

第45 - 46讲 磁盘调度策略



§4.5 Disk Scheduling Policies



Disk I/O

■ 大容量磁盘

- 每条磁道上都有一个读 / 写磁头，可以并行读 / 写

■ 中小型磁盘设备

- 每个盘面配置一个磁头，只能串行读 / 写
- 为了读 / 写某磁道、某扇区的数据，首先让磁头伸 / 缩，寻找指定磁道，再旋转磁盘，将相应扇区定位到磁头下面

Disk Performance Parameters

- To read or write, the disk head must be positioned at the desired track and at the beginning of the desired sector.
 - Seek time(寻道时间).
 - Rotational delay or rotational latency(旋转延迟).
 - Access time(访问时间), sum of seek time and rotational delay, the time it takes to get in position to read or write.
 - Data transfer, occurs as the sector moves under the head.

Timing of a Disk I/O Transfer

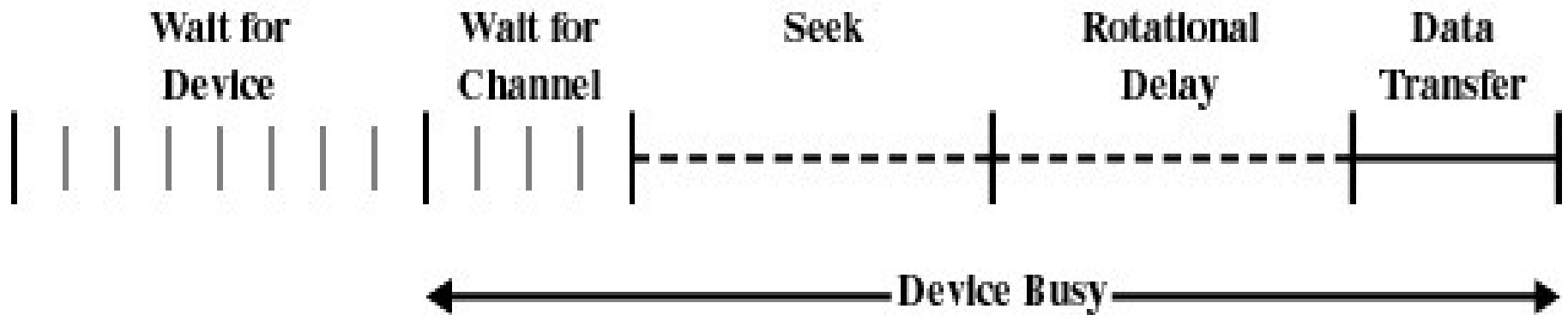


Figure 11.7 Timing of a Disk I/O Transfer

Seek Time

Time it takes to position the head at the desired track.

假设

T_s 为寻道时间,

s 为磁盘启动时间,

m 为与磁臂移动速度相关的常数,

n 为当前磁道到指定磁道的距离 (即磁臂移动所经过的磁道数)

那么, $T_s = m \times n + s$



Rotational Delay

Time it takes for the beginning of the sector to reach the head.

假设

T_r 为旋转延迟, r 为磁盘转速 (转数 / 单位时间)

那么

$$T_r = 1/(2r)$$

例:

1. 对于一个转速为 3600rpm 的硬盘而言, 其每旋转一周的时间为 16.7ms, 其平均旋转延迟为 8.3ms。
。
2. 对于一个转速为 300rpm 的软盘而言, 其每旋转一周的时间为 200ms, 其平均旋转延迟为 100ms。
。



Transfer Time

传输时间，指从磁盘上读数据或向磁盘上写数据所花费的时间

假设

T 为传输时间，

b 为传输的字节数，

N 为每个磁道上存放的字节数，

r 为磁盘转速（转数 / 单位时间）

那么

$$T = b / (r \times N)$$



Total Average Access Time

$$T_a = T_s + T_r + T$$



Rotational Positional Sensing (RPS, 旋转定位感知)

- 磁盘驱动器发出寻道命令后便释放相关的通道控制器，以便系统用它来处理其它 I/O 操作。
- 当磁臂（磁头）移到指定磁道时，磁盘驱动器驱动磁盘旋转，把指定扇区的起始位置定位到磁臂（磁头）下。
- 当指定扇区的起始位置定位到磁臂（磁头）下，磁盘驱动器重新申请通道控制器，建立到主机的通路。如果请求失败，磁盘驱动器将驱动磁盘旋转一周，再申请通道控制器。



