DATA STORAGE TECHNOLOGIES & NETWORKS (CS C446, CS F446 & IS C446)

LECTURE 16– STORAGE

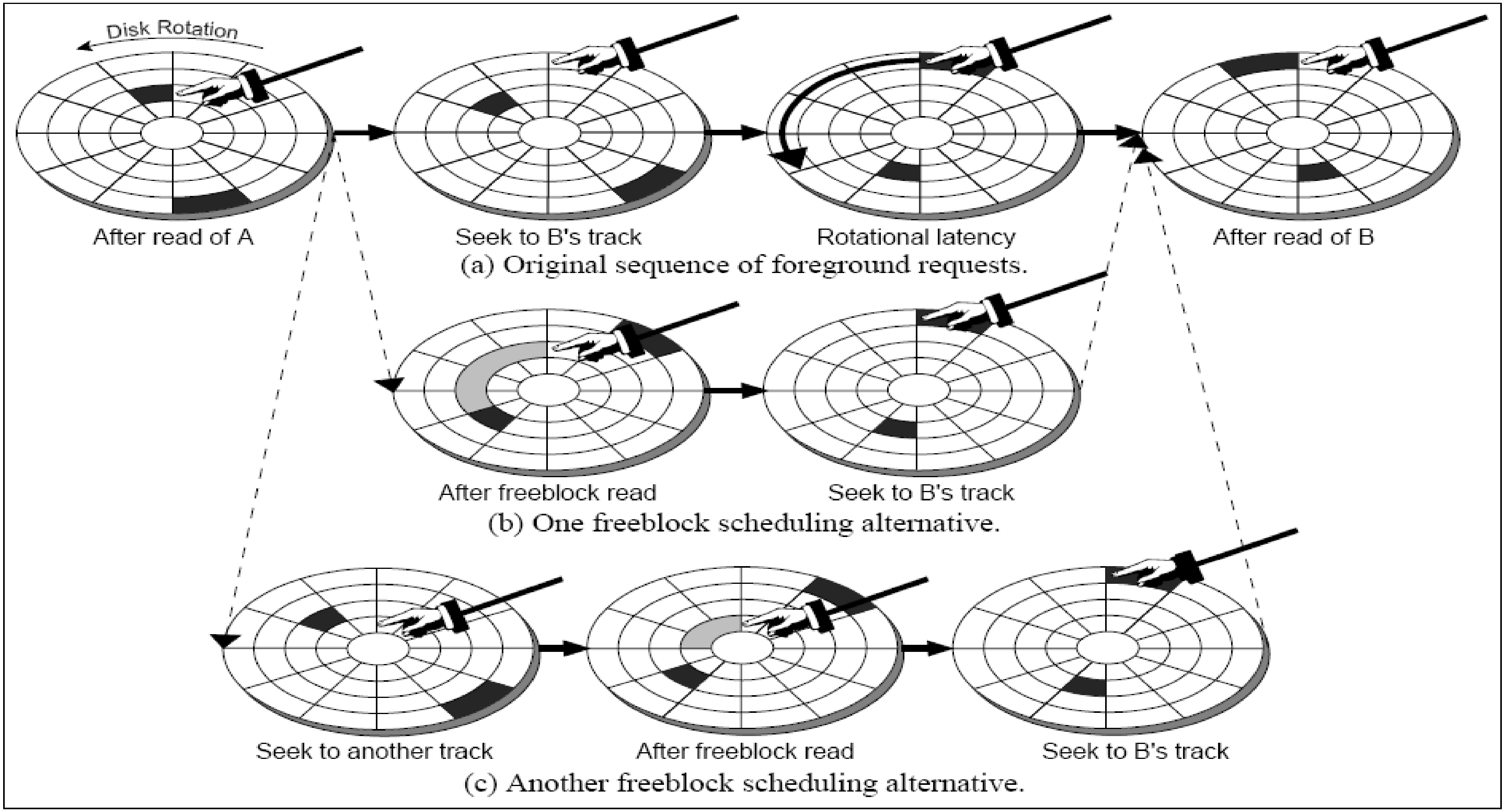
- ELEVATOR_LBN
- CYCLE_LBN
- SSTF_LBN
- ELEVATOR_CYL
- CYCLE_CYL
- SSTF_CYL
- SPTF_OPT
- SPCTF_OPT
- SATF_OPT
- WPTF_OPT
- WPCTF_OPT
- WATF_OPT

