第四部分：存储管理

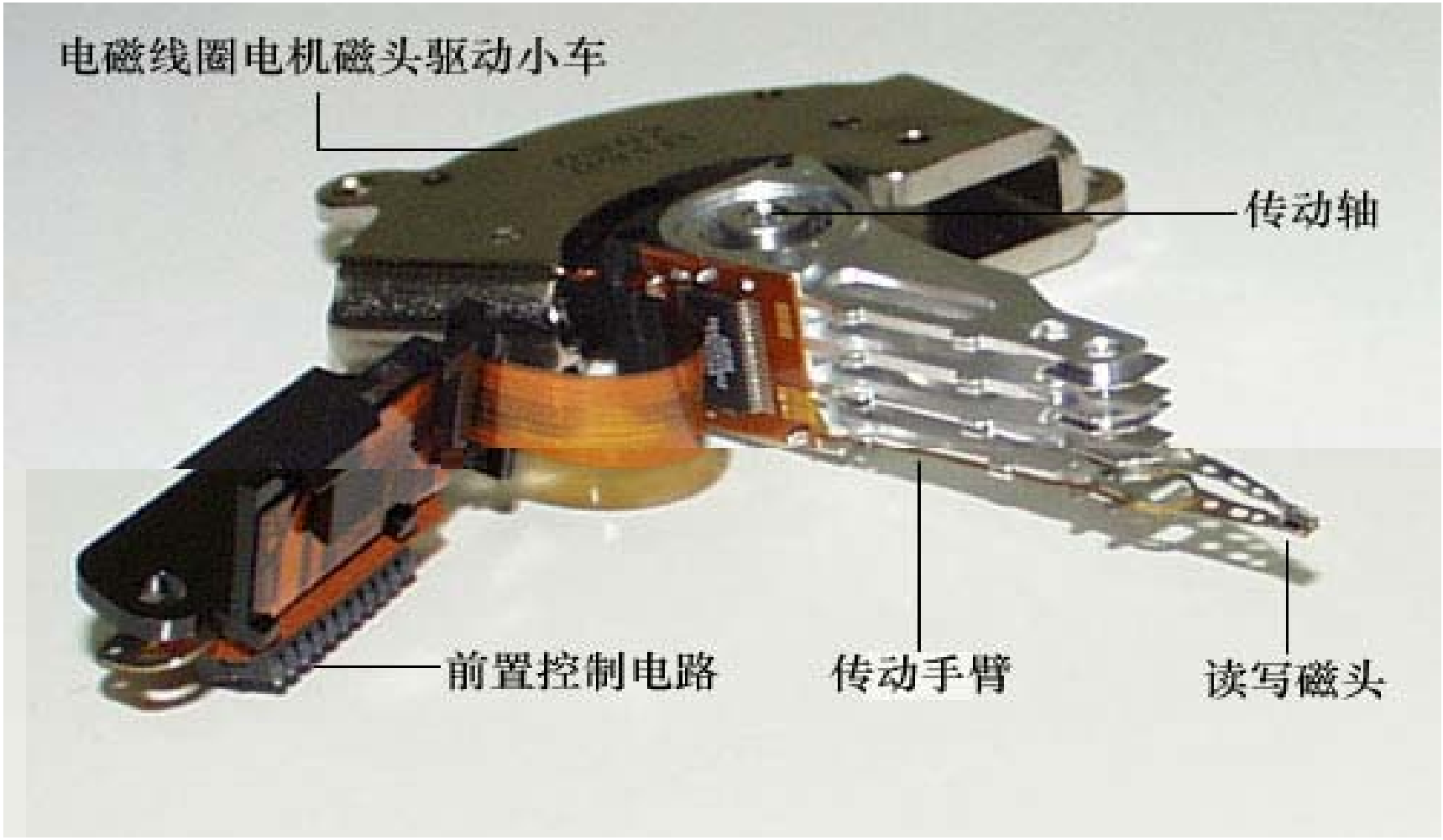
第十二章、大容量存储器结构

1. 概述
2. 磁盘结构
3. 磁盘调度
4. 磁盘管理
5. 交换空间管理
6. RAID结构
7. 第三级存储结构

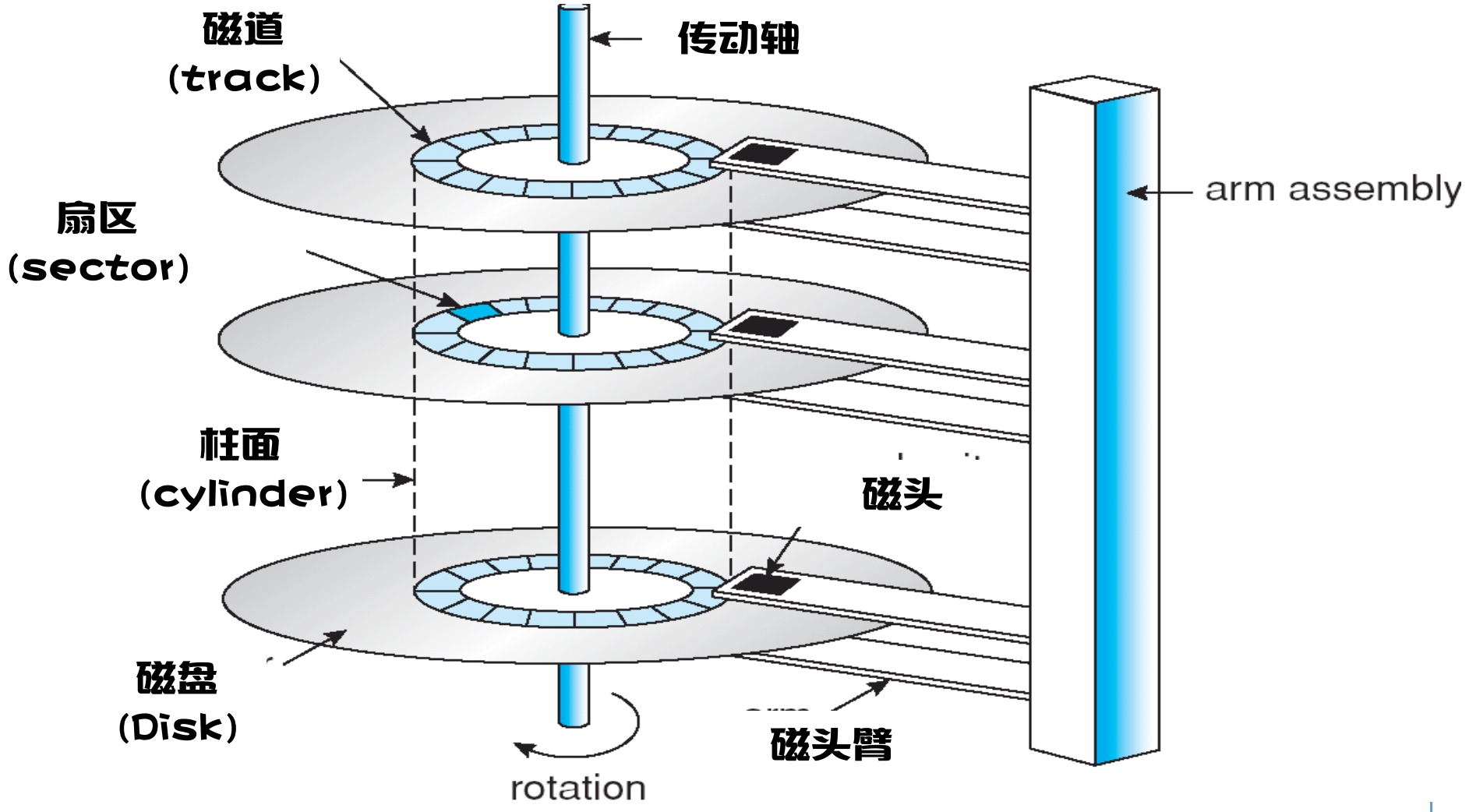
12.1 概述：磁盘

磁盘的结构

1. 外观
2. 立面图


电磁线圈电机磁头驱动小车
传动轴
前置控制电路
传动手臂
读写磁头

移动头磁盘机制


磁道
(track)
传动轴
扇区
(sector)
arm assembly
柱面
(cylinder)
磁头
磁盘
(Disk)
磁头臂
rotation

12.2 磁盘的结构

现代磁盘驱动器可以看做一个一维的逻辑块的数组。

- 编址方式：

  柱面，磁道、扇区
- 柱面：指的是各磁盘相同位置上磁道的集合

逻辑块由【柱面号，磁道号，扇区号】来定义

一维逻辑块数组按顺序映射到磁盘的扇区。

- 扇区0是最外面柱面的第一个磁道第一个扇区。
- 该映射是先按磁道内扇区顺序，再按柱面内磁道顺序，再按从外到内的柱面顺序来排序

磁盘的类型

1. **固定头磁盘**：**在每条磁道上都有一读/写磁头**，所有的磁头都被装在一刚性磁臂中。通过这些磁头可访问所有各磁道，并进行并行读/写，有效地提高了磁盘的I/O速度。 这种结构的磁盘主要用于大容量磁盘上。

2. **移动头磁盘**：**每一个盘面仅配有一个磁头**，也被装入磁臂中。为能访问该盘面上的所有磁道，该磁头必须能移动以进行寻道。可见，移动磁头仅能以串行方式读/写，致使其I/O速度较慢。

磁盘访问时间

- 磁盘访问时间

  = 定位时间（寻道时间+旋转延迟时间）+ 传输时间

1. 寻道时间 Ts

-把磁臂(磁头)移动到指定磁道上所经历的时间。该时间是启动磁臂的时间s与磁头移动n条磁道所花费的时间之和， 即 Ts=m × n+s，其中，m是一常数，与磁盘驱动器的速度有关，对一般磁盘，m=0.2；对高速磁盘，m≤0.1,磁臂的启动时间约为2 ms。 这样，对一般的磁盘， 其寻道时间将随寻道距离的增加而增大， 大体上是5~30 ms。

磁盘访问时间

2. 旋转延迟时间 T_τ

指定扇区移动到磁头下面所经历的时间。

I. 对于硬盘，典型的旋转速度大多为5400 r/min，每转需时11.1 ms，平均旋转延迟时间 T_τ 为5.55 ms；

II. 对于软盘，其旋转速度为300 r/min或600 r/min，这样，平均 T_τ 为50~100 ms。

磁盘访问时间

3. 传输时间 T_t

把数据从磁盘读出或向磁盘写入数据所经历的时间。

T_t的大小与每次所读/写的字节数b和磁盘的旋转速度有关：

$$T_t = \frac{b}{rN}$$

其中，r为磁盘每秒钟的转数；N为一条磁道上的字节数，当一次读/写的字节数相当于半条磁道上的字节数时，T_t与T_τ相同，因此，可将访问时间Ta表示为：

$$T_a = T_s + \frac{1}{2r} + \frac{b}{rN}$$

磁盘

- 磁盘是可移动或撤换的设备
- 通过I/O总线驱动与计算机相连
  - EIDE, ATA, SATA, USB, Fiber Channel, SCSI

- 通过I/O总线传输数据，谁实现？

1. **主机控制器**：在计算机内部总线末端的控制器
2. **磁盘控制器**：建立驱动或存储器阵列

磁带

- 永久存储大规模数据；
- 比磁盘慢；
- 数据迁移率与磁盘相当；
- 典型存储容量为20~200GB ；
- 用来备份，存储极少使用数据，系统间迁移媒介；
- 缠绕或反缠绕经过读写头；
- 通常技术是按带宽分为4mm，8mm和19mm，按技术分为LTO-2 和SDLT等。

