

PHP 异步并行编程

Rango-韩天峰

一个PHP程序员到C程序员的历程



觉醒

探索

超越

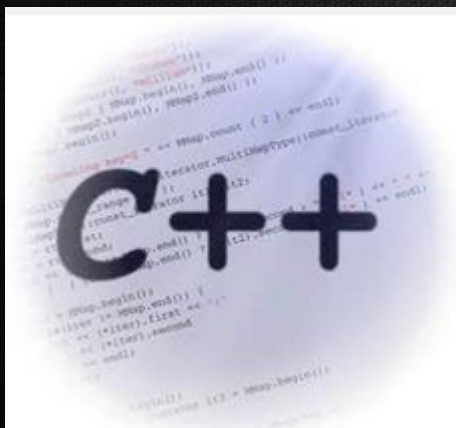
2010 年的我

- 写过Web框架，ORM，AutoLoad，编码规范，掌握面向对象，设计模式
- Javascript/HTML/CSS/Jquery/Ajax非常熟练
- Nginx+php-fpm/Apache+Mod_PHP各种配置部署UrlRewrite毫无压力
- 开发过大大小小10几个项目，开发上线的Web系统30多个
- PHP+MySQL+Memcache+KeyValueDB+队列异步化
- 了解PHP内核原理，看过鸟哥博客里的每一篇文章，读过walu.cc文章

PHPer 就这样了吗?

1. PHP除了Web还可以做什么?

2. PHPer和C++/Java程序员的差距



C++程序员眼中的
PHPer

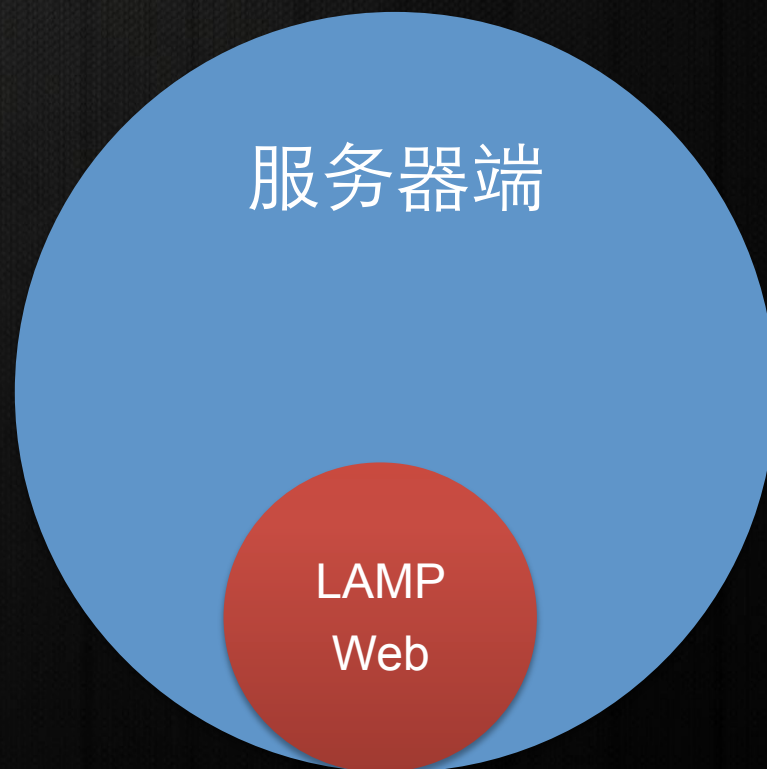




Java程序员眼中的PHPPer:

图样，图森破
Sometime naïve

- Web仅仅是整个服务器端的一部分，PHPer在服务器端方面还有很多未知领域可以去尝试
- PHPer在LAMP的保护下就像温室中的花朵，不经历外面世界的风雨，永远无法成长



开始探索之旅，学习底层网络通信

第一个Server，阻塞+fork子进程

```
$serv=stream_socket_server("tcp://0.0.0.0:8000", $errno, $errstr)  
or die("create server failed");
```

```
while(1) {  
    $conn = stream_socket_accept($serv);  
    if (pcntl_fork() == 0 ) {  
        $request = fread($conn);  
        //do some thing  
        //$response = "hello world";  
        fwrite($response);  
        fclose($conn);  
        exit(0);  
    }  
}
```

