第 6 章

CHAPTER 6

存储器层次结构

到目前为止，在对系统的研究中，我们依赖于一个简单的计算机系统模型，CPU 执行指令，而存储器系统为 CPU 存放指令和数据。在简单模型中，存储器系统是一个线性的字节数组，而 CPU 能够在一个常数时间内访问每个存储器位置。虽然迄今为止这都是一个有效的模型，但是它没有反映现代系统实际工作的方式。

实际上，*存储器系统*(memory system)是一个具有不同容量、成本和访问时间的存储设备的层次结构。CPU 寄存器保存着最常用的数据。靠近 CPU 的小的、快速的*高速缓存存储器*(cache memory)作为一部分存储在相对慢速的主存储器(main memory)中数据和指令的缓冲区域。主存缓存存储在容量较大的、慢速磁盘上的数据，而这些磁盘常常又作为存储在通过网络连接的其他机器的磁盘或磁带上的数据的缓冲区域。

存储器层次结构是可行的，这是因为与下一个更低层次的存储设备相比来说，一个编写良好的程序倾向于更频繁地访问某一个层次上的存储设备。所以，下一层的存储设备可以更慢速一点，也因此可以更大，每个比特位更便宜。整体效果是一个大的存储器池，其成本与层次结构底层最便宜的存储设备相当，但是却以接近于层次结构顶部存储设备的高速率向程序提供数据。

作为一个程序员，你需要理解存储器层次结构，因为它对应用程序的性能有着巨大的影响。如果你的程序需要的数据是存储在 CPU 寄存器中的，那么在指令的执行期间，在 0 个周期内就能访问到它们。如果存储在高速缓存中，需要 4～75 个周期。如果存储在主存中，需要上百个周期。而如果存储在磁盘上，需要大约几千万个周期！

这里就是计算机系统中一个基本而持久的思想：如果你理解了系统是如何将数据在存储器层次结构中上上下下移动的，那么你就可以编写自己的应用程序，使得它们的数据项存储在层次结构中较高的地方，在那里 CPU 能更快地访问到它们。

这个思想围绕着计算机程序的一个称为*局部性*(locality)的基本属性。具有良好局部性的程序倾向于一次又一次地访问相同的数据项集合，或是倾向于访问邻近的数据项集合。具有良好局部性的程序比局部性差的程序更多地倾向于从存储器层次结构中较高层次处访问数据项，因此运行得更快。例如，在 Core i7 系统，不同的矩阵乘法核心程序执行相同数量的算术操作，但是有不同程度的局部性，它们的运行时间可以相差 40 倍！

在本章中，我们会看看基本的存储技术——SRAM 存储器、DRAM 存储器、ROM 存储器以及旋转的和固态的硬盘——并描述它们是如何被组织成层次结构的。特别地，我们将注意力集中在高速缓存存储器上，它是作为 CPU 和主存之间的缓存区域，因为它们对应用程序性能的影响最大。我们向你展示如何分析 C 程序的局部性，并且介绍改进你的程序中局部性的技术。你还会学到一种描绘某台机器上存储器层次结构的性能的有趣方法，称为“存储器山(memory mountain)”，它展示出读访问时间是局部性的一个函数。

6.1 存储技术

计算机技术的成功很大程度上源自于存储技术的巨大进步。早期的计算机只有几千字