

协程是一种用户态的轻量级线程，协程的调度完全由用户控制。协程拥有自己的寄存器上下文和栈。协程调度切换时，将寄存器上下文和栈保存到其他地方，在切回来的时候，恢复先前保存的寄存器上下文和栈，直接操作栈则基本没有内核切换的开销，可以不加锁的访问全局变量，所以上下文的切换非常快。

线程的性能比协程更好，因为线程能利用多核达到真正的并行计算，如果任务设计得好，线程能几乎成倍的提高你的计算能力，说线程性能不好主要是需要设计导致存在大量的锁、切换、等待。而协程因为是非抢占式的，所以需要用户自己释放使用权来切换到其他线程，因此同一时间其实只有一个协程拥有运行权

协程是一个不由os内核抢占调度，而由程序管理在用户态自管理的协作式“线程”，用协程：

1. 省去了cpu线程切换的开销。
2. 降低了内存消耗。
3. 提高了cpu缓存命中率。
4. 整体上提高了性能。

协程与线程的比较：

1. 一个进程可以拥有多个协程，一个进程也可以单独拥有多个协程
2. 线程进程都是同步机制，而协程则是异步。
3. 协程能保留上一次调用时的状态，每次过程重入时，就相当于进入上一次调用的状态。

总结一下：一般是io密集型用多线程或者多进程，cpu密集型一般用多进程，强调非阻塞异步并发的一般都是使用协程。