第二个Server，改良版

```
$serv=stream_socket_server("tcp://0.0.0.0:8000", $errno, $errstr)
or die("create server failed");
for($i=0; $i < 32; $i ++ ) {
    if (pcntl_fork() == 0 ) {
        while(1) {
            $conn = stream_socket_accept($serv);
            if ($conn == false) continue;
            $request = fread($conn);
            //do some thing
            //$response = "hello world";
            fwrite($response);
            fclose($conn);
        } exit(0);
    }
}
```

初次尝试异步(1)

- 阅读圣经：《Unix网络编程》(UNP)
- select: stream_select / socket_select
- Nginx,memcache: libevent & epoll , 异步的核心就是它
- 学习PHP的libevent扩展

初次尝试异步(2)

```
$serv = stream_socket_server("tcp://0.0.0.0:8000", $errno, $errstr);  
//for($i=0; $i < 32; $i ++)  
$base = event_base_new();  
$event = event_new();  
  
function read_cb($socket, $flag, $base) {  
    fread($socket);  
    fwrite("hello world\n");  
}  
  
function accept_cb($socket, $flag, $base) {  
    $conn = stream_socket_accept($socket, 0);  
    stream_set_blocking($conn, 0);  
    $event = event_new();  
    event_set($event, $conn, EV_READ | EV_PERSIST, 'read_cb', $base);  
    event_base_set($event, $base);  
    event_add($event);  
}  
event_set($event, $socket, EV_READ | EV_PERSIST, 'accept_cb', $base);  
event_base_set($event, $base);  
event_add($event);  
event_base_loop($base);
```

2011年 加入腾讯开始用PHP写Server

- 朋友网当时是国内最大的SNS交友网站，也是国内规模最大PHP网站
- 超过4亿注册用户，月活跃用户超过2亿
- 全球唯一完全PHP开发的Server集群（超过1200台机器）
- 朋友网扩展epoll,pcntl,posix,sockets,mmap,shm,sysvmsg
- 3个重量级组件：PWS，SAPS，PSF
- 朋友网在10年就开始转向4层架构，服务化治理（SOA）之路，Web层仅负责前端展现层逻辑。所有的业务逻辑都放到了服务层来做。

朋友网 系统架构

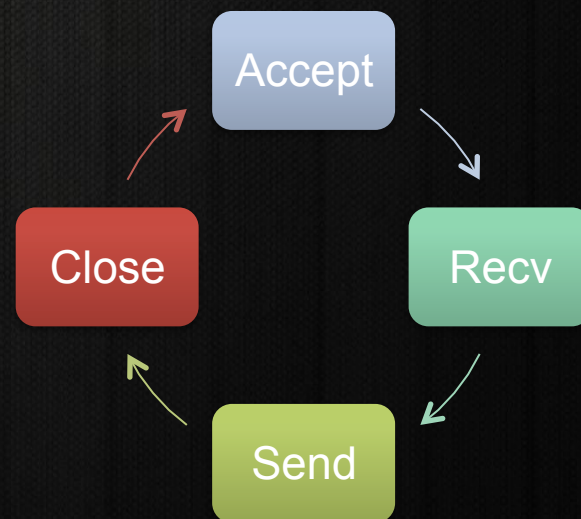


为什么不用FPM，而是自己写Server

- PHP的所有对象请求时创建，请求结束时全部销毁。对于普通PHP程序来说避免了内存泄漏。对于大型网站来说，这是严重的资源浪费。
- PHP-Server中每次请求仅销毁与请求相关的对象。与请求无关的全局对象都不需要销毁。直接在下一次请求中复用。
- APC/OpCache只能节省编译OpCode的时间，但OpCode还是要执行的。PHP-Server可以保留OpCode执行后的内存，节省资源。
- PHPer可以操控的范围更大了，局部缓存，Once操作，写文件合并，统计数据合并，长连接，对象持久化，数据库连接池等都可以做。

PHP应用服务器 PWS

- PHP开发的应用服务器，包括Http
协议解析/Request/Response/Cookie/Session
- 程序模型就是第二个Server中讲的
- 天然的Leader-Follower模式
- PWS集群：600多台机器
- 单机QPS峰值：2000
- 机器CPU：30% - 60%
- Web代码中使用并行SOAClient，并发地请求后端的SOA服务器
- Web代码中投递异步任务到SAPS服务



