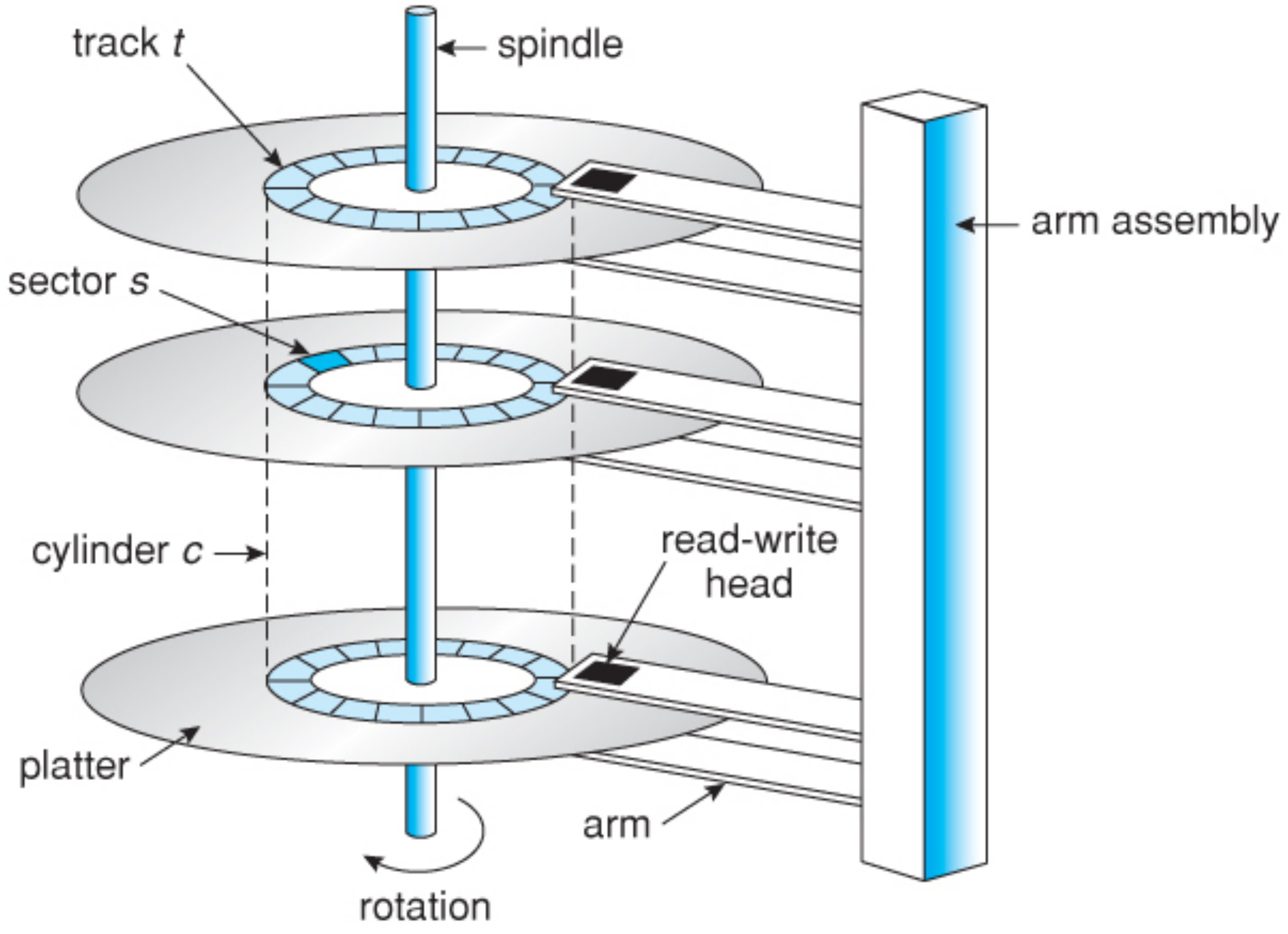
Hard Disk Drive

NDBI007: Practical class 1

Disk Structure

- Read-write *head*
- The surface of platters is divided into *tracks*
- The set of all tracks with the same diameter form a *cylinder*
- Track is divided into *sectors*