Example

■ **假定：**一个硬盘的扇区长度为 512 个字节，磁道长度为 32 个扇区，平均寻道时间为 20ms，传输速率为 1MB/s，转速为 3600rpm。显然，如果一个长度为 128K 个字节的文件存放在该硬盘上，那么该文件将在该硬盘上占用 256 个扇区。

■ **请问：**如果系统从该硬盘上完整地读入该文件，将花费多长时间？

若文件连续地存放在硬盘的 8 个相邻的磁道上，那么系统完整地读入该文件需要花费的时间：

$$(20 + 8.3 + 16.7) + (8.3 + 16.7) \times 7 = 220\text{ms}$$



若文件随机地存放在硬盘的 256 个扇区上，那么读入该文件需要花费的时间：

$$(20 + 8.3 + 0.5) \times 256 = 7373\text{ms}$$



比较与分析

比较前面两种结果可以发现：

- 如果文件的存储方式不同，系统访问文件的效率就不同；即，文件的存储方式影响着系统访问文件的效率。
- 文件的存储方式对系统访问文件效率的影响主要在于：访问文件总的寻道时间和总的旋转延迟。



结 论

当系统访问一组磁盘扇区时，如果能够减少总的寻道时间和总的旋转延迟，那么系统的访问效率将得到提高。



Disk Scheduling Policies

- For a single disk there will be a number of I/O requests.
- If requests are selected randomly, we will get the worst possible performance.



常见的磁盘调度策略

Table 11.3 Disk Scheduling Algorithms [WIED87]

| Name | Description | Remarks |
|-----------------------------------------------|--------------------------------------------------------------|--------------------------------------------|
| Selection according to requestor | | |
| RSS | Random scheduling | For analysis and simulation |
| FIFO | First in first out | Fairest of them all |
| PRI | Priority by process | Control outside of disk queue management |
| LIFO | Last in first out | Maximize locality and resource utilization |
| Selection according to requested item: | | |
| SSTF | Shortest service time first | High utilization, small queues |
| SCAN | Back and forth over disk | Better service distribution |
| C-SCAN | One way with fast return | Lower service variability |
| N-step-SCAN | SCAN of N records at a time | Service guarantee |
| FSCAN | N-step-SCAN with N = queue size at beginning of SCAN cycle | Load-sensitive |



基于请求者属性的磁盘调度策略

- Random Scheduling Strategy (RSS, 随机调度策略)
- First In First Out Strategy (FIFO, 先进先出策略)
- Priority-Based Strategy (PRI, 基于优先级策略)
- Last In First Out Strategy (LIFO, 后进先出策略)



Random Scheduling Strategy

- 随机地满足各个进程对磁盘的读写请求
- 该策略性能最差。可用作评估其它磁盘调度策略



First-in, First-out (FIFO)

- **Process request sequentially.**
- **Fair to all processes.**
- **Approaches random scheduling in performance if there are many processes.**



Priority-Based Strategy

- Goal is not to optimize disk use but to meet other objectives.
- Short batch jobs may have higher priority.
- Provide good interactive response time.



Last-in, First-out

- Good for transaction processing systems.
- The device is given to the most recent user so there should be little arm movement.
- Possibility of starvation since a job may never regain the head of the line.



基于被请求扇区位置的磁盘调度策略

- SSTF 算法
- SCAN 算法
- C-SCAN 算法
- N-step-SCAN 算法
- FSCAN 算法



Shortest Service Time First (最短寻道时间优先)

- Select the disk I/O request that requires the least movement of the disk arm from its current position.
- Always choose the minimum Seek time.



