# 性能优化与GIL

## python性能分析与优化，GIL

* cython解释器的内存管理并不是线程安全的
* 保护多线程情况下对python对象的访问
* cython使用简单的锁机制避免多个线程同时执行字节码

## GIL的影响

限制了程序的多核执行

* 同一个时间只能有一个线程执行字节码
* CPU密集程序难以利用多核优势
* IO期间会释放GIL，对IO密集程序影响不大

如何规避GIL的影响

* 区分CPU和IO密集程序
* CPU密集可以使用多进程+进程池
* IO密集使用多线程/协程
* cpython扩展

## 为什么有了GIL还要关注线程安全

python中什么操作才是原子的？一步到位执行完

* 一个操作如果是一个字节码指令可以完成就是原子的
* 非原子操作不是线程安全的
* 原子的是可以保证线程安全的
* 使用dis操作来分析字节码

## 服务端性能优化措施

web应用一般语言不会成为瓶颈

* 数据结构与算法优化
* 数据库层：索引优化，慢查询消除，批量操作减少IO，NoSQL
* 网络IO：批量操作，pipeline操作，减少IO
* 缓存：使用内存数据库 redis/memcached
* 异步：asyncio，celery
* 并发：gevent/多线程