磁盘附属

是计算机访问磁盘存储方式

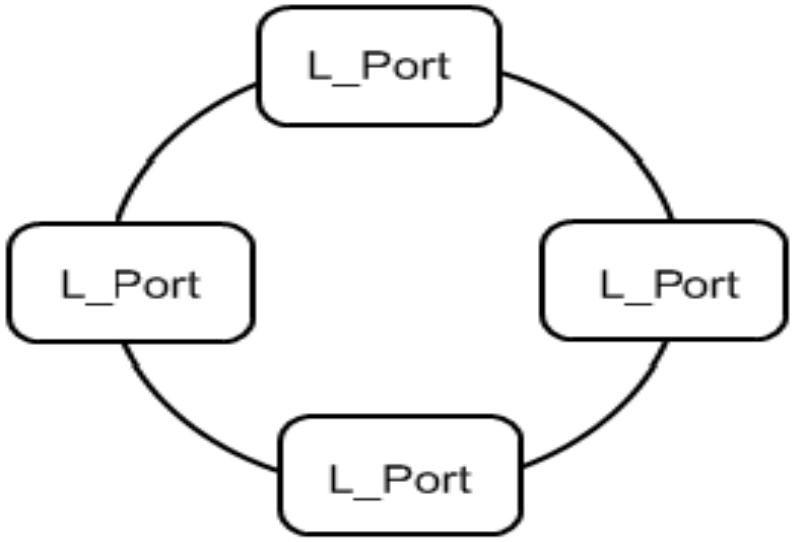
1. 主机附属存储 – 通过 I/O 端口
2. 网络附属存储 – 通过 DFS(Distributed File System)

1. 主机附属存储

- IDE，ATA允许每条I/O总线最多连接2个端口，
- SCSI在一根电缆上可多达16台设备
  - 一个SCSI引导器（控制器）请求操作
  - 15个存储设备（SCSI目标）
  - 每个目标能够提供8个逻辑单元的访问（磁盘附属于设备控制器）

磁盘附属

- 光纤通道（FC）是一种高速串行结构
- 组成大的交换网络，具有24位地址空间
- 裁定循环（FC-AL），可访问126个设备；

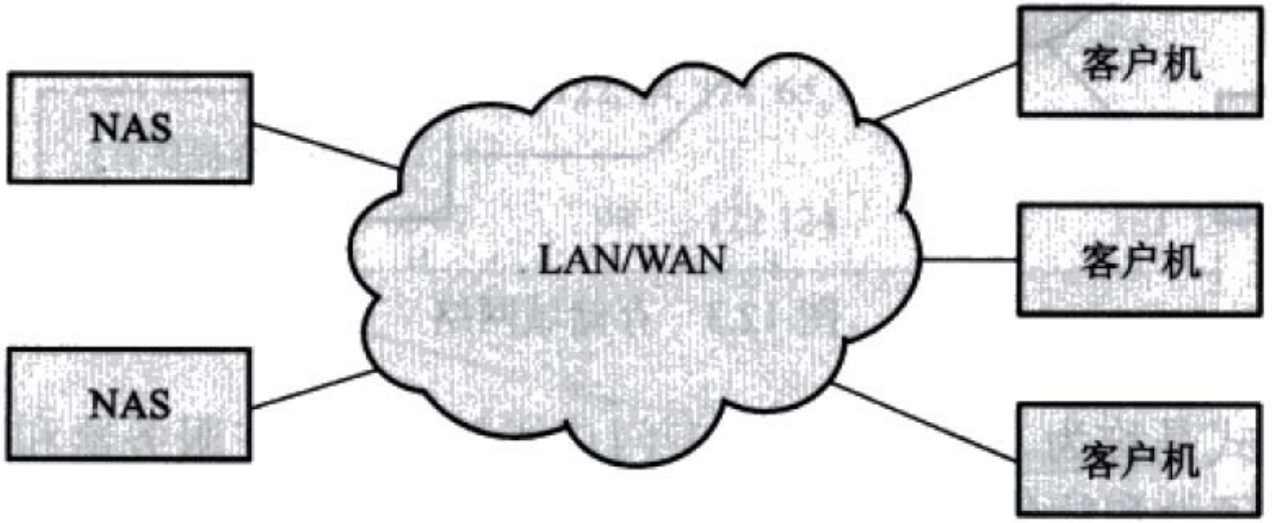


Arbitrated loop (FC-AL)

磁盘附属

2. 网络附属存储(networked-attached storage, NAS)

- 通过网络存储（网络协议），NFS和 CIFS是公共协议；
- 通过远程过程调用实现 (RPC: Remote Procedure Call)；
- iSCSI协议使用IP 网络传送SCSI协议

