

我们都知道web应用不同于winform、wpf等客户端应用，客户端应用为了保证UI渲染的一致性往往都是采用单线程模式，这个UI线程称为主线程，如果在主线程做耗时操作就会导致程序界面假死，所以客户端开发中使用多线程异步编程非常必要。

可web应用本身就是多线程模式，服务器会为每个请求分配工作线程。

既然async/await不能创建新线程，又不能使提高请求的响应速度，那.NET Web应用中为什么要使用async/await异步编程呢？

在 **web** 服务器上，**.NET Framework** 维护用于处理 **ASP.NET** 请求的线程池。当请求到达时，将调度池中的线程以处理该请求。如果以同步方式处理请求，则处理请求的线程将在处理请求时处于繁忙状态，并且该线程无法处理其他请求。

在启动时看到大量并发请求的 **web** 应用中，或具有突发负载（其中并发增长突然增加）时，使 **web** 服务调用异步会提高应用程序的响应能力。异步请求与同步请求所需的处理时间相同。如果请求发出需要两秒钟时间才能完成的 **web** 服务调用，则该请求将需要两秒钟，无论是同步执行还是异步执行。但是，在异步调用期间，线程在等待第一个请求完成时不会被阻止响应其他请求。因此，当有多个并发请求调用长时间运行的操作时，异步请求会阻止请求队列和线程池的增长。

那是不是所有的action都要用async/await呢？

不是。一般的磁盘IO或者网络请求等耗时操作才考虑使用异步，不要为了异步而异步，异步也是需要消耗性能的，使用不合理会适得其反。

结论

async/await异步编程不能提升响应速度，但是可以提升响应能力（吞吐量）。异步和同步各有优劣，要合理选择，不要为了异步而异步。