

1.9 重要主题

在此，小结一下我们旋风式的系统漫游。这次讨论得出一个很重要的观点，那就是系统不仅仅只是硬件。系统是硬件和系统软件互相交织的集合体，它们必须共同协作以达到运行应用程序的最终目的。本书的余下部分会讲述硬件和软件的详细内容，通过了解这些详细内容，你可以写出更快速、更可靠和更安全的程序。

作为本章的结束，我们在此强调几个贯穿计算机系统所有方面的重要概念。我们会在本书中的多处讨论这些概念的重要性。

1.9.1 Amdahl 定律

Gene Amdahl，计算领域的早期先锋之一，对提升系统某一部分性能所带来的效果做出了简单却有见地的观察。这个观察被称为 Amdahl 定律(Amdahl's law)。该定律的主要思想是，当我们对系统的某个部分加速时，其对系统整体性能的影响取决于该部分的重要性和加速程度。若系统执行某应用程序需要时间为 T_{old} 。假设系统某部分所需执行时间与该时间的比例为 α ，而该部分性能提升比例为 k 。即该部分初始所需时间为 αT_{old} ，现在所需时间为 $(\alpha T_{old})/k$ 。因此，总的执行时间应为

$$T_{new} = (1 - \alpha)T_{old} + (\alpha T_{old})/k = T_{old}[(1 - \alpha) + \alpha/k]$$

由此，可以计算加速比 $S = T_{old}/T_{new}$ 为


$$S = \frac{1}{(1 - \alpha) + \alpha/k} \quad (1.1)$$

举个例子，考虑这样一种情况，系统的某个部分初始耗时比例为 60% ($\alpha = 0.6$)，其加速比因子为 3 ($k = 3$)。则我们可以获得的加速比为 $1/[0.4 + 0.6/3] = 1.67$ 倍。虽然我们对系统的一个主要部分做出了重大改进，但是获得的系统加速比却明显小于这部分的加速比。这就是 Amdahl 定律的主要观点——要想显著加速整个系统，必须提升全系统中相当大的部分的速度。

旁注 表示相对性能

性能提升最好的表示方法就是用比例的形式 T_{old}/T_{new} ，其中， T_{old} 为原始系统所需时间， T_{new} 为修改后的系统所需时间。如果有所改进，则比值应大于 1。我们用后缀“ \times ”来表示比例，因此，“ $2.2\times$ ”读作“2.2 倍”。

表示相对变化更传统的方法是用百分比，这种方法适用于变化小的情况，但其定义是模糊的。应该等于 $100 \cdot (T_{old} - T_{new})/T_{new}$ ，还是 $100 \cdot (T_{old} - T_{new})/T_{old}$ ，还是其他的值？此外，它对较大的变化也没有太大意义。与简单地说性能提升 $2.2\times$ 相比，“性能提升了 120%”更难理解。

 **练习题 1.1** 假设你是个卡车司机，要将土豆从爱达荷州的 Boise 运送到明尼苏达州的 Minneapolis，全程 2500 公里。在限速范围内，你估计平均速度为 100 公里/小时，整个行程需要 25 个小时。

- 你听到新闻说蒙大拿州刚刚取消了限速，这使得行程中有 1500 公里卡车的速度可以为 150 公里/小时。那么这对整个行程的加速比是多少？
- 你可以在 www.fasttrucks.com 网站上为自己的卡车买个新的涡轮增压器。网站现货供应各种型号，不过速度越快，价格越高。如果想要让整个行程的加速比为 $1.67\times$ ，那么你必须以多快的速度通过蒙大拿州？