

Swoole v6 能否让 PHP 再次伟大?

大家好，我是码农先森。

现状

--

传统的 PHP-FPM 也是多进程模型的运行方式，但每个进程只能处理完当前请求，才能接收下一个请求。而且对于 PHP 脚本来说，只是接收请求和响应请求，并不参与网络通信。对数据库资源的操作，也是一次请求一次有效，用完即销毁不能复用，在系统高负载的情况下对数据库等资源的消耗会很大，能承受的并发量有限。

Swoole 的出现给 PHP 带来了一种新的运行方式，完全接管了 PHP-FPM 的功能，并且弥补了 PHP 在异步网络通信领域的空白。Swoole 提供了 PHP 的全生命周期管理，此外 Swoole 的常驻进程模式，也能够高效的利用资源，比如可以建立数据库连接池、共享内存变量等。还有 Swoole 中能够支撑高并发的利器「协程」，更加使 PHP 的性能上了一个新的台阶，甚至在某些特定场景下都可以与 Go 语言的性能相媲美。

虽说 Swoole 给 PHP 带来了很大的性能提升，但也还是一个基于多进程模型的异步通信扩展，多进程的模式也存在着许多的问题，比如跨进程间的通信、进程间的资源共享等问题。简而言之，多进程会带来一定的系统资源消耗及产生新的问题。

因此 Swoole 官方为了解决多进程的问题，引进了多线程的支持，这意味着 v6 版本之后，Swoole 将会变成单进程多线程的运行模式。

v6 新特性

根据 Swoole 作者韩天峰发布的预告，在 v6 版本中增加多线程的支持。其中多线程的实现是基于 PHP 的 ZTS 机制和 TSRMLS API，在 PHP 层面隔离所有全局变量，实现线程安全。Swoole v6 的多线程将是真正的多线程实现，在单进程的模式下所有的 PHP 程序代码均是在多核并行执行，能够高效的利用好 CPU 资源。

v6 版本还提供了线程安全的 Map 和 ArrayList 数据结构，可以实现跨线程的数据共享读写。在 Server 端的 Event Worker、Task Worker、User Process 等将全部替换为线程的运行方式，在同一个进程空间内执行，彻底摒弃了多进程的模式。

当然新的特性势必会带来新的开销，对于 Map 等共享的数据结构在多线程的模式下需要加锁，来避免数据竞争，可能会损耗一些性能。

以下是列举的一些线程相关的 API 方法：

- * use Swoole\Thread 线程对象。
- * use Swoole\Thread\Map 线程安全下的 Map 数据结构。
- * use Swoole\Thread\ArrayList 线程安全下的 ArrayList 数据结构。
- * Swoole\Thread::getId() 获取当前线程的 ID。
- * Swoole\Thread:: getArguments() 获取父线程传递给子线程的参数列表。
- * Swoole\Thread::join() 等待子线程退出，请注意 \$thread 对象销毁时会自动执行 join()，这可能会导致进程阻塞。
- * Swoole\Thread::joinable() 检测子线程是否已退出。
- * Swoole\Thread::detach() 使子线程独立运行，不再需要 Thread::join()。
- * Swoole\Thread::HARDWARE_CONCURRENCY 硬件层支持的并行线程数量。
- * Swoole\Thread::\$id 获取子线程的 ID。
- * Swoole\Thread::exec() 开启一个新的线程。

最后

自 Swoole 从 2012 年发布第一个版本开始，就扛起了 PHP 领域异步通信的大旗，但这多年以来 Swoole 的发展也是实属不易。还记得刚开始时的异步回调模式的套娃式编程方式，开发起来异常艰难，到后来的同步式编程，直接降低

了 PHP 程序员的学习门槛，让 PHP 在实时通信、物联网通信、游戏开发等领域也能大展拳脚，同时在 PHP 的发展史上也产生了重大的影响。

随着 Go 语言在编程界的持续火热，Swoole 常常被 PHP 程序员拿来和 Go 语言一决高下，总是被诟病 Swoole 无法有效利用多核 CPU、进程间的通信困难等问题。话又说回来，Swoole 作为一个 PHP 的扩展程序和天生具有高性能的 Go 语言自然是不可比拟的，但 Swoole 也是在逐渐的向 Go 语言靠近，比如 Swoole 中也使用了「go、channel」关键词来实现协程及通信通道，虽说底层的实现机制还是大不相同的。

当然 Swoole 也在不断地努力持续优化，就像将要推出的 v6 版本增加多线程的支持，来改变目前多进程的局面。至于这个版本对 PHP 发展来说有没有很大的影响，我认为影响有限。但对 Swoole 的发展还是有很大的影响，毕竟以后再也不用受多进程的困扰了，这也是一大进步。

在 Web 领域作为世界上最好的语言，尽管 PHP 近年来的发展不尽如人意，但作为一名 PHPer 也有必要和有义务一起来维护和推动 PHP 生态的发展。

> 欢迎、分享、点赞、收藏、在看，我是公众号「码农先森」作者。

原文链接: <https://juejin.cn/post/7384696986845085731>