PHP分布式队列服务 SAPS

- PHP写的分布式队列服务，300台机器是运行这个服务的。
包括头像处理，图片压缩，邮件发送，短信服务，关系链分析等等
- 基于epoll + mmap + sysvmsg扩展，全部用PHP开发，由3个部分组成
- QueueClient：接受从Web发来的任务，连接到对应服务的ServerQueue，并传输任务数据到QueueServer
- QueueServer：将任务存入磁盘，与Worker之间通过MsgQueue通信。
QueueServer根据MsgQueue队列的负载情况，决定是否交给Worker执行
- Worker：多进程抢占式取队列，取到任务后转入执行状态。
- 300台机器的集群，承载了200多个服务，最大一项的服务日均超过10亿笔

逻辑层服务器框架 PSF

- PSF：PHP写的逻辑层Server框架，模型与Nginx相同。
- 300台机器的服务层Server集群
- 请求量最大的系统消息服务，1秒钟要处理2万次请求
- 性能：-c 5000 -n 100000，在8核8G机器上的测试
- TCP短连接4万QPS，TCP长连接8万QPS，UDP6万QPS
- 我在朋友网就是负责PSF框架的底层研发工作

C扩展

Swoole

PHP框架

朋友网PSF

PHPDaemon



WorkerMan

React

2012年8月 Swoole项目正式开始

- 基于过去3年的探索经验，结合对Node.js，Golang，C++ muduo的分析学习
决定开发一个全新的，超越一切的，超高性能Server框架
- 使用纯C进行开发，整个Server都是用扩展实现，仅业务逻辑部分使用PHP
C语言可以保证整个Server的性能，又能提供过去不能实现的功能
- 不依赖任何第三方库，包括libevent，libev等等
- 提供一个多线程EventLoop，全异步非阻塞，可以利用到多核
connection fd是连续的，共享的，连接之间可以互发数据
- 提供一个专门的Acceptor线程，用于处理TCP新连接，解决惊群问题
- 提供一个Manager+多进程Worker架构，构建一个高可靠性的PHP运行环境

