3. 外存储器

3.1. 磁盘组:多磁盘绕中轴高速旋转

3.1.1. 物理记录定位

1 柱面: 同一垂直位置, 各盘面相应磁道的集合, 编号从外向内0, 1, 2, ..., L

2 磁头:每个盘面一个磁头,编号从上而下0,1,2,...,H,表示有效盘面

3 扇区: 盘面被分为若干大小相同的扇区, 编号为0, 1, 2, ..., n

3.1.2. 磁盘编址

1 逻辑块: 传输数据的最小单位

3.1.3. 盘块物理地址&盘块号

1 物理地址组成: [柱面号L] [磁头/盘面号M] [扇区号N] 2 物理块→唯一盘块号: $B = (i \times M \times N) + (j \times N) + k$

范围: $i=0\ldots,L-1; j=0,\ldots,M-1; k=0,\ldots,N-1$

 $1.(i \times M \times N)$: 达第 i个柱面之前的总扇区数

2. $(j \times N)$: 当前第i柱面,到达第 j个盘面之前的扇区数

3. k: 当前盘面上的扇区号

③ 块号→物理块(一下计算自动强制转换int) M×N=每柱面的扇区总数→B/(M×N)=之前经历了几个完整柱面

B%(M×N)=当前柱面中扇区的偏移数→[B%(M×N)]/N=当前柱面中经历了几个完整盘块

• 柱面号i=B/(M×N)