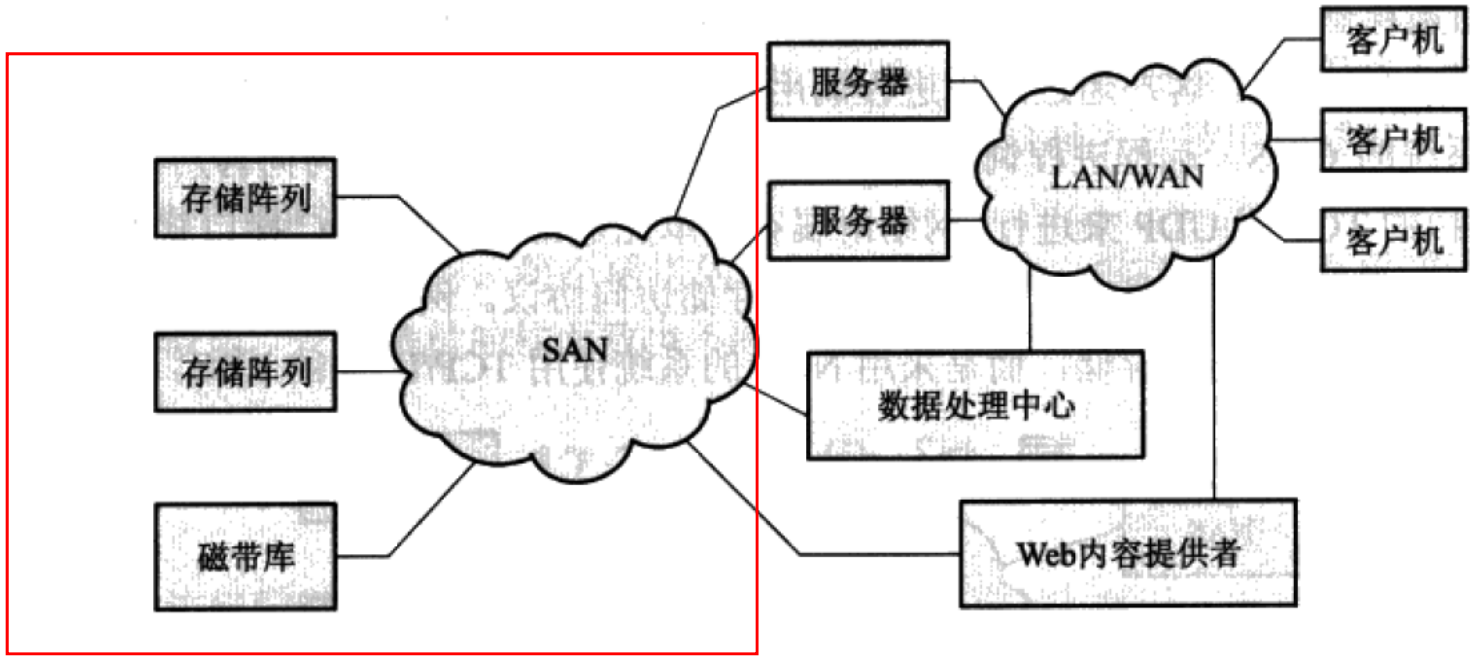


存储区域网络

SAN: storage area network

服务器与存储器单元之间的私有网络，采用存储协议，通常在大规模存储环境下使用；

多台主机附属于多个存储数组 – 柔性



12.3 磁盘调度

有效使用磁盘意味着要有较快的**访问速度**和较宽的**磁盘带宽**。

磁盘带宽：所传递的总字节数除以从服务请求开始到最后传递结束时的总时间。

访问时间有以下两个主要部分

- 寻道时间
- 旋转延迟

} 定位时间

- 传输时间

最小化寻道时间

- 寻道时间与寻道距离有关

磁盘调度

调度磁盘I/O请求服务，采用好的方式能够提高访问时间和带宽。

磁盘I/O系统调用请求，指定了一些信息：

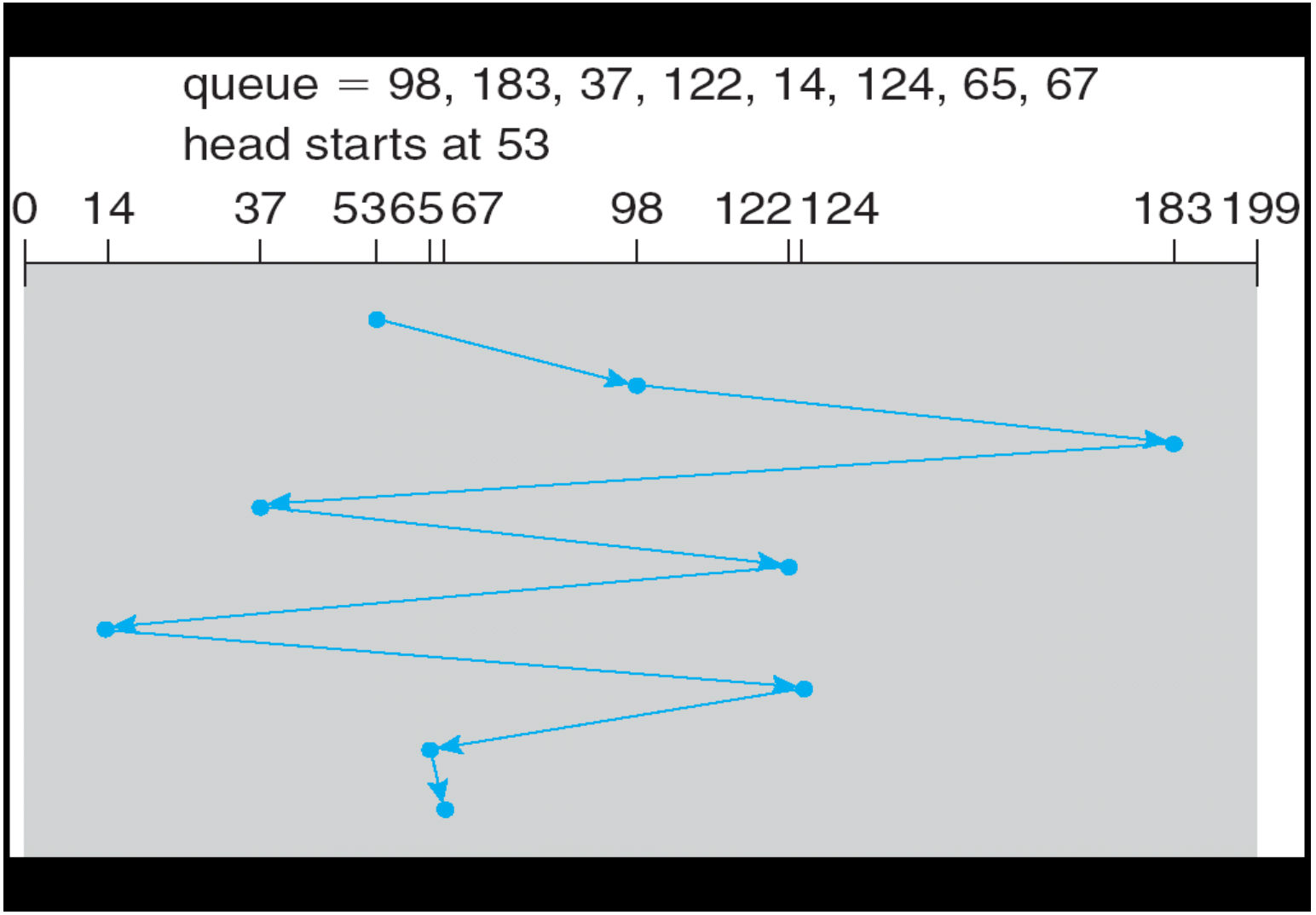
- 输入/输出
- 磁盘地址
- 内存地址
- 所传输的扇区数

调度算法

11.4 磁盘调度算法

1. FCFS 算法
2. SSTF 算法
3. SCAN 和 C-SCAN 算法
4. LOOK 和 C-LOOK 算法
5. N-STEP-SCAN 和 F-SCAN

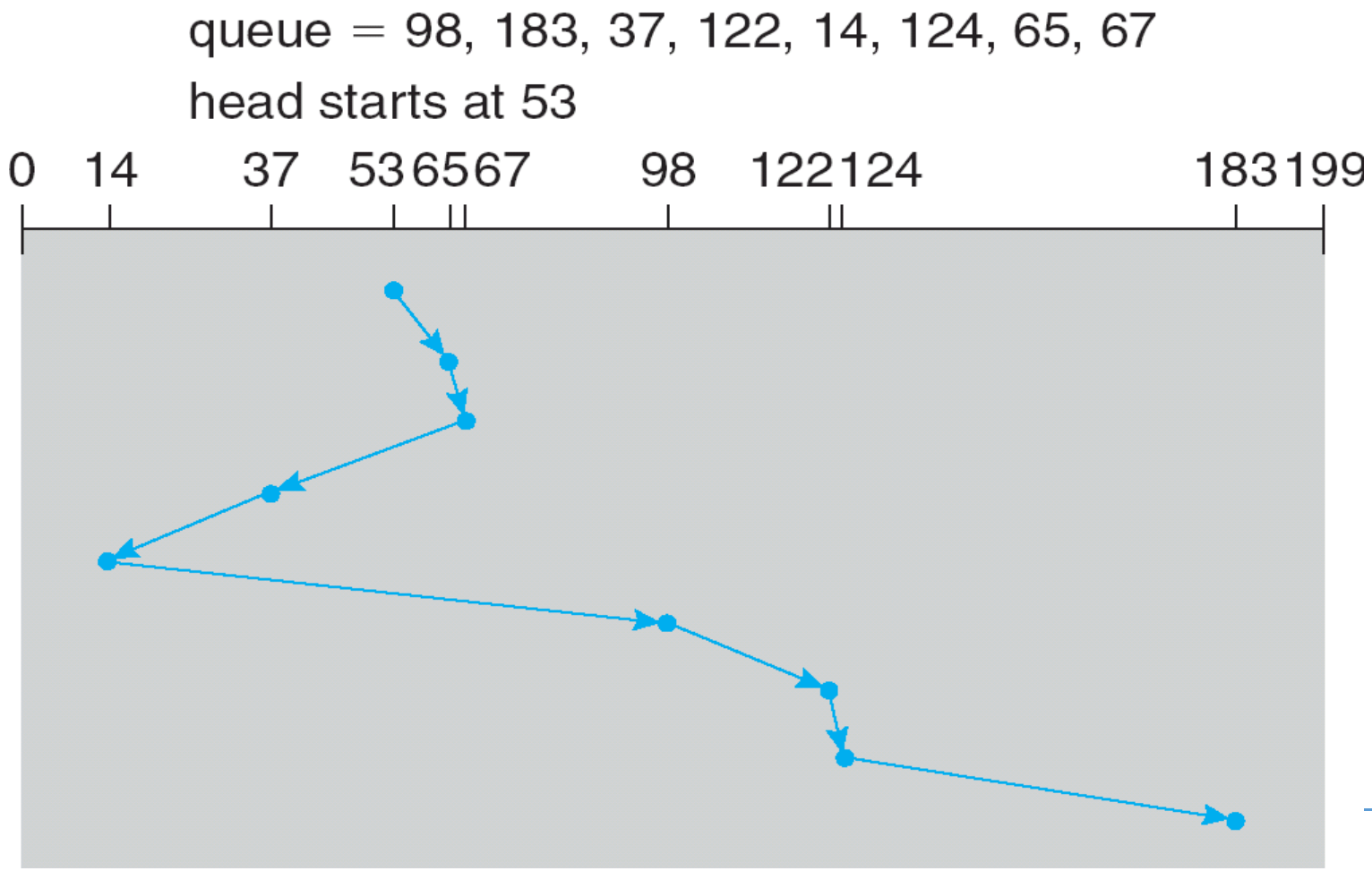
（1）FCFS（先来先服务算法）



（2）SSTF（最短寻道时间优先算法）

从当前磁头位置选择最短寻道时间的请求

SSTF（shortest-seek-time-first）基本上是一种最短作业优先（SJF）调度，与SJF调度一样，它可能导致某些请求的饥饿。

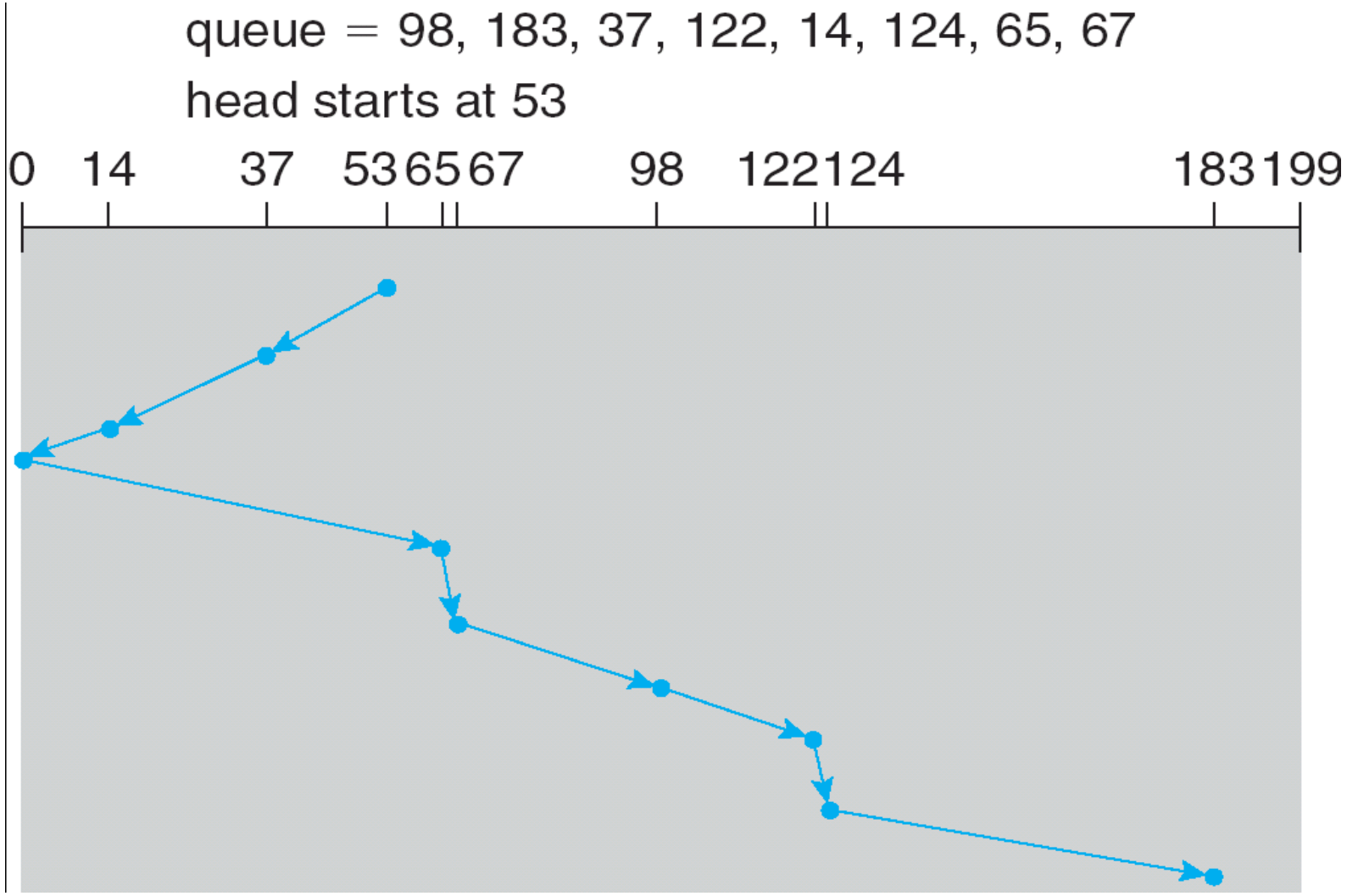


(3) SCAN调度

算法（也叫做“电梯”算法）

- 磁臂从磁盘的一端开始向另一端移动；
- 同时当磁头移过每个柱面时，处理位于该柱面上的服务请求；
- 当到达另一端时，磁头改变移动方向，继续处理；
- 磁头在磁盘上来回扫描。

(3) SCAN调度


queue = 98, 183, 37, 122, 14, 124, 65, 67
head starts at 53
0
14
37
53
65
67
98
122
124
183
199

