

不同的速度变化着。特别地, DRAM 和磁盘访问时间远远大于 CPU 周期时间。系统通过将存储器组织成存储设备的层次结构来弥补这些差异, 在这个层次结构中, 较小、较快的设备在顶部, 较大、较慢的设备在底部。因为编写良好的程序有好的局部性, 大多数数据都可以从较高层得到服务, 结果就是存储系统能以较高层的速度运行, 但却有较低层的成本和容量。

程序员可以通过编写有良好空间和时间局部性的程序来显著地改进程序的运行时间。利用基于 SRAM 的高速缓存存储器特别重要。主要从高速缓存取数据的程序能比主要从内存取数据的程序运行得快得多。

## 参考文献说明

内存和磁盘技术变化得很快。根据我们的经验, 最好的技术信息来源是制造商维护的 Web 页面。像 Micron、Toshiba 和 Samsung 这样的公司, 提供了丰富的当前有关内存设备的技术信息。Seagate 和 Western Digital 的页面也提供了类似的有关磁盘的有用信息。

关于电路和逻辑设计的教科书提供了关于内存技术的详细信息[58, 89]。IEEE Spectrum 出版了一系列有关 DRAM 的综述文章[55]。计算机体系结构国际会议(ISCA)和高性能计算机体系结构(HPCA)是关于 DRAM 存储性能特性的公共论坛[28, 29, 18]。

Wilkes 写了第一篇关于高速缓存存储器的论文[117]。Smith 写了一篇经典的综述[104]。Przybylski 编写了一本关于高速缓存设计的权威著作[86]。Hennessy 和 Patterson 提供了对高速缓存设计问题的全面讨论[46]。Levinthal 写了一篇有关 Intel Core i7 的全面性能指南[70]。

Stricker 在[112]中介绍了存储器山的思想, 作为对存储器系统的全面描述, 并且在后来的工作描述中非正式地提出了术语“存储器山”。编译器研究者通过自动执行我们在 6.6 节中讨论过的那些手工代码转换来增加局部性[22, 32, 66, 72, 79, 87, 119]。Carter 和他的同事们提出了一个高速缓存可知晓的内存控制器(cache-aware memory controller)[17]。其他的研究者开发出了高速缓存不知晓的(cache oblivious)算法, 它被设计用来在不明确知道底层高速缓存存储器结构的情况下也能运行得很好[30, 38, 39, 9]。

关于构造和使用磁盘存储设备也有大量的论著。许多存储技术研究者找寻方法, 将单个的磁盘集成更大、更健壮和更安全的存储池[20, 40, 41, 83, 121]。其他研究者找寻利用高速缓存和局部性来改进磁盘访问性能的方法[12, 21]。像 Exokernel 这样的系统提供了更多的对磁盘和存储器资源的用户级控制[57]。像安德鲁文件系统[78]和 Coda[94]这样的系统, 将存储器层次结构扩展到了计算机网络和移动笔记本电脑。Schindler 和 Ganger 开发了一个有趣的工具, 它能自动描述 SCSI 磁盘驱动器的构造和性能[95]。研究者正在研究构造和使用基于闪存的 SSD 的技术[8, 81]。

## 家庭作业

- 6.22 假设要求你设计一个每条磁道位数固定的旋转磁盘。你知道每条磁道的位数是由最里层磁道的周长决定的, 可以假设它就是中间那个圆周的周长。因此, 如果你把磁盘中间的洞做得大一点, 每条磁道的位数就会增大, 但是总的磁道数会减少。如果用  $r$  来表示盘面的半径,  $x \cdot r$  表示圆洞的半径, 那么  $x$  取什么值能使这个磁盘的容量最大?
- 6.23 估计访问下面这个磁盘上扇区的平均时间(以 ms 为单位):

参数	值
旋转速率	15 000RPM
$T_{\text{avg seek}}$	4 ms
平均扇区数/磁道	800

- 6.24 假设一个 2MB 的文件, 由 512 个字节的逻辑块组成, 存储在具有下述特性的磁盘驱动器上: