public $pro;
    public function __construct(){
        $this->pro = &$GLOBALS['key'];
    }
}

$http->on('request', function($req,$response){
    $response->end("<h1>Hello Swoole.</h1>");
    $obj = new classA();
    $obj->pro['string_key'] = "the huge string";
});

$http->start();
```

- 引申：不敢随便赋值和传递；担心忘记unset

利用工具定位泄漏


进程列表 - 调试器 - 管理平台
192.168.99.100:9501
不安全 | www.swoole-cloud.com:44261/sysinfo/pid/
SWOOLE
当前项目: 默认项目-275
test (275-hZmWed)
首页
应用监控
应用追踪
调试器
进程列表
阻塞检测
内存泄漏
性能分析
机器信息
系统管理
首页 / 调试器 / 进程列表
192.168.99.100 (guoxinhua-VirtualBox)
全部
默认
2019-07-26
筛选
ID
时间
类型
PID
进程名称
内存占用
CPU占用
操作
49208
2019-07-26 10:58:40
CLI
12017
phpserver.php
37.844 MB
0
开启阻塞检测
开启泄漏检测
开启性能分析
首页
上一页
1
下一页
尾页
FPM进程忙闲(%)
繁忙
空闲
100
80
60
40
20
0
09:27:01
09:35:39
09:44:23
09:53:04
10:01:46
10:10:27
10:19:09
10:27:51
10:36:33
10:45:14
10:53:57

我们需要工具

```php
$http = new swoole_http_server("0.0.0.0", 9501, SWOOLE_BASE);

class classA{
    public $pro;
    public function __construct(){
        $this->pro = &$GLOBALS['key'];
    }
}

$http->on('request', function($req,$response){
    $response->end("<h1>Hello Swoole.</h1>");
    $obj = new classA();
    $obj->pro['string_key'] = "the huge string";
    unset($obj->pro['string_key']);
});

$http->start();
```

其他应用场景—协程脏读

```php
$_array = [];
$serv->on("Request", function ($req, $resp){
    global $_array;
    //请求 /a (协程 1 )
    if ($request->server['request_uri'] == '/user_id=1') {
        $_array['name'] = '张三';
        co::sleep(1.0);//中断本次请求 保存函数状态=>挂起
        echo $_array['name'];//脏读
        $resp->end($_array['name']);
    }
    //请求 /b (协程 2 )
    else {
        $_array['name'] = '李四';
        $resp->end();
    }
});
```

其他应用场景—FPM迁移代码到swoole