(4) C-SCAN算法

是SCAN调度的变种，主要提供一个更为均匀的等待时间。

算法

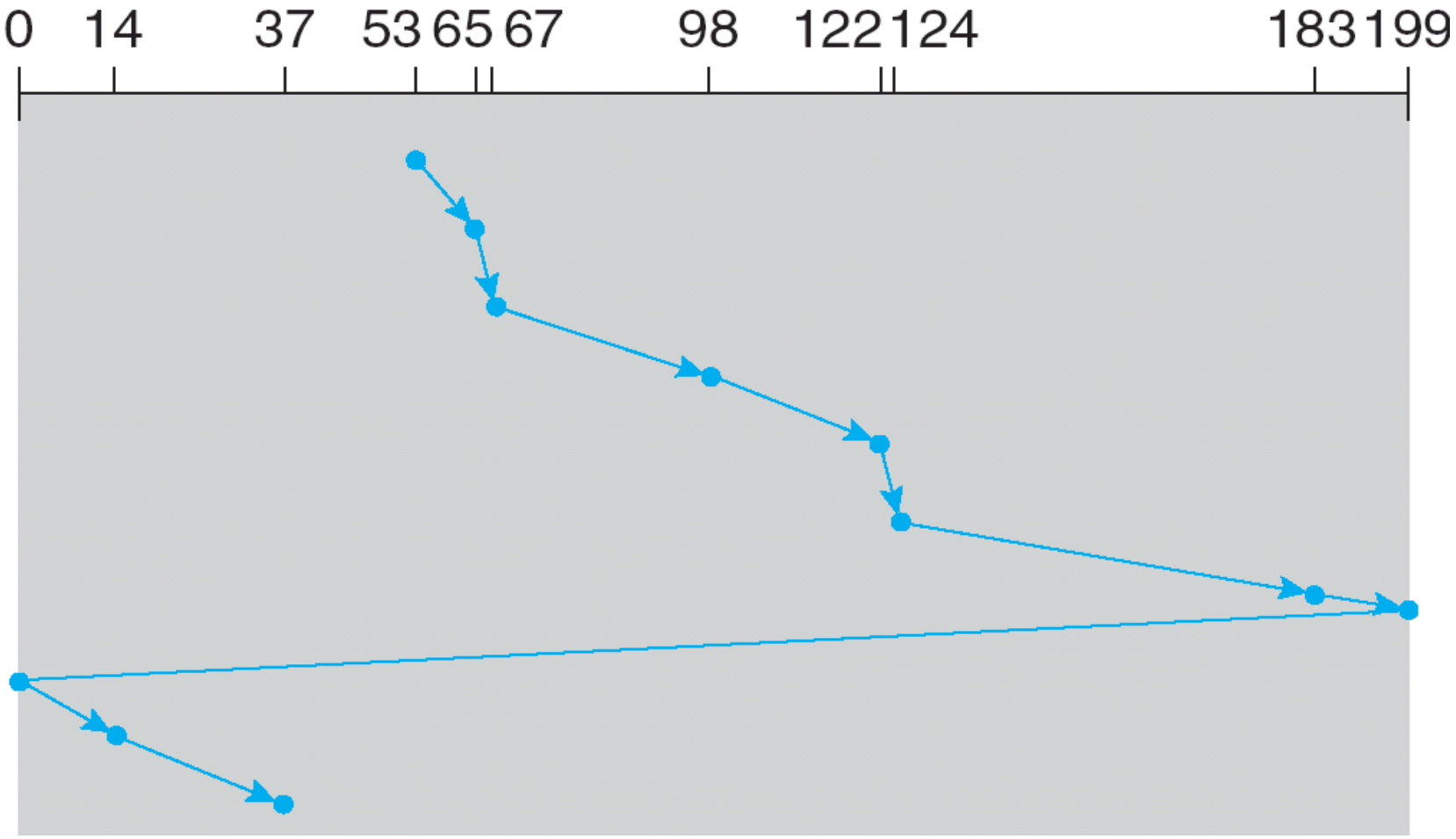
- 磁头从磁盘一端向另一端移动；
- 随着移动而不断地处理请求；
- 当磁头到头时，立即返回到磁盘开始端；
- 返回时并不处理请求。

C－SCAN调度算法基本上将柱面当做一个环链，以将最后柱面和第一柱面相连。

(4) C-SCAN算法

queue = 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53



(5) LOOK调度与C-LOOK调度

事实上，SCAN与C－SCAN算法都不是那样实现的。

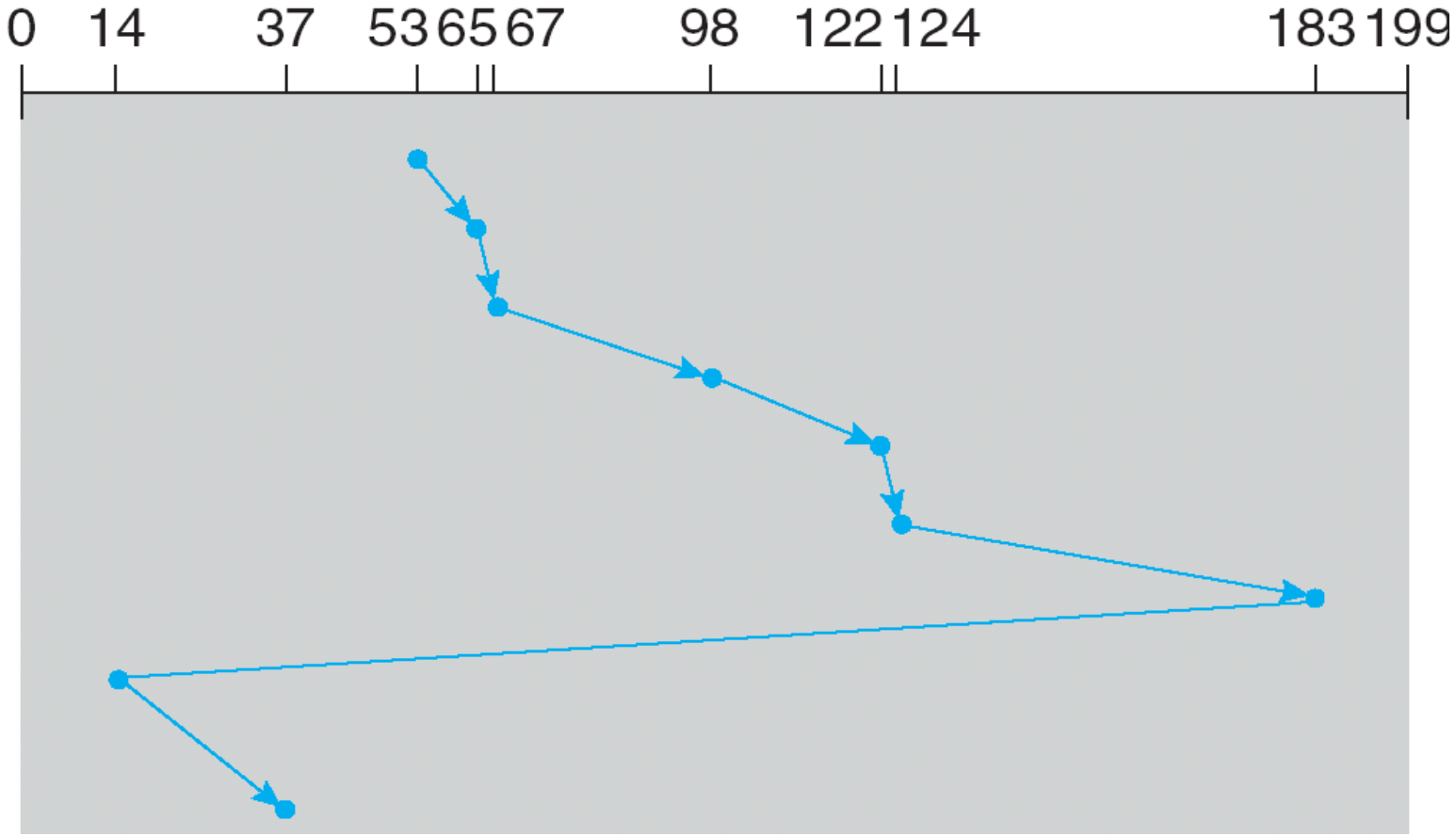
通常，磁头只移动到一个方向上最远的请求为止。接着，它马上回头，而不是继续到磁盘的尽头。