• 盘面号j=[B%(M×N)]/N 扇区号k=B%N

3.2. 磁盘访问时间

总时间 $T_a = T_S + T_r + T_t$

1 寻道时间 $T_S = mn + s$: 移动磁头到指定磁道所需时间

1. m: 磁头移动一个磁道所需时间,一般磁盘=0.2/高速磁盘≤0.1

2. n: 移动磁道数 3. s: 磁臂启动时间,一般为2ms

2 旋转延迟时间 $T_r=rac{1}{2r}$:磁头到达了磁道后,让扇区移到磁头所需时间

1. r: 磁盘的转速,公式基于假设——平均需要转0.5圈找到扇区

3 传输时间 $T_t=rac{b}{rN}$:读出或写入数据所需的时间

1. b: 所读/写的字节数

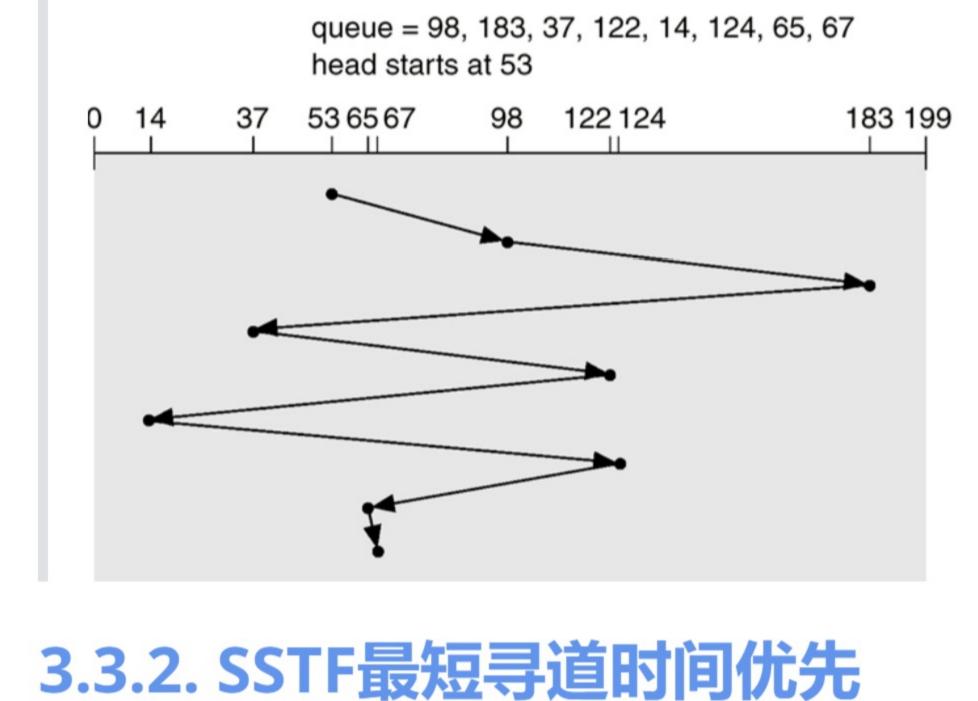
→磁盘IO提速小妙招——提高磁盘性能/设置Cache/选择好的调度算法

2. N: 一条磁道上的字节数→b/N=需要读多少个磁道(转多少圈)

3.3. 磁盘调度

3.3.1. FCFS

如图磁头共移动了640个柱面距离。



2 特点:实质上是SJF(短作业优先),会引起某些请求饥渴

37 53 65 67

0 14

3 示例:磁头移动总距离236柱面

1 含义: 选磁头当前位置, 所需寻道时间最短的请求

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

98 122 124

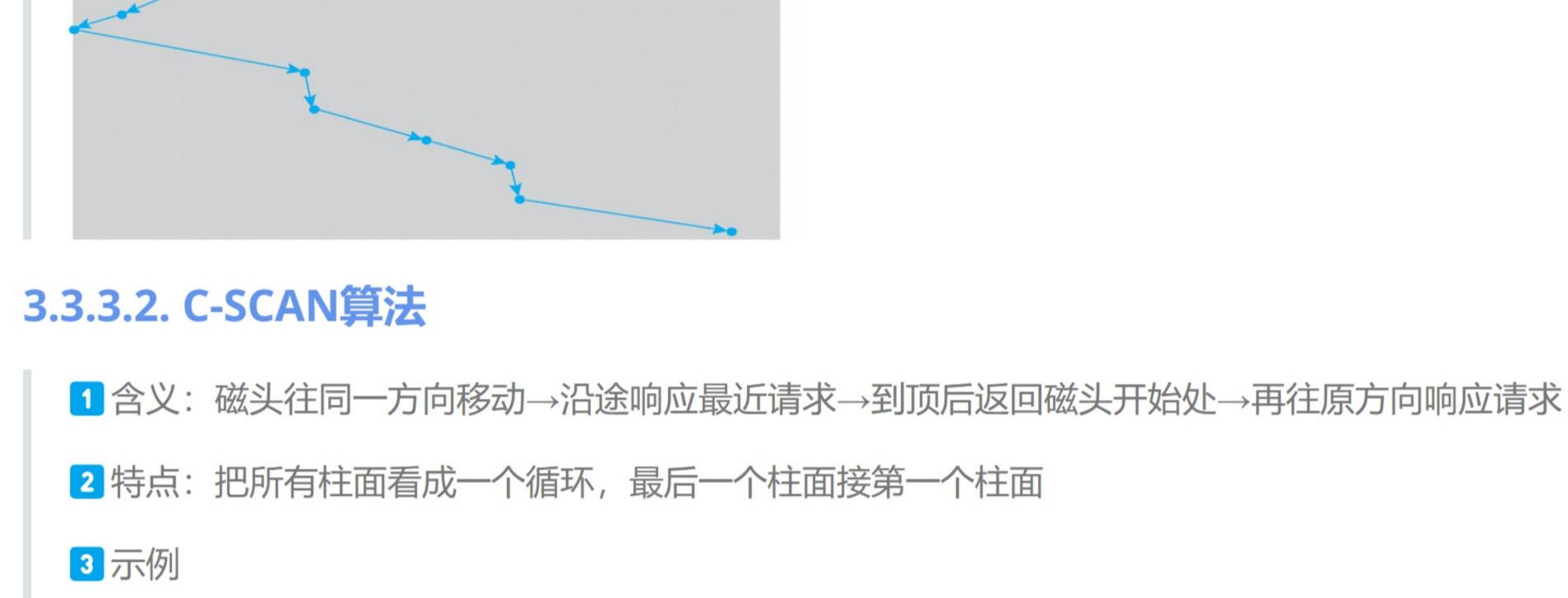
183 199



2 特点: 磁头只会出现一次转向, 转向时要先回到磁盘的最内/最外端

3 示例:磁头移动总距离236柱面

queue = 98, 183, 37, 122, 14, 124, 65, 67 head starts at 53 37 536567 98 122124 183199