```php
if(!isset(classA::$pro)){
    classA::$pro = $_GET['param'];
}

echo classA::$pro;
```

2.性能篇

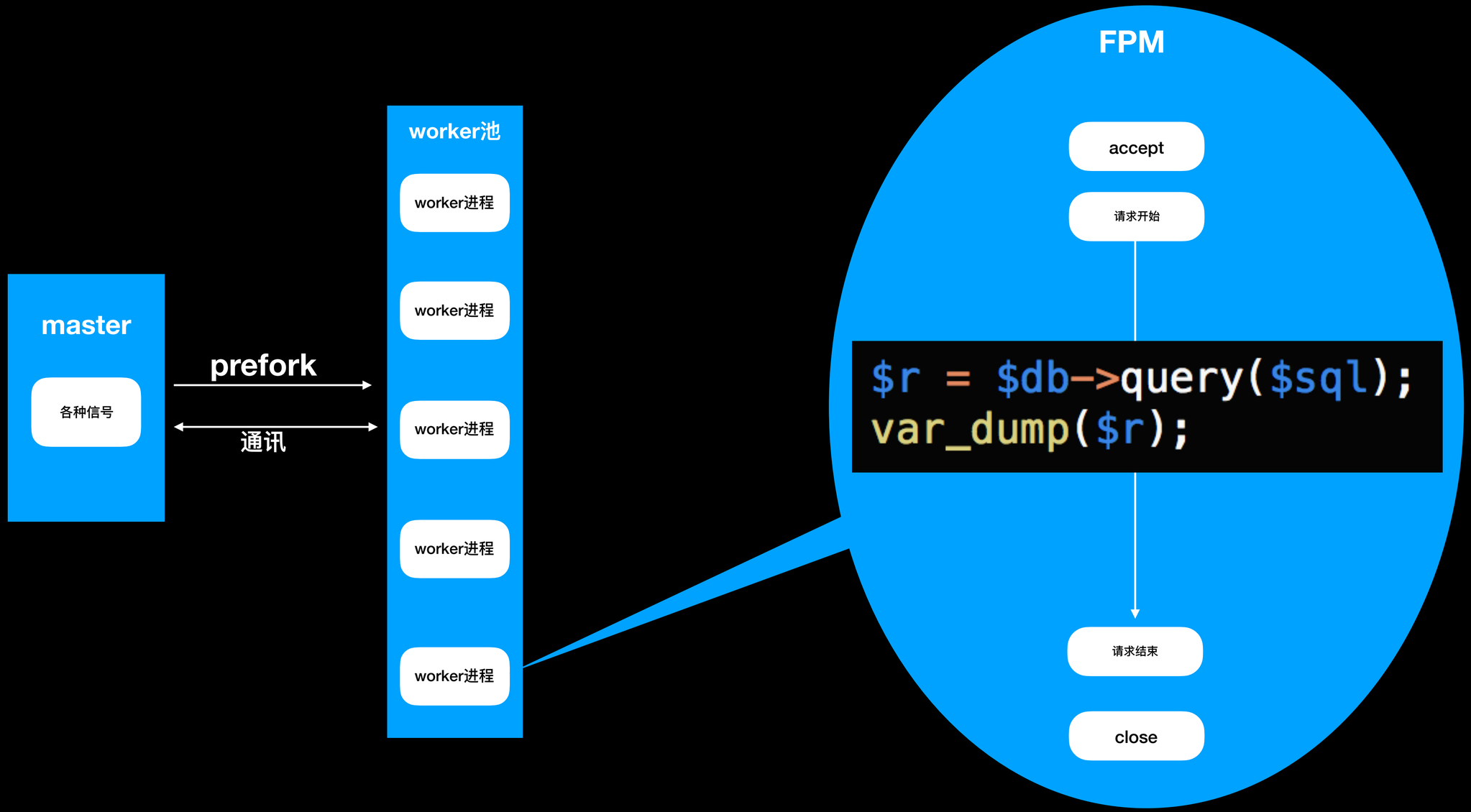
机器比人便宜，加机器就好了

别扯了 瓶颈肯定在mysql

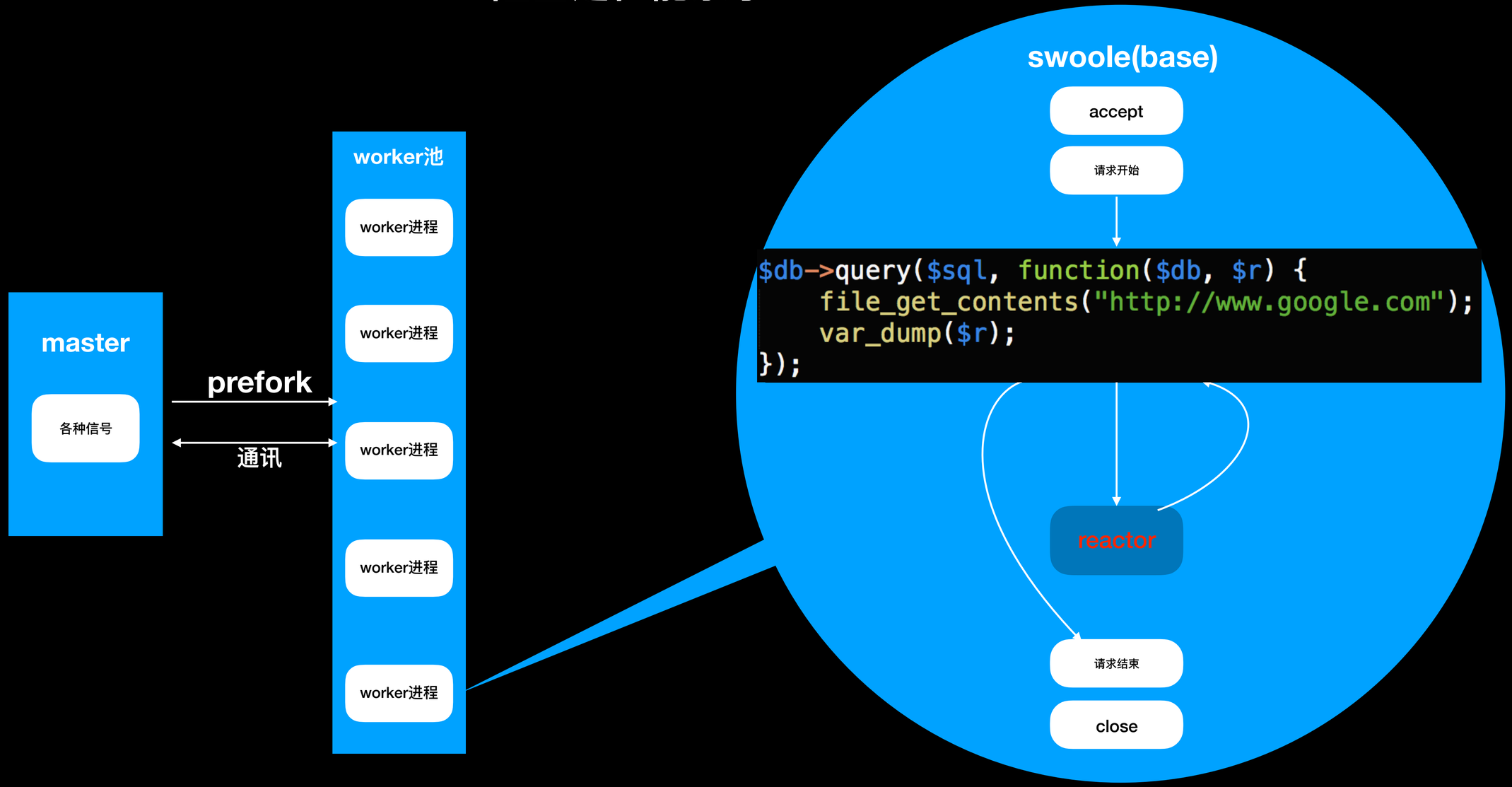
PHP这语言性能高不了(e.g php7_warpper)

做基础架构:我这个接口为啥这么慢，帮看看呗

阻塞是性能杀手—FPM


master
各种信号
prefork
通讯
worker池
worker进程
worker进程
worker进程
worker进程
worker进程
FPM
accept
请求开始
$r = $db->query($sql);
var_dump($r);
请求结束
close

阻塞是性能杀手—Swoole



一个具体的阻塞例子