这种形式的SCAN和C-SCAN称为LOOK和C-LOOK调度。

C-LOOK

queue 98, 183, 37, 122, 14, 124, 65, 67

head starts at 53



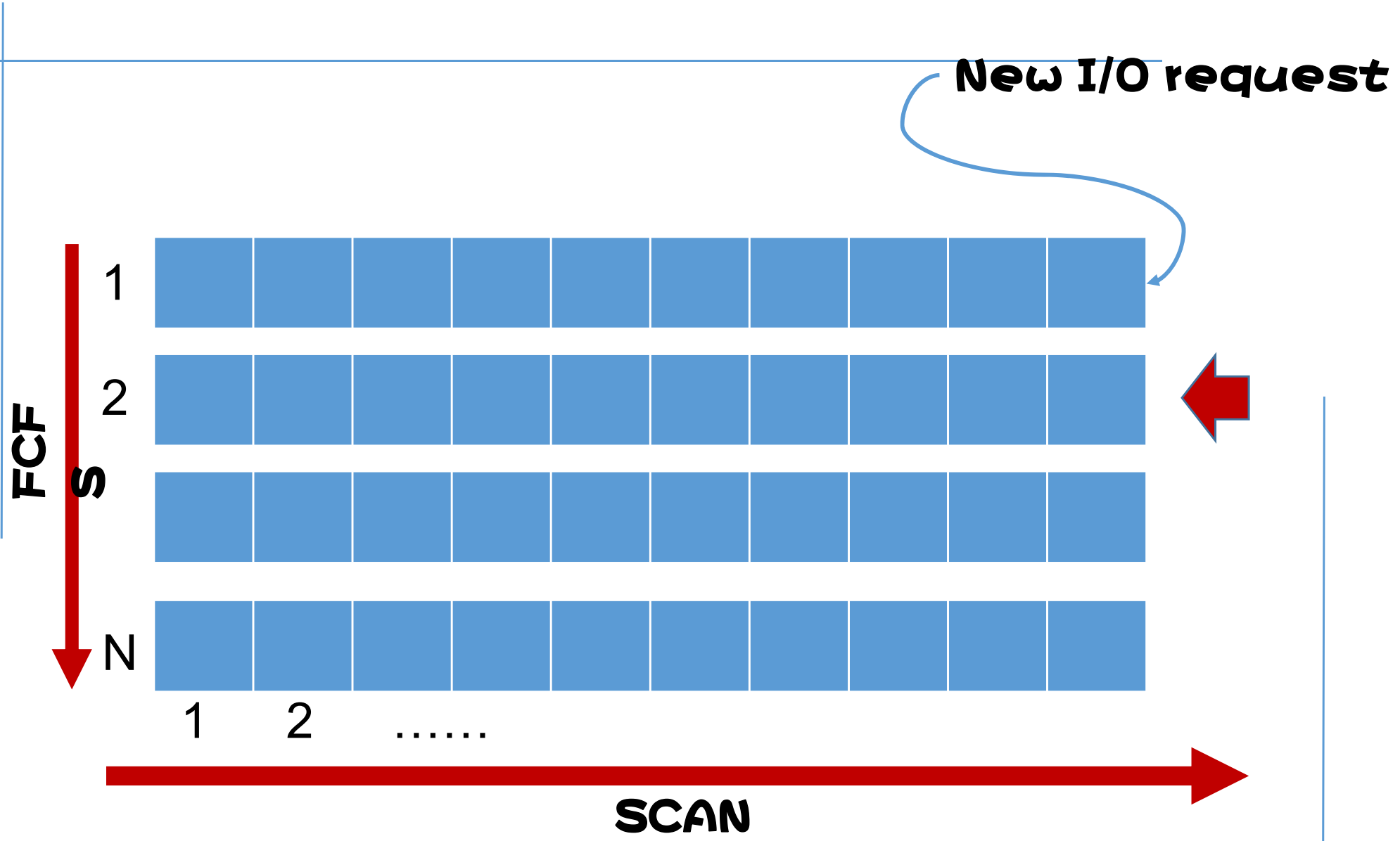
(6) N-Step-SCAN和FSCAN调度算法

N-Step-SCAN算法

- 在SSTF、 SCAN、CSCAN几种调度算法中， 都可能出现磁臂停留在某处不动的情况， 例如，有一个或几个进程对某一磁道有较高的访问频率， 即这个(些)进程反复请求对某一磁道的I/O操作，从而垄断了整个磁盘设备。
- 我们把这一现象称为“磁臂黏着”Armstickiness。在高密度磁盘上容易出现此情况。

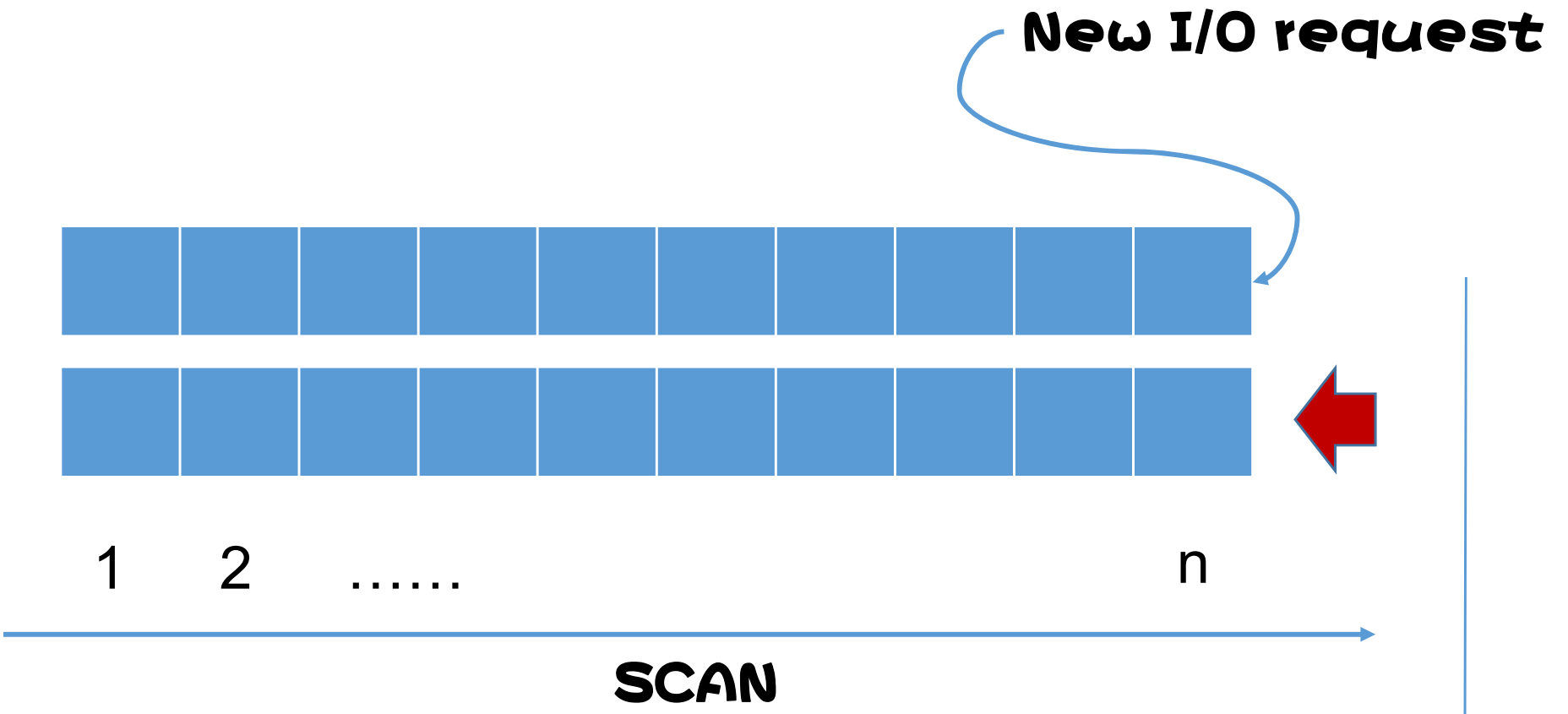
(6) N-Step-SCAN和FSCAN调度算法

- N步SCAN算法是将磁盘请求队列分成若干个长度为N的子队列，磁盘调度将按FCFS算法依次处理这些子队列。
- 而每处理一个队列时又是按SCAN算法，对一个队列处理完后，再处理其他队列。
- 当正在处理某子队列时，如果又出现新的磁盘I/O请求，便将新请求进程放入其他队列，这样就可避免出现粘着现象。
- 注：当N值取得很大时，会使N步扫描法的性能接近于SCAN算法的性能；当N=1时，N步SCAN算法便蜕化为FCFS算法。


New I/O request
1
2
N
FCFS
1
2
.....
SCAN

(7) FSCAN调度算法

- FSCAN算法实质上是N步SCAN算法的简化， 即FSCAN只将磁盘请求队列分成两个子队列。
- 一个是由当前所有请求磁盘I/O的进程形成的队列，磁盘调度按SCAN算法进行处理。
- 在扫描期间，将新出现的所有请求磁盘I/O的进程， 放入另一个等待处理的请求队列。这样，所有的新请求都将被推迟到下一次扫描时处理。


New I/O request
1
2
……
n
SCAN

磁盘调度算法的选择

- SSTF（最短寻道时间优先算法）较为普通且很有吸引力
- SCAN和C-SCAN对磁盘负荷较大的系统会执行得更好，这是因为它不可能产生饥饿问题。
- 对于任何调度算法，性能依赖于请求的类型与数量
- 磁盘服务请求很大程度上受文件分配方法的影响
- 磁盘调度算法应作为一个操作系统的独立模块，这样如果有必要，它可以替换成另一个不同的算法。
- SSTF或LOOK是比较合理的缺省算法。
- 其它考虑
  - 旋转时间
  - 磁盘请求优化

其它问题

操作系统比较难以调度来改善旋转等待，这是因为现代磁盘并不透露逻辑块的物理位置。

事实上OS对请求服务顺序还有其他限制，如：

- 按需分页比I/O的优先级高
- 有时写操作比读操作更重要

12.4 磁盘管理-磁盘格式化

低级格式化或**物理格式化**（Raw/Low Format）- 将磁盘分成磁盘控制器能读与写的扇区

1. 扇区的数据结构
    1. 头，包含扇区号码；
    2. 数据区域；
    3. 尾部，包含纠错代码（error-correcting code，ECC）

ECC

1. 在写，更新时
2. 在读，计算和检查时
    - 软错误 - correct
    - Corrupted - dealt later

磁盘管理

为了让磁盘能够存储文件，操作系统还必须在磁盘上记录自己的数据结构，即

I. 分成一个或多个柱面（分区）；

II. 逻辑格式化或“做文件系统”

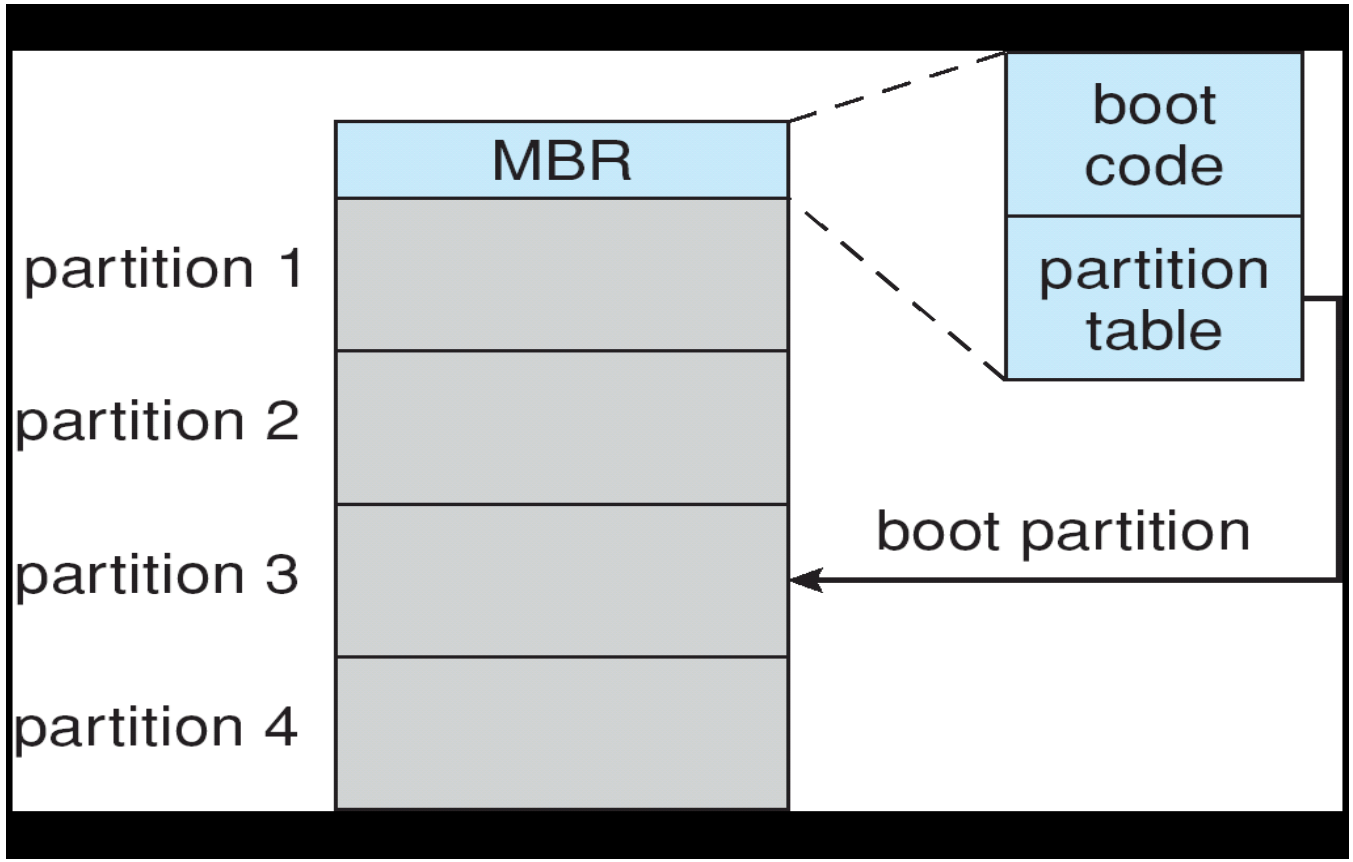
- 映射空闲/可分配空间
- 初始化空目录

引导块

- 绝大多数系统只在启动ROM中保留一个很小的自举装入程序（bootstrap），其作用是初始化，启动OS（保存在磁盘的启动块上）。

Windows 2000中从磁盘上启动

Master Boot Record



坏块 (bad block)

