第四章

分析

好的，所以你已经学会了关键的性能优化技术，并可以将它们应用到你的代码中。但是，当我们讨论过的技术都不起作用时你会怎么做？

你要分析。

分析是回答“什么减慢了代码的速度？”唯一可靠方法。分析可能困难又耗时，但他确实没有任何捷径。如果你不能通过查看代码或采取有根据的猜测来优化，你就必须进行分析。

一旦你确切地知道什么在减慢你的速度，修复它就变得不值一提了。所以现在我将教你分析的秘密，这能使你更容易地发现是什么变慢了。

让我们通过将其分解为两个基本部分来开始我们对分析的探索。首先，测量内存或CPU使用率并将其归因于代码中的特定位置，通常是函数调用。其次，解释结果以识别代码的缓慢部分。这是两种截然不同的活动，你需要以不同的方式思考它们。

测量是一项纯粹的工程任务，而且很简单。你可以手动完成或使用分析器工具。我将展示如何使用这些工具。

解释测量更复杂，但秘诀是将其视为工艺，而不是工程任务。我见过许多优秀的软件开发人员正在放弃分析，正是因为他们试图像工程师一样地去分析。你的左脑会认为分析是麻烦又无趣的。但是当让你的右脑参与其中后，相反你会发现它有趣且令人兴奋。

因为分析是一种工艺，至少我会把它当作一种工艺来教——换句话说，通过例子。我将展示我如何使用具体示例来分析我的代码，然后你可以通过实例来提炼出适合你的技术。一旦你学会示例中的模式，这将变得很容易。所以，动用你的右脑，让我们开始吧。