```php
$http = new swoole_http_server("0.0.0.0", 9501, SWOOLE_BASE);

$http->on('request', function($req,$response){
                $opts=array(
                        "http"=>array(
                                "method"=>"GET",
                                "timeout"=>1
                         ),
                );
                $context = stream_context_create($opts);

                file_get_contents("http://www.google.com", false, $context);

                $response->end("<h1>Hello Swoole.</h1>");

});

$http->start();
```

用工具定位阻塞

第一步：strace -p 123456

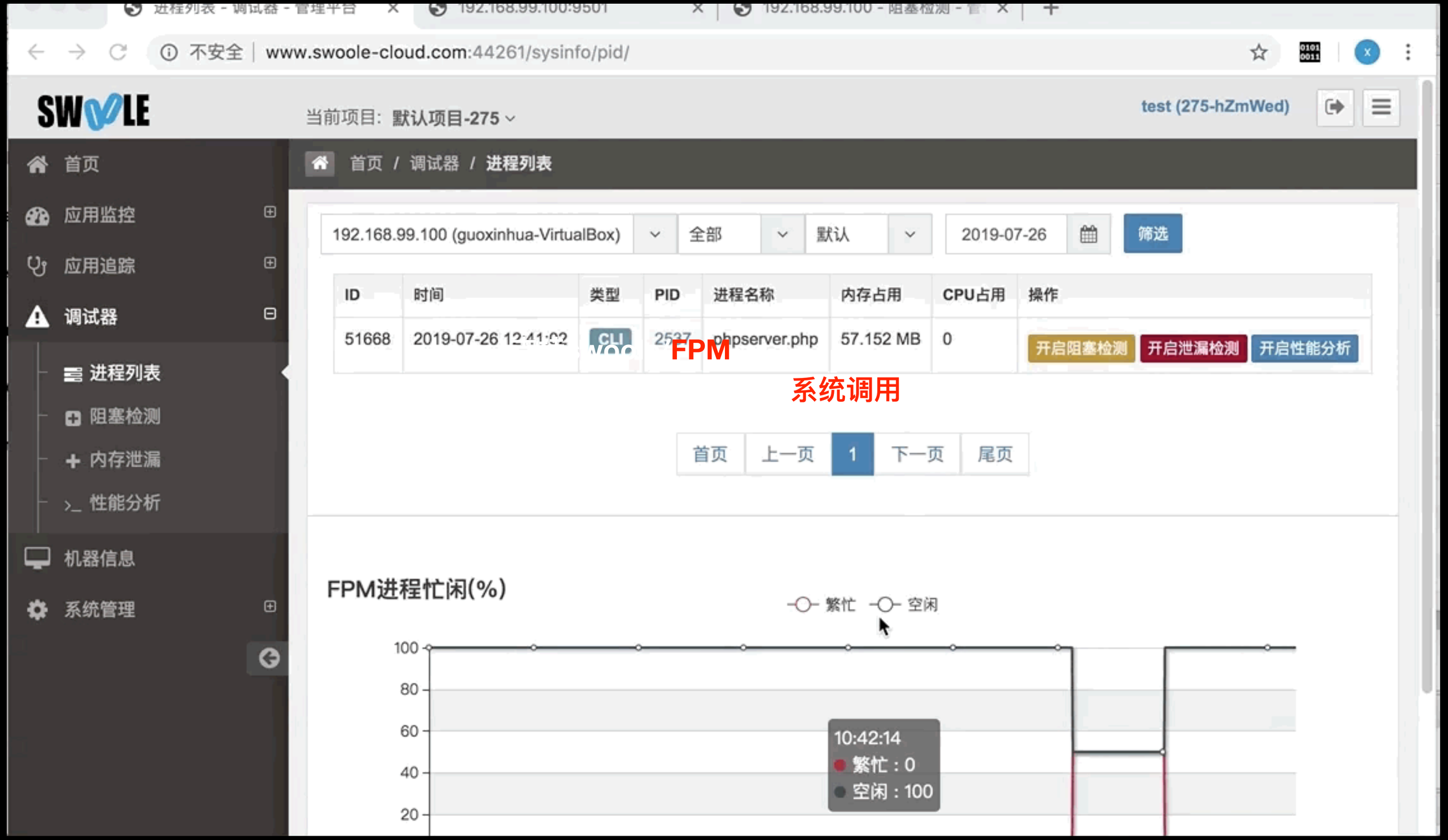
```
mprotect(0x7f2a805e7000, 4096, PROT_READ) = 0
munmap(0x7f2a8c53f000, 142071)          = 0
stat("/etc/resolv.conf", {st_mode=S_IFREG|0644, st_size=172, ...}) = 0
open("/etc/resolv.conf", O_RDONLY|O_CLOEXEC) = 3
fstat(3, {st_mode=S_IFREG|0644, st_size=172, ...}) = 0
mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0)
read(3, "# Dynamic resolv.conf(5) file fo"..., 4096) = 172
read(3, "", 4096)                       = 0
close(3)                                = 0
munmap(0x7f2a8c67f000, 4096)            = 0
uname({sys="Linux", node="guoxinhua-VirtualBo...", ...}) = 0
socket(PF_INET, SOCK_DGRAM|SOCK_NONBLOCK, IPPROTO_IP) = 3
connect(3, {sa_family=AF_INET, sin_port=htons(53), sin_addr=inet_addr("1
poll([{fd=3, events=POLLOUT}], 1, 0)    = 1 ([{fd=3, revents=POLLOUT}])
sendmmsg(3, {{{msg_name(0)=NULL, msg_iov(1)=[{"ik\1\0\0\1\0\0\0\0\0\0\3w
n=0, msg_flags=0}, 32}}, 2, MSG_NOSIGNAL) = 2
poll([{fd=3, events=POLLIN}], 1, 5000
```

第二步：lsof -p 123456