对于简单磁盘如使用IDE控制器的磁盘，坏扇区可手工处理：format，chkdsk。

扇区备用（sector sparing or forwarding）

- 低级格式化将备用扇区放在一边；
- 当OS试图读一个坏块时；
- 控制器检查 ECC，并报告给 OS；
- 当OS重启时，告诉控制器替换；
- 控制器将转移到新的地址
- 备用扇区和备用柱面

扇区滑动（sector slipping）方法

10	11	12	13	14	15	16	17	18	spare
defective		11	12	13	14	15	16	17	18

12.5 交换空间管理

OS的低级任务

1. 交换空间概念： 虚拟内存使用磁盘空间作为主存的扩展交换空间的使用

- Swap：保存整个进程映像，包括代码段和数据段
- Switch：存储换出内存的页

注：交换空间太小容易造成死机现象

交换空间管理

2. 交换空间的位置

I. 交换空间在普通文件系统上加以创建。通常是文件系统内的一个简单大文件（如Windows）。这种方式实现简单但效率较低。

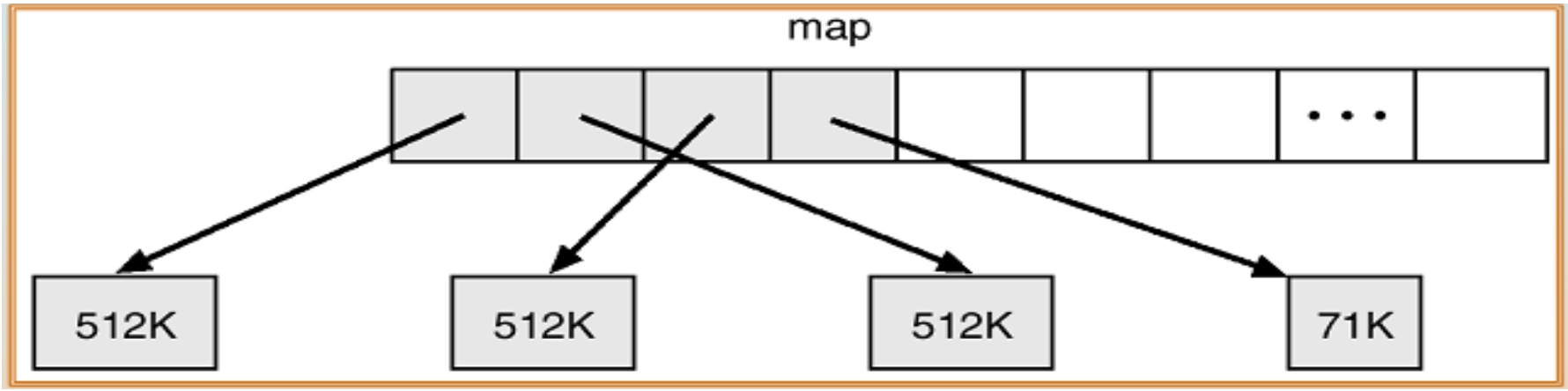
II. 交换空间创建在独立的磁盘分区上（如Unix/Linux）。

III. Option: 有些OS较为灵活，可以由系统管理员来选择使用以上哪种方式。

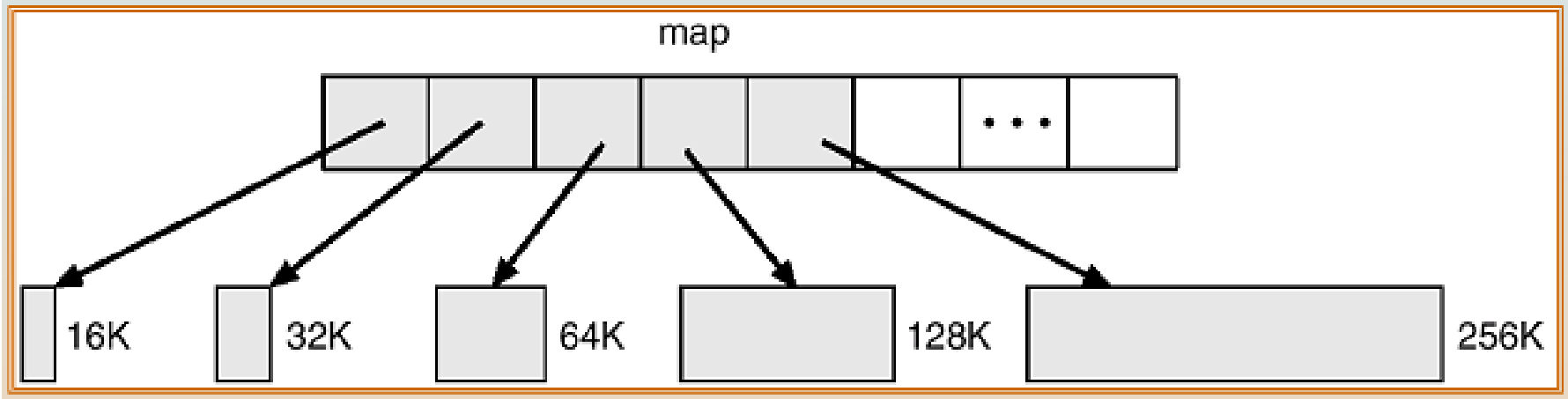
3. 交换空间管理

- 4.3 BSD在进程启动的时候分配交换空间，用来保存文本段（代码段）和数据段
- 内核使用交换映射来跟踪交换空间的使用
- Solaris 2当页被强制换出内存的时候分配交换空间。

BSD系统的代码段交换表



BSD系统的数据段交换表

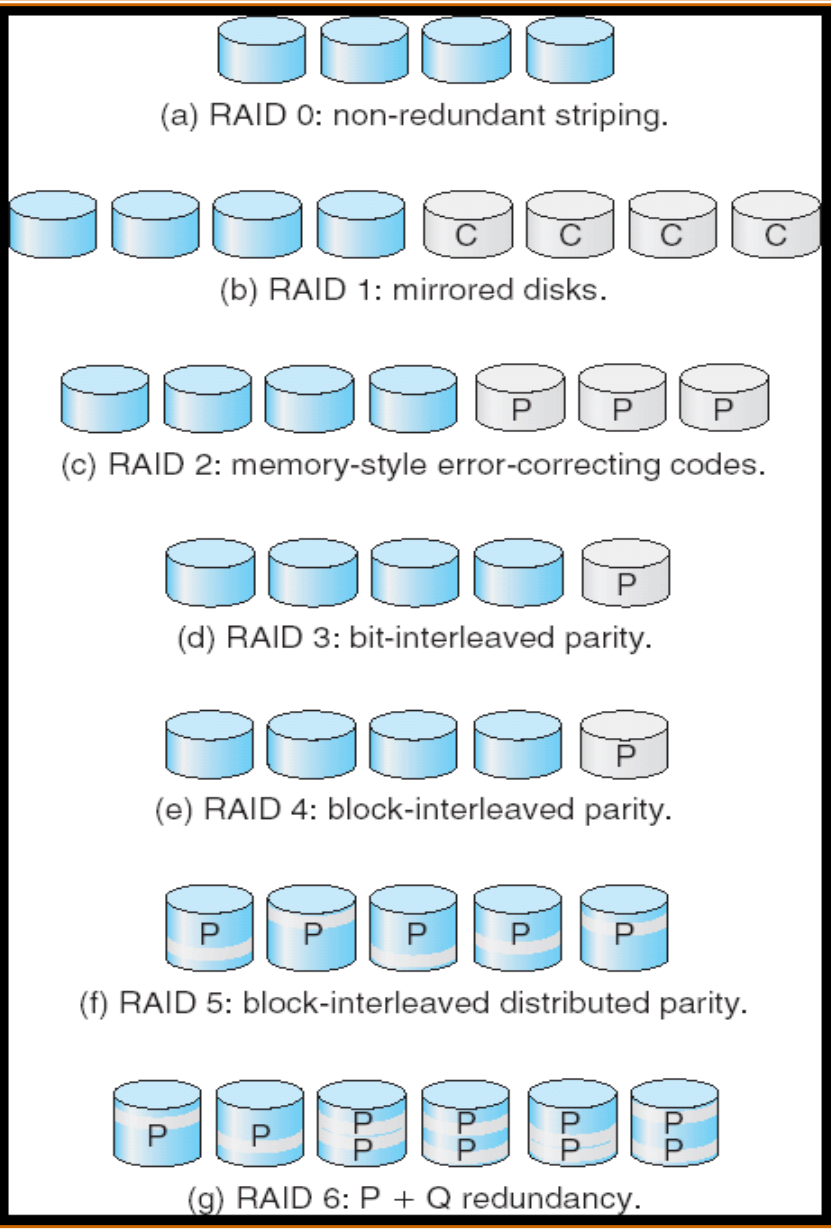


12.6 RAID结构

RAID (Redundant Array of Independent Disks)
– 多个磁盘通过冗余实现可靠性

1. 通过冗余改善可靠性
   - 假设单个磁盘出错的概率为α，则n个磁盘出错的概率为α/n。如果只存储数据的一个拷贝，只要n个磁盘中的一个磁盘出错，数据就出现错误。因此n个磁盘的出错率大于1个磁盘的出错率。
   - 可靠性问题的解决方法是引入冗余。
     - 镜像
2. 通过并行处理改善性能
   - 数据分散：通过在多个磁盘上分散数据，能够改善传输率。
     - 位级分散：以位为单位分散数据
     - 块级分散：以块为单位分散数据

RAID级别



(a) RAID 0: non-redundant striping.

(b) RAID 1: mirrored disks.

