 **练习题 1.2** 公司的市场部向你的客户承诺，下一个版本的软件性能将改进 $2\times$ 。这项任务被分配给你。你已经确认只有 80% 的系统能够被改进，那么，这部分需要被改进多少（即 k 取何值）才能达到整体性能目标？

Amdahl 定律一个有趣的特殊情况是考虑 k 趋向于 ∞ 时的效果。这就意味着，我们可以取系统的某一部分将其加速到一个点，在这个点上，这部分花费的时间可以忽略不计。于是我们得到

$$S_{\infty} = \frac{1}{(1-\alpha)} \quad (1.2)$$

举个例子，如果 60% 的系统能够加速到不花时间的程度，我们获得的净加速比将仍只有 $1/0.4 = 2.5\times$ 。

Amdahl 定律描述了改善任何过程的一般原则。除了可以用在加速计算机系统方面之外，它还可以用在公司试图降低刀片制造成本，或学生想要提高自己的绩点平均值等方面。也许它在计算机世界里是最有意义的，在这里我们常常把性能提升 2 倍或更高的比例因子。这么高的比例因子只有通过优化系统的大部分组件才能获得。

1.9.2 并发和并行

数字计算机的整个历史中，有两个需求是驱动进步的持续动力：一个是我们想要计算机做得更多，另一个是我们想要计算机运行得更快。当处理器能够同时做更多的事情时，这两个因素都会改进。我们用的术语并发 (concurrency) 是一个通用的概念，指一个同时具有多个活动的系统；而术语并行 (parallelism) 指的是用并发来使一个系统运行得更快。并行可以在计算机系统的多个抽象层次上运用。在此，我们按照系统层次结构中由高到低的顺序重点强调三个层次。

1. 线程级并发

构建在进程这个抽象之上，我们能够设计出同时有多个程序执行的系统，这就导致了并发。使用线程，我们甚至能够在一个进程中执行多个控制流。自 20 世纪 60 年代初期出现时间共享以来，计算机系统中就开始有了对并发执行的支持。传统意义上，这种并发执行只是模拟出来的，是通过使一台计算机在它正在执行的进程间快速切换来实现的，就好像一个杂耍艺人保持多个球在空中飞舞一样。这种并发形式允许多个用户同时与系统交互，例如，当许多人想要从一个 Web 服务器获取页面时。它还允许一个用户同时从事多个任务，例如，在一个窗口中开启 Web 浏览器，在另一窗口中运行字处理器，同时又播放音乐。在以前，即使处理器必须在多个任务间切换，大多数实际的计算也都是由一个处理器来完成的。这种配置称为单处理器系统。

当构建一个由单操作系统内核控制的多处理器组成的系统时，我们就得到了一个多处理器系统。其实从 20 世纪 80 年代开始，在大规模的计算中就有了这种系统，但是直到最近，随着多核处理器和超线程 (hyperthreading) 的出现，这种系统才变得常见。图 1-16 给出了这些不同处理器类型的分类。

多核处理器是将多个 CPU (称为“核”) 集成到一个集成电路芯片上。图 1-17 描述的是一个

所有的处理器

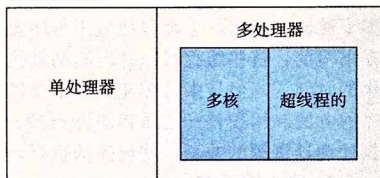


图 1-16 不同的处理器配置分类。随着多核处理器和超线程的出现，多处理器变得普遍了