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

head starts at 53 53 65 67 98 122 124 14

183 199



只分两个子队列 1. 当前队列: 所有请求磁盘I/O的进程, 按SCAN处理

2 示例

0 14

37 53 65 67

3.3.3.4. F-SCAN算法

3.3.4. LOOK&C-LOOK算法

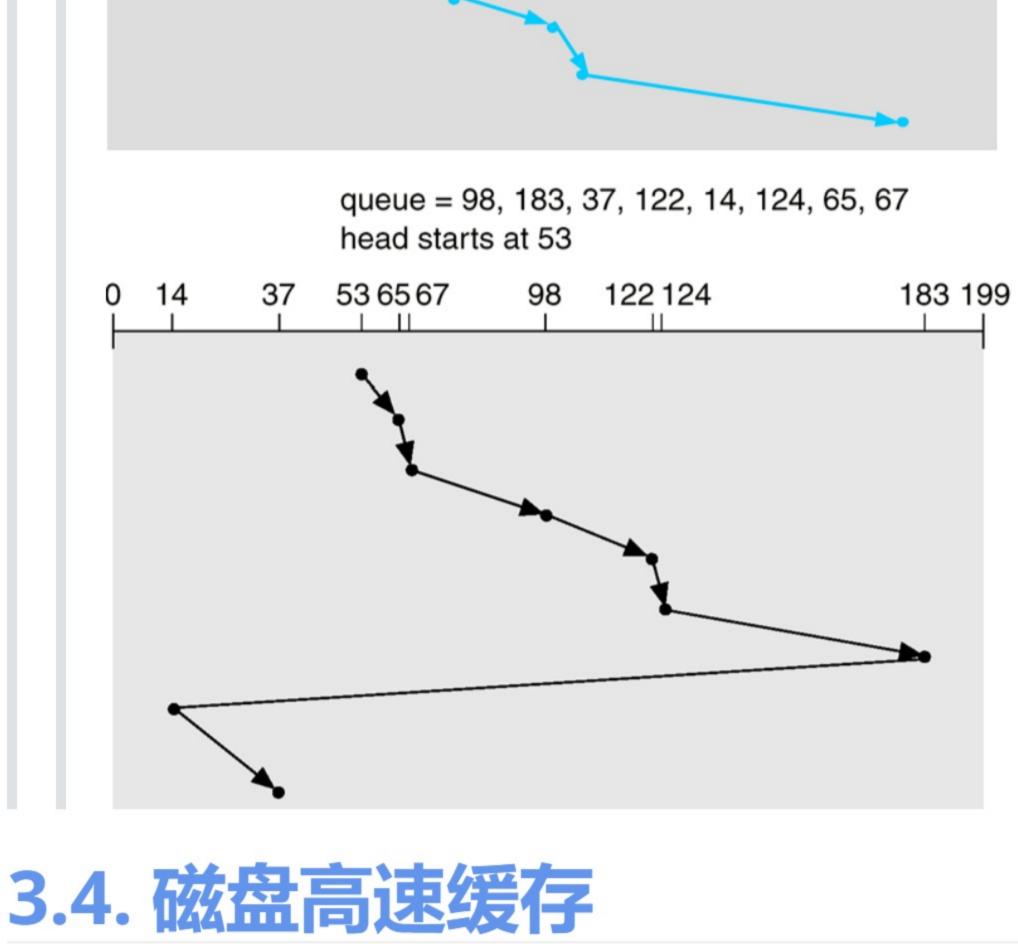
1 含义:区别于SCAN算法,转向/返回的时候不需要回到磁盘最内/最外端

183 199

·Queue=98,183,37,122,14,124,65,67 head stars at 53 ·total head movement of 208 cylinders.

122 124

2. 等待队列:SCAN期间新出现的请求进程



11含义:用内存空间来暂存磁盘数据,物理属于内存,逻辑属于磁盘

2 形式:

1. 内存中开辟一个单独的大小固定的空间 2. 把所有未利用内存空间变为一个缓冲池

3 数据交付:如何将磁盘Cache数据给请求的进程

1. 数据交付: Cache数据→请求进程的内存工作区 2. 指针交付:将指向Cache某区域的指针,交给请求进程

4 置換算法: LRU