(c) RAID 2: memory-style error-correcting codes.

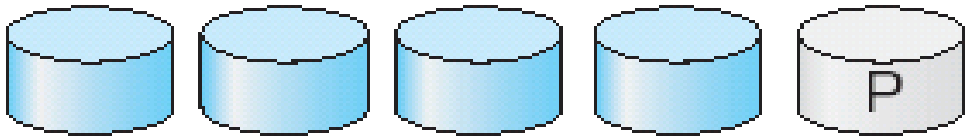
(d) RAID 3: bit-interleaved parity.

(e) RAID 4: block-interleaved parity.

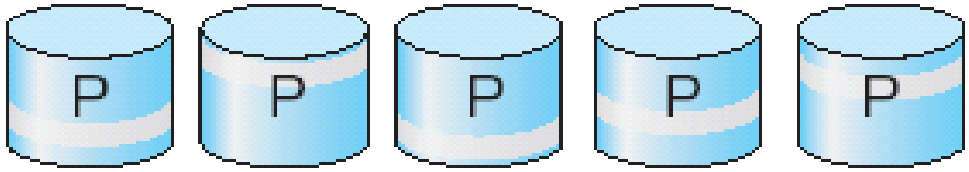
(f) RAID 5: block-interleaved distributed parity.

(g) RAID 6: P + Q redundancy.

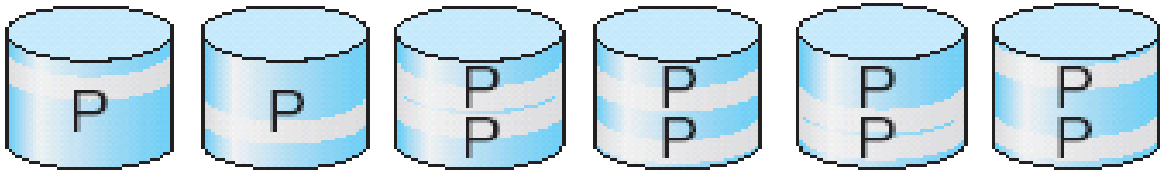
RAID级别（cont.）



(e) RAID 4: block-interleaved parity.

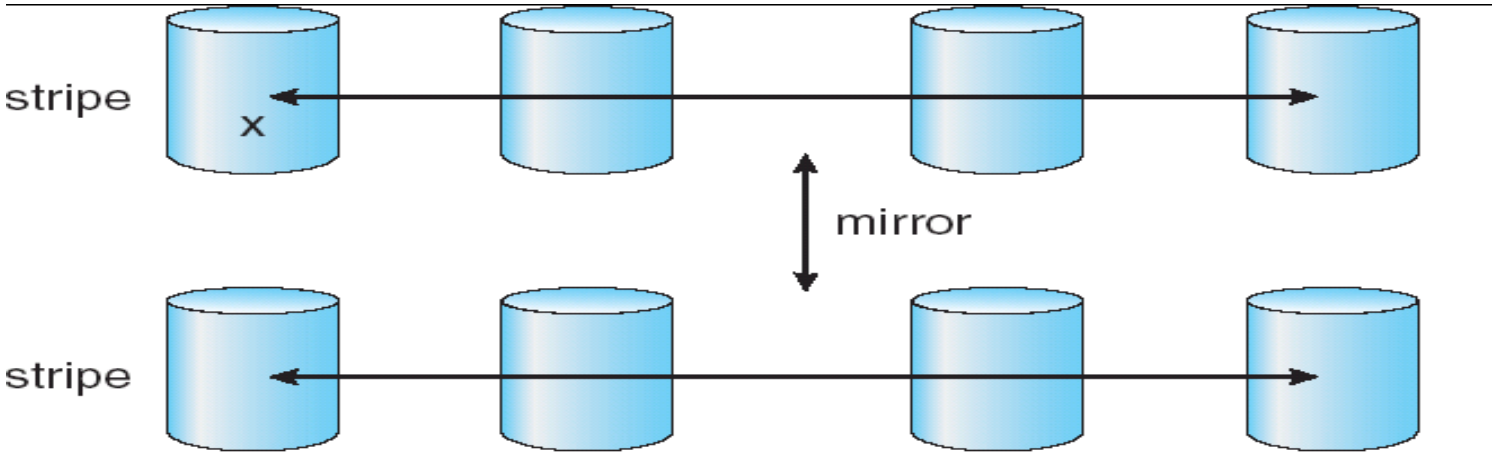


(f) RAID 5: block-interleaved distributed parity.

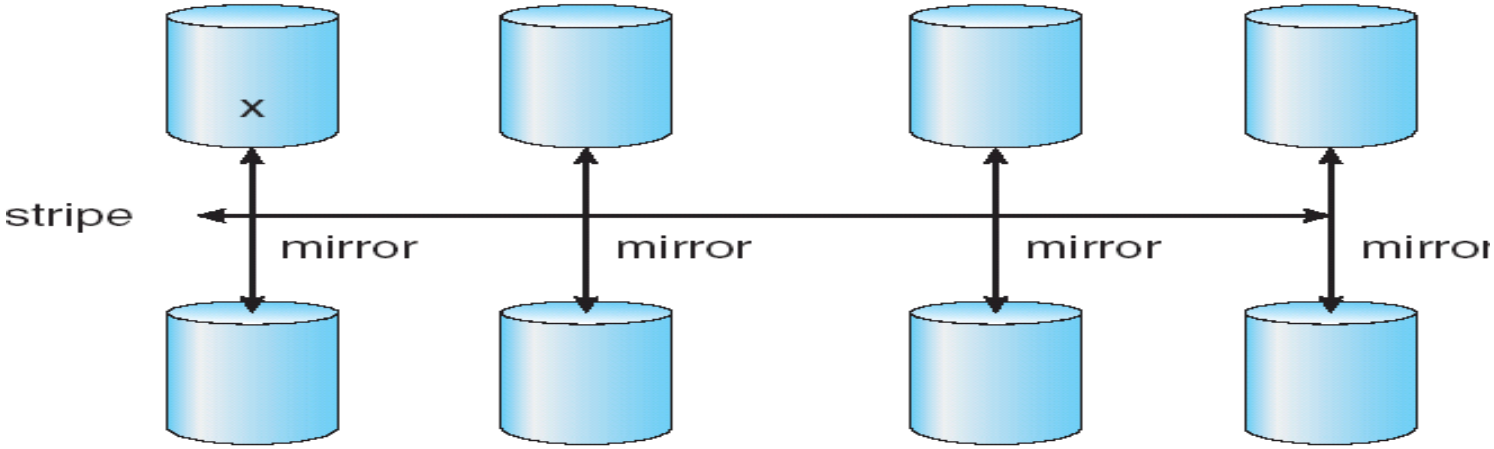


(g) RAID 6: P + Q redundancy.

RAID 0 + 1 和 1 + 0



a) RAID 0 + 1 with a single disk failure.



b) RAID 1 + 0 with a single disk failure.

稳定存储实现

预写式日志方案要求稳定存储。

磁盘写可能发生的情况：

- 成功完成
- 部分失败
- 全部失败

系统为每个逻辑块维护2个物理块：

- 将信息写到第一个物理块上；
- 把同样的信息写到第二个物理块上；
- 当第二块写完后才声明操作完成。

稳定存储实现

从错误中恢复

- 比较每对物理块；
- 两块相同且没有检测到差错 => OK
- 一块中包含错误=> 用另一块替换
- 都没有错误，当两块内容不同=> 用第二块替换第一块。

存储阵列中增加NVRAM作为缓存

12.7 第三级存储结构

可移动磁盘

- 软盘
- 磁光盘（激光与磁场同时作用于盘面上的磁性材料）
- 光盘：相位变化盘（CD-RW、DVD-RW），读写盘（read-write disk）、一次写多次读的盘（Write-once, read-many-times, WORM, 如CD-R和DVD-R）、一次写盘（如CD-ROM、DVD）

第三级存储结构

磁带

- 容量大，但随机访问要比磁盘寻道时间慢很多

未来技术

- 全息照相存储器；
- 另一种热门研究的存储技术是基于微电子机械系统（MEMS）。

操作系统支持

OS的两个主要任务是管理物理设备和为应用程序提供一个虚拟机器的抽象。

对于磁盘，OS提供了两种抽象

- 生设备（raw device）
- 文件系统

应用接口

- 对磁盘，基本操作为read、write、seek
- 对磁带，则没有seek，有locate操作
- 绝大多数磁带驱动器有一个read position操作以返回磁头所处的逻辑块号
- 绝大多数磁带驱动器，写一块具有副作用：即会删除写位置之后的所有内容。

操作系统支持

文件命名

- 有些类型的可移动介质已经标准化，以致于所有计算机按同样方式进行使用。如CD，音乐CD具有统一格式，可为任何驱动器所使用。

层次存储管理

- 自动光盘塔（robotic jukebox）：切换磁带或光盘驱动器内有可移动盘，而无需人工干预
- 层次存储系统扩展了存储层次，使其不但包括内存和外存还包括可移动存储。
- 虽然虚拟内存系统可直接扩展到第三层次存储器，但是事实上这种扩展很少实现。
- 可移动存储通常用来扩展文件系统。

