

# [精选] PHP程序员 到最后为什么要 学习swoole

PHP自学中心 2022-10-08 08:30 发表于广东

收录于合集

#php的自我修养

166个

商务合作加微信：2230304070

学习与交流：PHP技术交流微信群

首先，我们要知道什么是swoole

swoole是PHP的异步、并行、高性能网络通信引擎，使用纯C语言编写，提供了PHP语言的异步多线程服务器，异步TCP/UDP网络客户端，异步MySQL，异步Redis，数据库连接池，AsyncTask，消息队列，毫秒定时器，异步文件读写，异步DNS查询。Swoole内置了Http/WebSocket服务器端/客户端、Http2.0服务器端。

## 为什么要使用 Swoole

常驻内存，避免重复加载带来的性能损耗，提升海量性能

协程异步，提高对 I/O 密集型场景并发处理能力（如：微信开发、支付、登录等）

方便地开发 Http、WebSocket、TCP、UDP 等应用，可以与硬件通信

PHP 高性能微服务架构成为现实



phpstorm 全家桶激活码，支持所有版本，手慢无

链接：<http://web.52shizhan.cn/activity/s2abxc>

提取码：HJM0083L

## 常驻内存

目前传统 PHP框架，在处理每个请求之前，都要做一遍加载框架文件、配置的操作。这可能已经成为性能问题的一大原因，而使用 Swoole 则没有这个问题，一次加载多次使用。

## 协程

如下图所示，这是同一个线程处理并发请求的场景，比如你某个接口中需要调用其它 api 接口或读写大文件，传统同步阻塞和协程异步的优势就体现了出来。



说到协程，就得先简单说说进程和线程，众所周知进程是很占用资源的，为了处理请求大量创建进程肯定是得不偿失的。而多线程应用就比较多了，在 CPU 层面有几个核心就会执行几个任务，线程一旦创建的多了，就会有线程调度的损耗。

协程是在单线程基础上实现的，它可以最大限度利用 CPU 资源，而不会在等待 I/O 时白白浪费。当然，协程数越多占用的内存也就越多，不过这个是可以接受的，相比进程和线程，占用的资源是相对较少的。

使用协程时，遇到读写文件、请求接口等场景，会自动挂起协程，把 CPU 让给其它协程执行任务，这样可以提升单线程的 CPU 资源利用率，减少浪费，从而提高性能。

### 协程代码示例：

```
<?php
use Swoole\Coroutine as co;

// 协程
$time = microtime(true);
// 创建10个协程
for($i = 0; $i < 10; ++$i)
{
    // 创建协程
    go(function() use($i){
        co::sleep(1.0); // 模拟请求接口、读写文件等I/O
        echo $i, PHP_EOL;
    });
}
swoole_event_wait();
echo 'co time:', microtime(true) - $time, ' s', PHP_EOL;

// 同步
$time = microtime(true);
// 创建10个协程
for($i = 0; $i < 10; ++$i)
{
    sleep(1); // 模拟请求接口、读写文件等I/O
    echo $i, PHP_EOL;
}
```

```
}  
echo 'sync time:', microtime(true) - $time, ' s', PHP_EOL;
```

运行结果：

```
0  
9  
8  
7  
6  
5  
4  
3  
2  
1  
co time:1.0087130069733 s  
0  
1  
2  
3  
4  
5  
6  
7  
8  
9  
sync time:10.010055065155 s
```

从上面结果可以看出，协程方式执行并不是顺序的，性能更高，在sleep时会把当前线程的任务执行权交给其他协程。

## 创建 Http 服务

其实也没想象中的难，看代码：

```
$http = new swoole_http_server("127.0.0.1", 9501);  
$http->on('request', function ($request, $response) {  
    $response->end("<h1>Hello Swoole. #".rand(1000, 9999)."</h1>");  
});  
$http->start();
```

学习Swoole需要掌握哪些基础知识

大概就下面这些

- 了解Linux操作系统进程和线程的概念
- 了解Linux进程/线程切换调度的基本知识
- 了解进程间通信的基本知识，如管道、UnixSocket、消息队列、共享内存 SOCKET
- 了解SOCKET的基本操作如accept/connect、send/recv、close、listen、bind
- 了解SOCKET的接收缓存区、发送缓存区、阻塞/非阻塞、超时等概念 IO复用
- 了解select/poll/epoll
- 了解基于select/epoll实现的事件循环，Reactor模型

- 了解可读事件、可写事件 TCP/IP网络协议
- 了解TCP/IP协议
- 了解TCP、UDP传输协议 调试工具
- 使用 gdb 调试Linux程序
- 使用 strace 跟踪进程的系统调用
- 使用 tcpdump 跟踪网络通信过程
- 其他Linux系统工具，如ps、lsof、top、vmstat、netstat、sar、ss等



**原文链接:** <https://www.jb51.net/article/142518.htm>

以上就是本篇分钟的全部内容，**希望各位程序员们努力提升个人技术。**最后，小编温馨提示：每天阅读5分钟，每天学习一点点，每天进步一点点。



**PHP自学中心**

学习PHP的好助手，了解php最前沿技术，或是深入理解技术问题，学习php技巧与开发...

378篇原创内容

---

公众号

**点个赞** ○ **再走吧**

收录于合集 #php的自我修养 166

下一篇 · [精选] PHP8新特性：注解，你也该掌握了

喜欢此内容的人还喜欢

## 里程碑！用自己的编程语言实现了一个网站

crossoverJie



## 面向对象五大设计原则

精益码农



## 12 年的祖传“屎山”代码，年收入竟超 1.4 亿元？程序员劝“接盘侠”：赶紧退退退！

非著名程序员

