

或多个小型到中型的基于 SRAM 的高速缓存存储器，可以在几个 CPU 时钟周期内访问它们。然后是一个大的基于 DRAM 的主存，可以在几十到几百个时钟周期内访问它们。接下来是慢速但是容量很大的本地磁盘。最后，有些系统甚至包括了一层附加的远程服务器上的磁盘，要通过网络来访问它们。例如，像安德鲁文件系统 (Andrew File System, AFS) 或者网络文件系统 (Network File System, NFS) 这样的分布式文件系统，允许程序访问存储在远程的网络服务器上的文件。类似地，万维网允许程序访问存储在世界上任何地方的 Web 服务器上的远程文件。

### 旁注 其他的存储器层次结构

我们向你展示了一个存储器层次结构的示例，但是其他的组合也是可能的，而且确实也很常见。例如，许多站点(包括谷歌的数据中心)将本地磁盘备份到存档的磁带上。其中有些站点，在需要时由人工装好磁带。而其他站点则是由磁带机器人自动地完成这项任务。无论在何种情况中，磁带都是存储器层次结构中的一层，在本地磁盘层下面，本书中提到的通用原则也同样适用于它。磁带每字节比磁盘更便宜，它允许站点将本地磁盘的多个快照存档。代价是磁带的访问时间要比磁盘的更长。来看另一个例子，固态硬盘在存储器层次结构中扮演着越来越重要的角色，连接起 DRAM 和旋转磁盘之间的鸿沟。

### 6.3.1 存储器层次结构中的缓存

一般而言，高速缓存(cache，读作“cash”)是一个小而快速的存储设备，它作为存储在更大、也更慢的设备中的数据对象的缓冲区域。使用高速缓存的过程称为缓存(caching，读作“cashing”)。

存储器层次结构的中心思想是，对于每个  $k$ ，位于  $k$  层的更快更小的存储设备作为位于  $k+1$  层的更大更慢的存储设备的缓存。换句话说，层次结构中的每一层都缓存来自较低一层的数据对象。例如，本地磁盘作为通过网络从远程磁盘取出的文件(例如 Web 页面)的缓存，主存作为本地磁盘上数据的缓存，依此类推，直到最小的缓存——CPU 寄存器组。

图 6-22 展示了存储器层次结构中缓存的一般性概念。第  $k+1$  层的存储器被划分成连续的数据对象组块(chunk)，称为块(block)。每个块都有一个唯一的地址或名字，使之区别于其他的块。块可以是固定大小的(通常是这样的)，也可以是可变大小的(例如存储在 Web 服务器上的远程 HTML 文件)。例如，图 6-22 中第  $k+1$  层存储器被划分成 16 个大小固定的块，编号为 0~15。

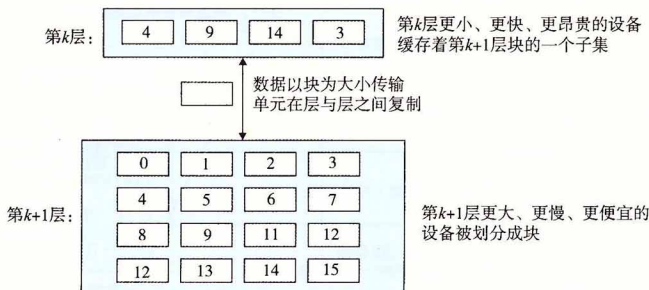


图 6-22 存储器层次结构中基本的缓存原理