Swoole项目的更多想法

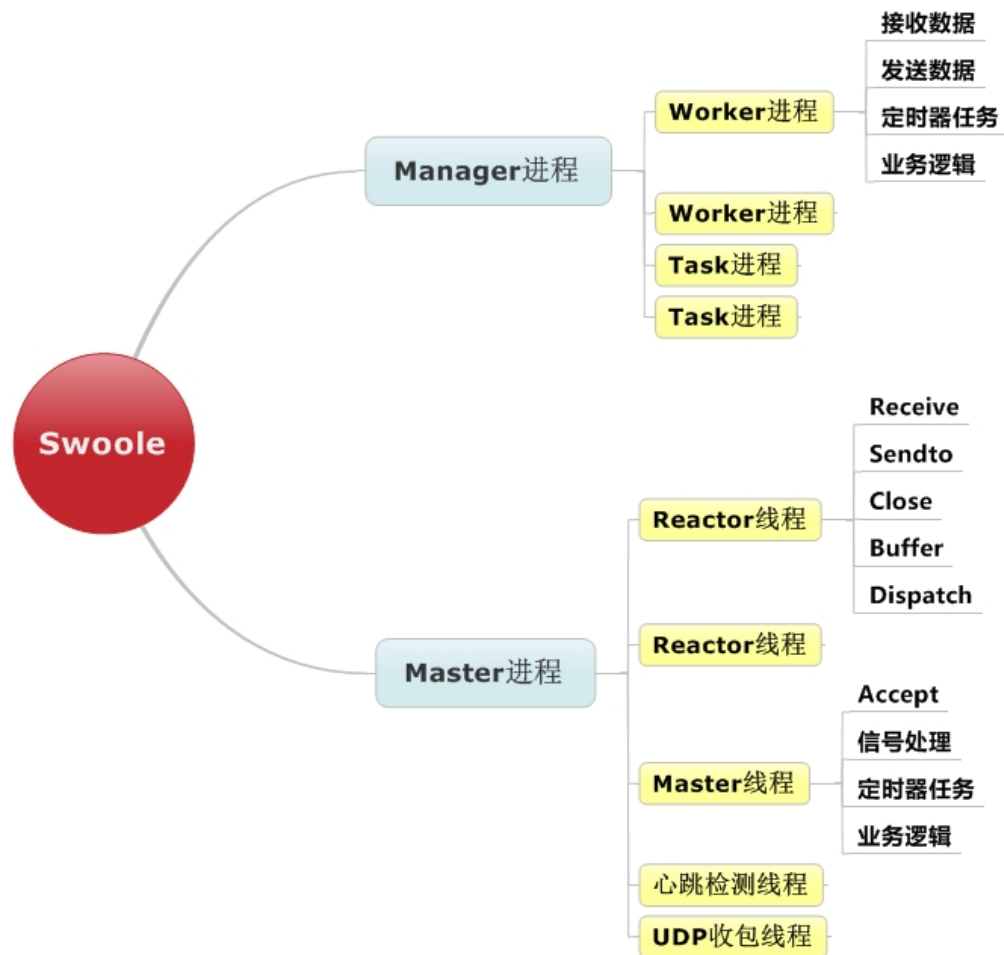
- 既要支持全异步非阻塞，又要支持同步阻塞。

还可以半异步（EventWorker）半同步（TaskWorker）

- 同时支持TCP/UDP/UUDP(UnixSocket)/IPv4/IPv6
- 支持Http与WebSocket协议
- 支持SSL/TLS隧道加密
- 支持同步/异步TCP/UDP/UUDP/IPv4/IPv6客户端
- 支持Lua和C/C++，可以编译为libswoole.so作为C库
- 异步MySQL，异步文件读写，异步DNS查询，异步Redis/Memcache
- 异步HttpClient，异步MongoDB-Client等更多

最好的Server模型

坚持 做



Swoole用到的系统调用

- 内存管理

FixedPool 固定分配内存池

RingBuffer 环形内存缓冲区

- 定时器: timerfd
- 信号处理: signalfd
- 数据通信: eventfd + mmap
- 发送大文件: sendfile
- 读写文件: Linux Native AIO
- 进程间通信: unix sock + msg_queue

Server

```
$serv = new swoole_server("127.0.0.1", 9501);  
$serv->on('connect', function ($serv, $fd){  
    echo "Client: Connected.\n";  
});  
$serv->on('receive', function ($serv, $fd, $from_id, $data){  
    $serv->send($fd, 'Swoole: '.$data);  
    $serv->close($fd);  
});  
$serv->on('close', function ($serv, $fd){  
    echo "Client: Closed.\n";  
});  
$serv->start();
```


Client

```
$client = new swoole_client(SWOOLE_TCP | SWOOLE_ASYNC);
$client->on("connect", function($cli) {
    $cli->send("hello world\n");
});
$client->on("receive", function($cli, $data){
    echo "Received: $data\n";
});
$client->on("error", function($cli){
    echo "connect failed\n";
});
$client->on("close", function($cli){
    echo "closed\n";
});
$client->connect('127.0.0.1', 9501);
```

真正的异步MySQL

```
$db = new mysqli;  
  
$db->connect('127.0.0.1', 'root', 'root', 'test');  
  
$db->query("show tables", MYSQLI_ASYNC);  
  
swoole_event_add(swoole_get_mysqli_sock($db), function($db_sock) {  
    $res = $db->reap_async_query();  
  
    var_dump($res->fetch_all(MYSQLI_ASSOC));  
});
```

更多接口

- 自动协议解析: EOF_Check/Length_check
- TCP心跳检测: `$serv->heartbeat();`
- 向客户端发送文件: `$serv->sendfile($sock, "/tmp/big.jpg");`
- 异步任务: `$serv->task("some data");`
- 重启所有Worker: `$serv->reload();`
- 结束服务器: `$serv->shutdown();`
- 得到所有TCP连接: `$serv->connection_list();`
- 获取TCP信息: `$serv->connection_info($sock);`
- 毫秒定时器: `$serv->addtimer(100); //100ms`

swoole模块介绍

模块名称	介绍
swoole_server	强大的TCP/UDP Server框架
swoole_client	TCP/UDP客户端，支持同步并发调用，也支持异步事件驱动。
swoole_event	EventLoop底层API，让用户可以将PHP socket, stream, 管道等加入到事件循环中。
swoole_async	异步IO接口，提供了 异步文件系统IO，异步DNS 查询，异步MySQL等API。
swoole_process	进程管理模块，可以方便的创建子进程，进程间通信，进程管理。
swoole_buffer	内存缓存管理工具，像C一样进行指针操作，又无需关心内存的申请和释放，无需担心内存越界。
swoole_table	共享内存/行锁，彻底解决线程/进程间数据共享，加锁同步等难题。

A full-page background image featuring a vast, deep blue ocean with gentle waves in the foreground. The horizon line is straight and divides the image roughly in half. Above the horizon, the sky is a brilliant, clear blue, populated by several large, fluffy white cumulus clouds. Sunlight rays are visible, radiating from behind the clouds towards the top corners of the frame. The overall mood is bright, clear, and serene.

Q & A