```
php     18498 root  mem    REG     8,1   149120 7340376 /lib/x86_64-linux-gnu/ld-2.19.so
php     18498 root    0u   CHR   136,2      0t0       5 /dev/pts/2
php     18498 root    1u   CHR   136,2      0t0       5 /dev/pts/2
php     18498 root    2u   CHR   136,2      0t0       5 /dev/pts/2
php     18498 root    3u  IPv4 3557208      0t0     TCP 10.0.4.15:8500->www.google.com:http (SYN_SENT)
```

很多时候fd没有具体的标志
strace不好做数字量化
不知道PHP层的调用堆栈

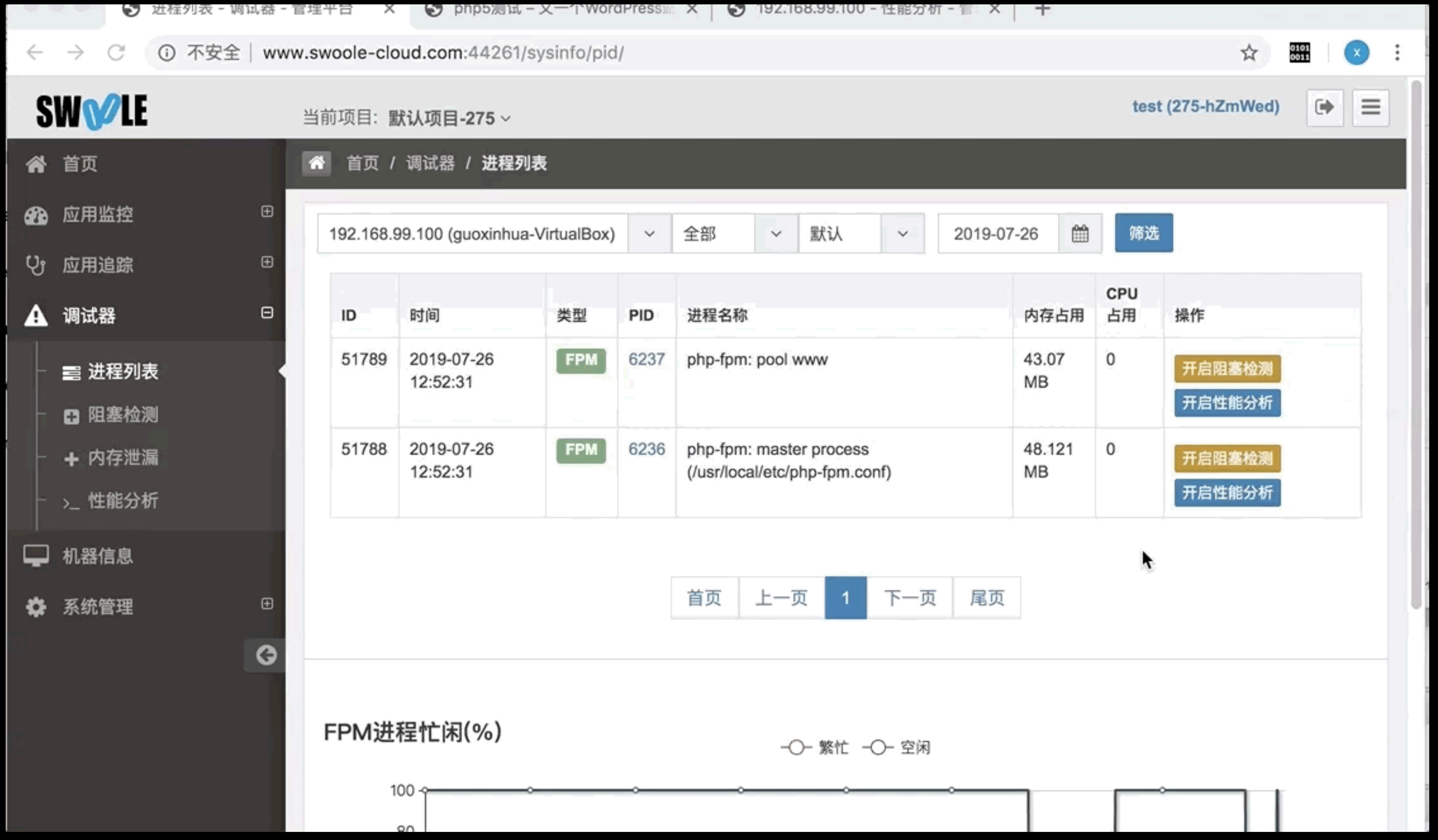
用工具定位阻塞


不安全 www.swoole-cloud.com:44261/sysinfo/pid/
SWOOLE
当前项目: 默认项目-275
test (275-hZmWed)
首页
应用监控
应用追踪
调试器
进程列表
阻塞检测
内存泄漏
性能分析
机器信息
系统管理
首页 / 调试器 / 进程列表
192.168.99.100 (guoxinhua-VirtualBox)
全部
默认
2019-07-26
筛选
ID
时间
类型
PID
进程名称
内存占用
CPU占用
操作
51668
2019-07-26 12:41:02
CLI
2527
phpserver.php
57.152 MB
0
开启阻塞检测
开启泄漏检测
开启性能分析
FPM
系统调用
首页
上一页
1
下一页
尾页
FPM进程忙闲(%)
繁忙
空闲
100
80
60
40
20
10:42:14
繁忙 : 0
空闲 : 100

除了mysql慢查询也会性能问题！！！

