**Multithreading Costs**

由单线程走向多线程应用不仅仅提供益处。也有些花销。不要仅仅是多线程化一个应用仅仅因为你可以。你应当有一个好的想法，这么做的收益比花销要大。当处于疑惑中，试着测量应用的性能或者响应性，而非推测。

**More complex design**

尽管多线程应用的某些部分比单线程应用简单，但另外一些部分要更复杂。访问共享数据的多个线程执行的代码需要格外注意。线程交互并非总是简单的。不正确的线程同步导致的错误很难检测、复现和修复。

**Context Switching Overhead**

当一个CPU从执行一个线程切换到执行另一个，CPU需要保存当前线程的本地数据、程序指针等，并加载下一个将要执行的线程的本地数据、程序指针等。这个切换被叫做“上下文(context)切换”。CPU切换从在一个线程的上下文中执行到在另一个线程的上下文中执行。

上下文切换花销不便宜。非必要你不会想要在线程之间切换

你可以在Wikipedia查到更多关于上下文切换:

[**http://en.wikipedia.org/wiki/Context\_switch**](http://en.wikipedia.org/wiki/Context_switch)

**Increased Resource Consumption**

A thread needs some resources from the computer in order to run. Besides CPU time a thread needs some memory to keep its local stack. It may also take up some resources inside the operating system needed to manage the thread. Try creating a program that creates 100 threads that does nothing but wait, and see how much memory the application takes when running.一个线程为了运行需要一些计算机上的资源。包括CPU时间在内，线程还需要一些内存来保持本地栈。它也可能花费一些操作系统内部必需的资源来管理线程。试着创建一个程序，创建100个线程除了等待什么也不做，然后看看应用运行时占用多少内存。