Shortest Service Time First

- 保证在满足两次磁盘读写请求之间磁头移动距离最小，但并不保证在满足一组磁盘读写请求时总的磁头移动距离最小
- 该算法的性能比 FIFO 策略的性能好
- 当不断有新的磁盘读写请求被提交时，该算法可能会导致饿死现象。



SCAN （扫描算法、电梯调度法）

- **Arm moves in one direction only, satisfying all outstanding requests until it reaches the last track in that direction.**
- **Direction is reversed.**



SCAN

- 该算法的性能与 SSTF 算法的性能非常接近；
但与 SSTF 算法相比，该算法避免了饿死现象
- 如果当磁头刚刚从靠近磁盘某边的某个磁道上移过并向磁盘另一边移动时，恰好有一个进程请求访问该磁道，那么该请求将被严重推迟。



C-SCAN （单向扫描算法）

- Restricts scanning to one direction only.
- When the last track has been visited in one direction, the arm is returned to the opposite end of the disk and the scan begins again.



C-SCAN

果当磁头刚刚从靠近磁盘某边的某个磁道上移过并向磁盘另一边移动时，恰好有一个进程请求访问该磁道，那么该请求将被延迟：

- 若使用 SCAN 算法： $2T$
- 使用 C-SCAN 算法： $T+S_{\max}$

中， T 为磁头从磁盘的一边扫描到另一边所花费的时间， S_{\max} 为磁头从磁盘的一边直接返回另一边所花费的时间；通常， $T \gg S_{\max}$



N-step-SCAN

- 基于各个磁盘读写请求的提交次序把所有磁盘读写请求组织在若干队列中；其中，每个队列包含 N 个磁盘读写请求；
- 如果存在一个长度小于 N 的磁盘读写请求队列，那么当一个新的磁盘读写请求到达时系统将把它加入到该队列中；



N-step-SCAN (continue)

- 如果不存在一个长度小于 N 的磁盘读写请求队列，那么当一个新的磁盘读写请求到达时系统将其建立一个新的队列；
- 系统将使用 FIFO 策略依次处理每个磁盘读写请求队列；
- 系统将使用 SCAN 算法处理每个队列中的磁盘读写请求。
- 该算法避免了“磁臂粘着 (Arm Stickiness)”现象。



FSCAN

- Two queues, one queue is empty for new request
- 当前所有磁盘读写请求被组织在第一个队列中；系统将使用 SCAN 算法处理该队列中的磁盘读写请求；
- 在系统处理第一个队列中的磁盘读写请求时，如果有新的磁盘读写请求到达，那么新的磁盘读写请求将被组织在第二个队列中；
- 一旦第一个队列中的磁盘读写请求被处理完毕，系统便转向第二个队列以便使用 SCAN 算法处理其中的磁盘读写请求；
- 上述第二步和第三步将被系统重复使用。



Example

- **假定：**当前有 9 个磁盘读写请求；这 9 个磁盘读写请求要访问的磁道号按照各个磁盘读写请求到达的次序依次为：
55 、 58 、 39 、 18 、 90 、 160 、 150 、 38 、 184 。
此外，磁头当前位于 100 号磁道上。如果系统使用 SCAN 算法或 C-SCAN 算法，那么我们还假定磁头当前的移动方向为磁道号增长的方向。
- **请问：**如果系统分别使用 FIFO 策略、 SSTF 算法、 SCAN 算法、 C-SCAN 算法调度磁盘，那么系统处

FIFO

| 被访问的磁道号 | 移动距离(磁道数) |
|---------|-----------|
| 55 | 45 |
| 58 | 3 |
| 39 | 19 |
| 18 | 21 |
| 90 | 72 |
| 160 | 70 |
| 150 | 10 |
| 38 | 112 |
| 184 | 146 |
| 总的寻道长度 | 498 |
| 平均寻道长度 | 55.3 |



SSTF

被访问的磁道号

移动距离(磁道数)

90

10

58

32

55

3

39

16

38

1

18

20

150

132

160

10

184

24

总的寻道长度

248

平均寻道长度

27.6



电子科技大学
University of Electronic Science and Technology of China

SCAN

被访问的磁道号

移动距离(磁道数)

150

50

160

10

184

24

90

94

58

32

55

3

39

16

38

1

18

20

总的寻道长度

250

平均寻道长度

27.8



电子科技大学
University of Electronic Science and Technology of China

C-SCAN

被访问的磁道号

移动距离(磁道数)

150

50

160

10

184

24

18

166

38

20

39

1

55

16

58

3

90

32

总的寻道长度

322

平均寻道长度

35.8



电子科技大学
University of Electronic Science and Technology of China