```php
/*
 *  传入股票代码 返回股票信息
 */
function getStock($code){
    $all_stock = apcu_fetch('all_stock');
    $stocks = json_decode($all_stock,true);
    return $stocks[$code];
}

getStock("300104");
```

profile工具—Xhprof



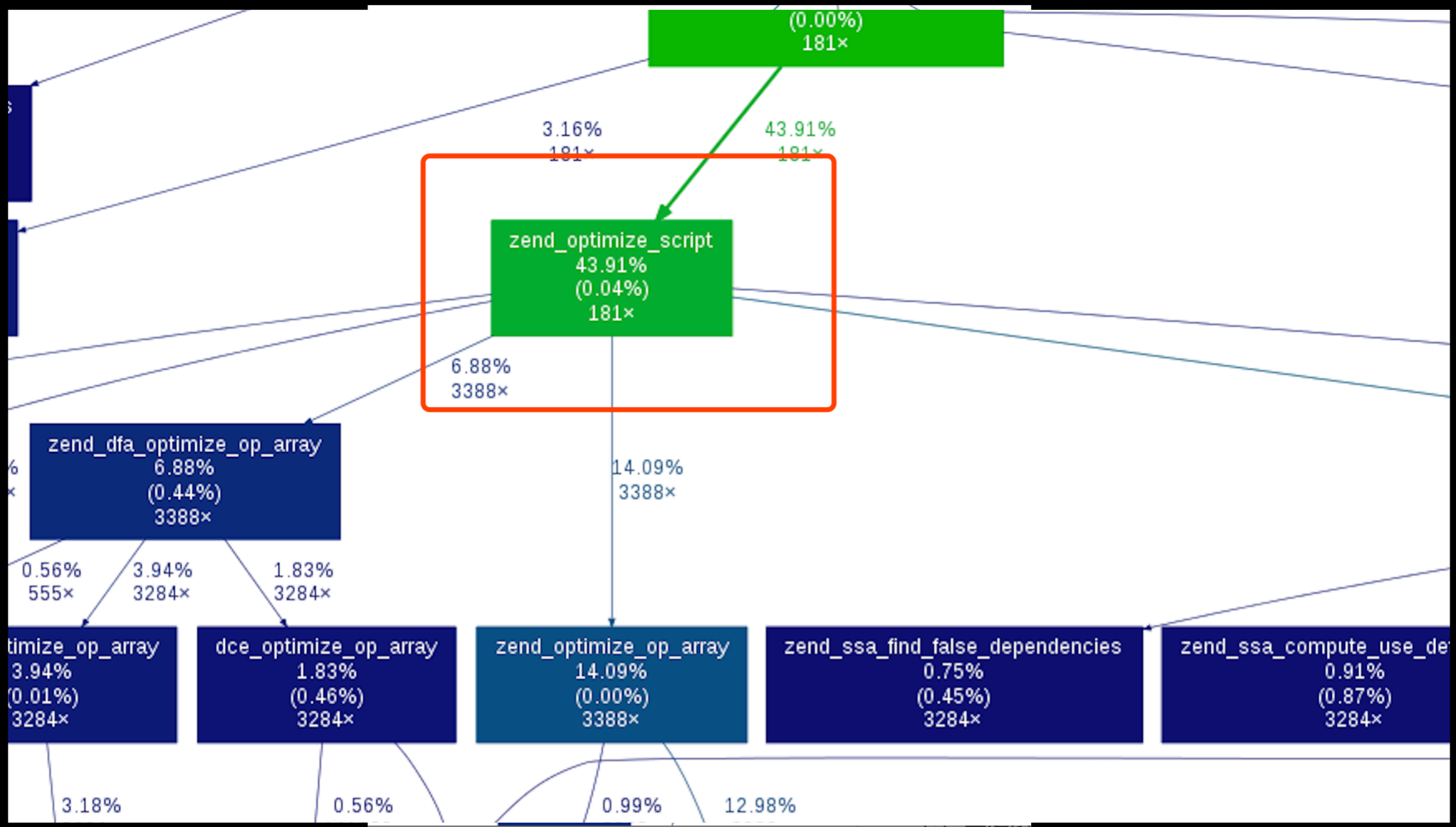
高性能总结

好的语言，框架等是前提：
高性能都是根据实际业务profile压榨出来的

PHP生态：
扩展层极致性能，PHP层极致开发效率

c语言profile工具—callgrind

```
valgrind --tool=callgrind php server.php
```



c语言profile工具—perf

```
void foo(){
    int i,j;
    for(i=0;i<=100000000;i++){
            j = 0;
    }
}

void main(){
    foo();
}

gcc -o test -O0 -g test.c
perf record -a -g ./test
perf report -g
```

```
Percent |
        |
        |  Disassembly of section .text:
        |
        |  00000000000005fa <foo>:
        |  foo():
        |  #include <stdio.h>
        |
        |  void foo(){
        |    push   %rbp
        |    mov    %rsp,%rbp
        |           int i,j;
        |           for(i=0;i<=100000000;i++){
        |    movl   $0x0,-0x4(%rbp)
        |  ↓ jmp    18
        |                   j = 0;
        |  d:  movl   $0x0,-0x8(%rbp)
        |           for(i=0;i<=100000000;i++){
  18.26 |      addl   $0x1,-0x4(%rbp)
  62.86 | 18:  cmpl   $0x5f5e100,-0x4(%rbp)
  18.88 |    ↑ jle    d
        |           }
        |  }
        |    nop
        |    pop    %rbp
        |  ← retq
```

PHPCON 官方渠道：

官网：http://www.phpconchina.com

公众号：PHPCon

纪念品购买渠道： https://k.weidian.com/H3=4lVho

微信交流群：

添加个人微信号「PHPConChina」自动通过后，输入加群密码：11643 稍等自动拉群

Thanks