性能（速度、可靠性、价格）

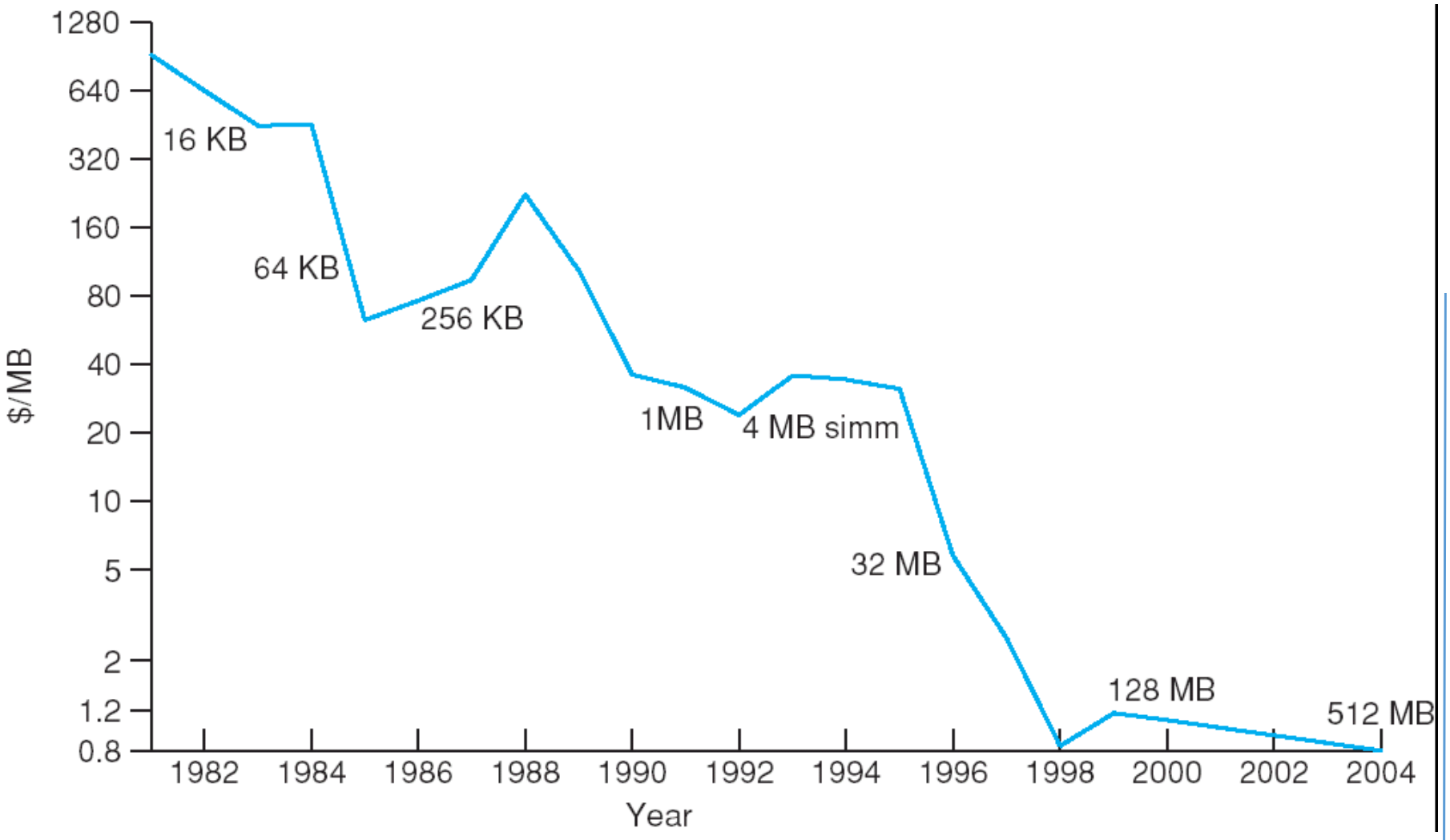
速度

- 带宽
  - 持续带宽：一个大传输的平均速率，即字节数量被传输时间所除
  - 有效带宽：计算整个时间内（包括寻道或定位时间、盘片切换时间等）的平均值。
  - 驱动器的带宽通常指持续带宽
- 延迟
  - 磁盘比磁带快，磁带的随机访问要比磁盘的随机访问慢数千倍。
  - 光盘塔的延迟就更大了。切换盘片耗时

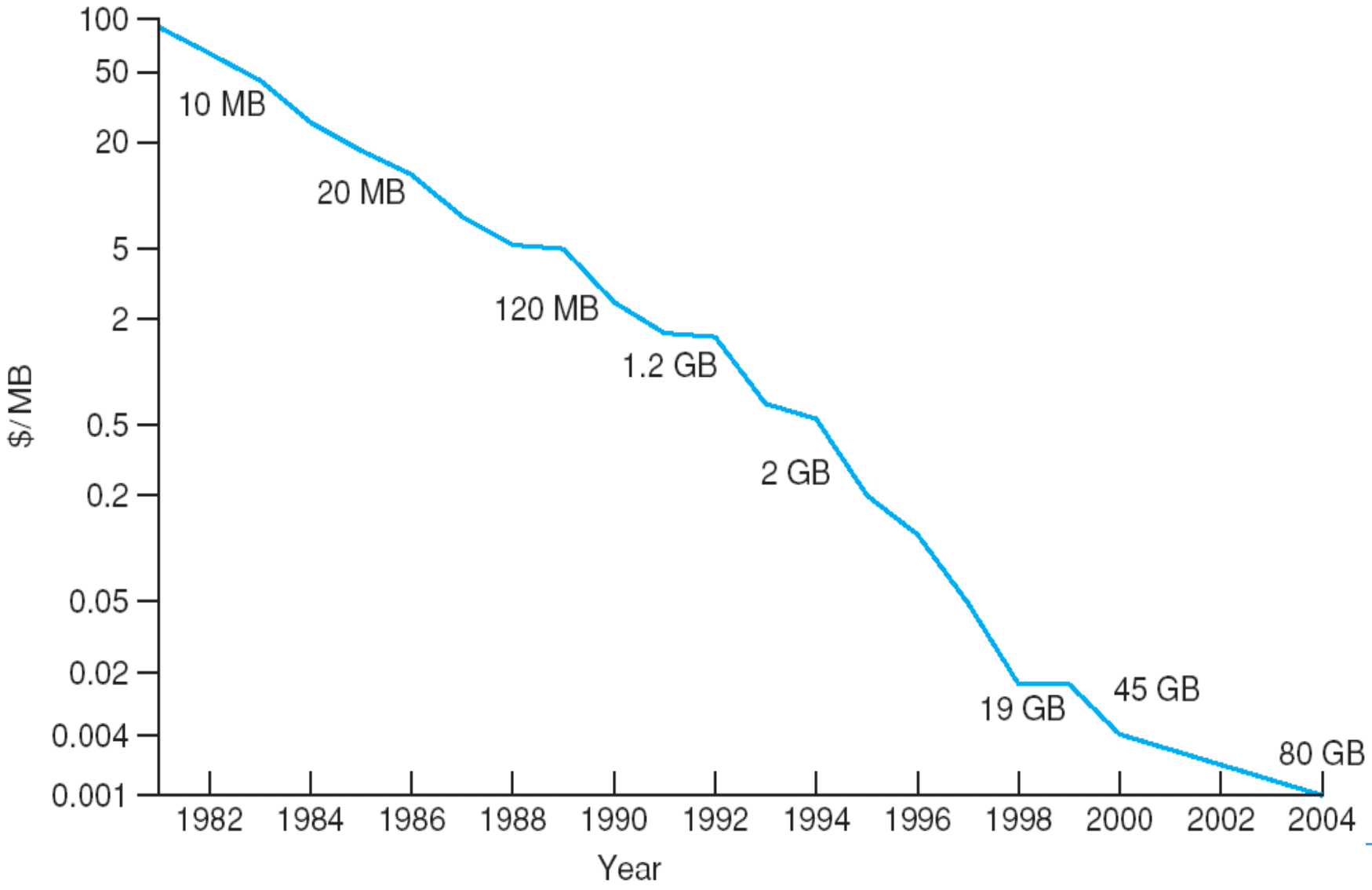
性能（速度、可靠性、价格）

- 可靠性
  - 可移动磁盘与固定磁盘相比，其可靠性要差，因为它更容易受到外界环境的影响。
  - 光盘比磁盘或磁带更为可靠。
- 价格
  - 主存的价格比磁盘存储的价格高很多
  - 硬磁盘每兆字节的价格比磁带的价格更有竞争力。（如果一个磁带驱动器上只用一盒磁带的话）
  - 以往，最便宜的磁带驱动器与最便宜的磁盘驱动器具有相近的存储能力。

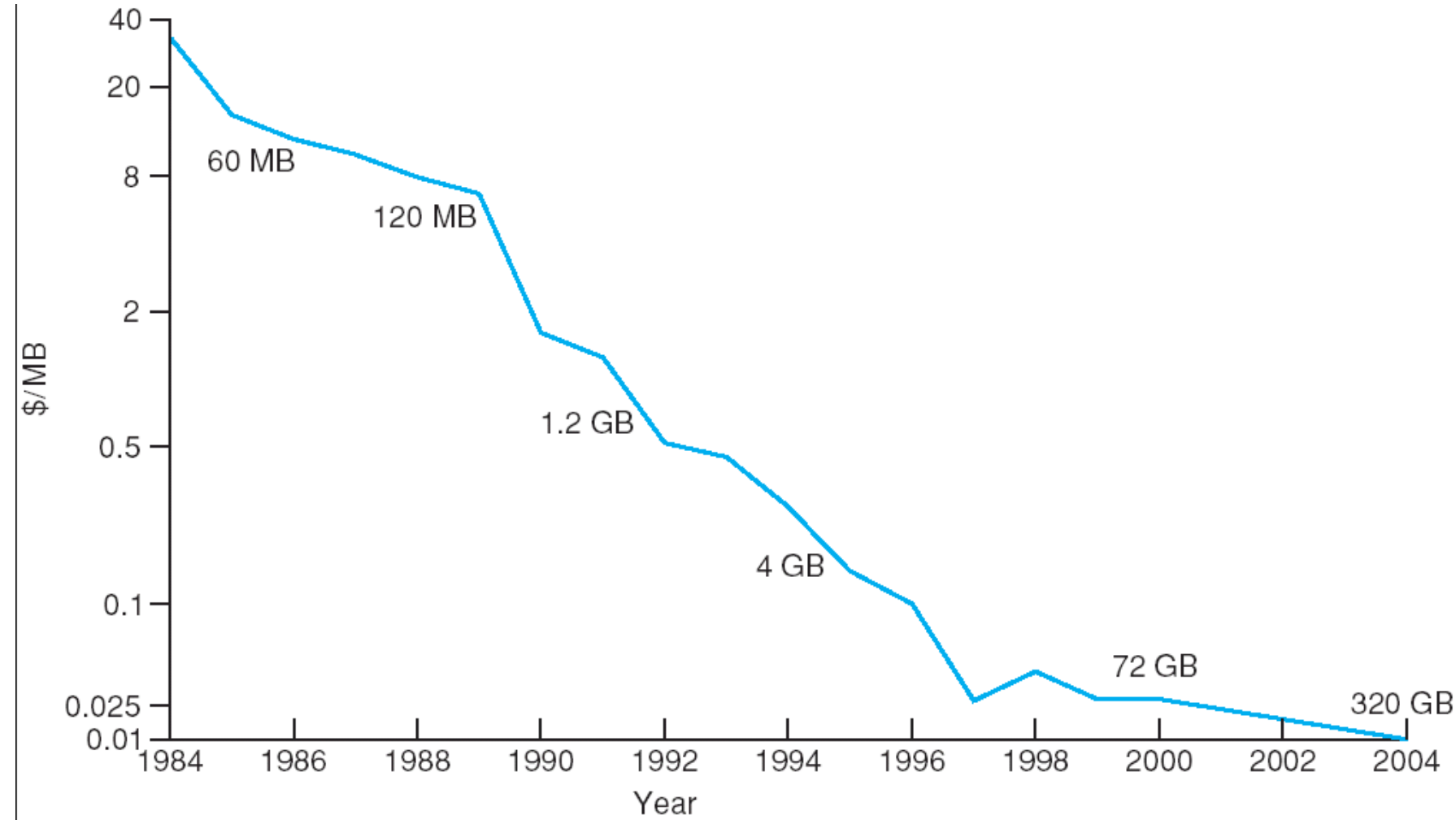
1981年到2004年DRAM价格



1981年到2004年硬磁盘价格（每兆字节）



1984年到2004年磁带价格



Q&A