- LBN → Logical Block Number
- CYL → Cylinder
- OPT → Optimal
- SSTF → Shortest Seek Time First
- SPTF → Shortest Positioning Time First
- SPCTF → Shortest Positioning and Caching Time First
- SATF → Shortest Access Time First
- Weighted Positioning Time First
- Weighted Positioning and Caching Time First
- Weighted Access Time First

- ASPTF_OPT
- ASPCTF_OPT
- ASATF_OPT
- VSCAN_LBN
- VSCAN_CYL
- PRI_VSCAN_LBN
- PRI_ASPTF_OPT
- PRI_ASPCTF_OPT
- SDF_APPROX
- SDF_EXACT
- SPTF_ROT_OPT
- SPTF_ROT_WEIGHT
- SPTF_SEEK_WEIGHT

- SATFUF → Shortest Access Time First with Urgent Forcing
- ASPTF → Aged Shortest Positioning Time First

- ZSPTF → Zone based Shortest Positioning Time First
    - Finish request from same zone before moving to next zone
- Zone based arrival-Time-constrained Shortest Positioning Time First (ZTSPTF)
    - improves the fairness of ZSPTF [similar to FCFS] by only servicing requests that arrive before ZTSPTF begins scheduling requests in a zone



Free Block Scheduling

Taken from “Towards Higher Disk Head Utilization: Extracting Free Bandwidth from Busy Disk Drives”, Lumb et. al.

Scheduling – Seek Time

- Basic Model:
  - Seek Time = c*s
    - where s is the seek distance and c is a constant
  - Flaw:
    - There is a constant overhead per head movement – settling time
- More robust Model:
  - Seek Time = $c_0 + c_1 * s$
    - where s is the seek distance, c_0 is a constant overhead and c_1 is a constant factor.
    - e.g. Seek Time (for a specific disk) = 2 + 0.00009 * s (in ms)

Scheduling – Seek Time

- Practical Model:
  - Non-linear relation between seek-time and seek-distance:
    - Head has to accelerate and decelerate
    - Derive a simple seek time equation!

Scheduling – Seek Time & Rotational Latency

- Scheduling at the OS level:
  - Hard to incorporate rotational latency due to delay in passing information
- Scheduling at the disk (controller) level:
  - Possible to incorporate rotational latency in addition to seek time
    - Shortest Positioning Time First (SPTF)
      - Problems similar to SSTF
  - Algorithms are more complicated.
  - Queue size is often small

Scheduling – Pragmatics

- Some practical parameters:
  - Disks are usually lightly loaded
  - Locality is an important factor
    - Probability that two contiguous requests are for the same cylinder: 2/3
    - It makes sense to store / allocate files to enable sequential access
    - Track to next track requires constant seek time:
      - Sectors on adjacent tracks are skewed so that access can continue w/o rotational latency
    - Head switching may also require time:
      - Sectors on (the same cylinder in) adjacent surfaces may also be skewed

Scheduling – Pragmatics

- Some practical parameters [continued]:
  - Most modern disks include cache
    - Referred to as on-disk cache or disk buffer
      - [Not to be confused with buffer cache which is part of the OS]
    - Minimum buffer required anyway for one logical block – a few sectors
  - Uses of disk buffer:
    - Caching
      - retaining previously accessed blocks
    - Prefetching (or read-ahead)
      - Reading the (rest of the) track even if not requested
    - Fast writes
      - Controller confirms write operations before writing onto the disk (i.e. once data is received on cache)

Scheduling – Pragmatics

- Some practical parameters [continued]:
  - Typical I/O workloads:
    - QoS guarantee vs. best effort
    - Typical file access vs. multimedia content
    - Sequential access vs. random access
  - Typical Filesystem storage model
    - Files in the same directory
    - Average size of files
    - File data vs. metadata (inode information)