Zone bit recording

- The tracks closest to the outer edge contain more sectors per track
- The data transfer speed over the outside cylinders is higher since the angular speed is constant regardless which track is being read

source: https://www.cs.uic.edu/~jbell/CourseNotes/OperatingSystems/10_MassStorage.html

Important Terms

Rotational latency r

- Time needed to come to the right track
- Single rotation is equal to $2 \bullet r$

$$r = \frac{1}{rotational\_speed}$$

Seek time s

- Time needed to move read-write head from one track to another
- *Average seek time* from one random track (cylinder) to any other is the most common seek time metric
- *Track-to-track seek time* is the amount of time that is required to seek between adjacent tracks
- *Full-track seek time* (full stroke) is the time needed to seek data from the first track to the last

Block transfer time btt

- Time needed to read block to memory (buffer)

Track Capacity (TC)

- Track capacity can be based on different characteristics*
- The size of a sector is constant
- As the number of sector differ (i.e., zone bit recording), we *expect the estimated track capacity to differ*

User cylinders

$$TC = \frac{capacity}{data\_heads \bullet user\_cylinders}$$

$$TC = \frac{75 \bullet 10^9}{10 \bullet 27724}$$

$$TC \approx 0.28\ MB$$

IBM Deskstar HDD

Capacity	75 GB
Data heads	10
User cylinders	27,724

Sectors per track (SPT)**

$$TC = SPT \bullet sector\_size$$

* All used characteristics can be found in the data sheet for the IBM Deskstar HDD

** SPT is not provided for the IBM Deskstar HDD as the number of sectors per track is not constant

Exercise 1.1

Estimate track capacity based on *latency* (r) and *media transfer rate* (MTR)*

- Media transfer rate uses bits not bytes as a unit (1B = 8b)
- Use MTR (max) measured at the outer edge of the HDD
- Use $2 \bullet r$ since we need the amount of time required to full rotation of plates
- Transfer speed on outer edge is maximal, hence the result is the upper bound

IBM Deskstar HDD	
Media transfer rate	448 Mb/s
Latency	4.17 ms

$$MTR = \frac{TC}{2 \bullet r}$$

* All used characteristics can be found in the data sheet for the IBM Deskstar HDD

Exercise 1.2

Estimate track capacity based on *sustained data rate* (SDR)*

- SDR is computed as the average transfer speed
  - Hence, we must consider:
    - The time taken to get heads to the right track
    - The time taken to switch tracks in a single cylinder, i.e., $head\_switch\_time$
      - The value is not presented in data sheet, consider it to be $\pm 1\ ms$

IBM Deskstar HDD

Data heads	10
head_switch_time	1 ms
track_to_track_time	1.2 ms
Sustained data rate	37 MB/s

- To get SDR, we have to:
  - Move heads to a cylinder
  - Read the whole cylinder, one track to another. Only one head can be read at a certain time
  - Move heads to another cylinder, i.e., $track\_to\_track\_time$

$$SDR = \frac{data\_heads \bullet TC}{2 \bullet r \bullet data\_heads + (data\_heads - 1) \bullet head\_switch\_time + track\_to\_track\_time}$$

* All used characteristics can be found in the data sheet for the IBM Deskstar HDD

Example 1.3: Reading Fully Fragmented File From the HDD

- Consider fully fragmented file, i.e., the blocks are not adjacent
  - We assume uniformly distributed blocks
- File size is 1 GB
- Block size is 4 kB

- The process of reading fragmented data looks like this:

1. Move heads to the right cylinder
2. Read a sector
3. Continue with 1. until the whole file is read

Example 1.3 (Continued)

First, we need to know how many blocks form the 1 GB file, i.e., the *block count* (BC)

$$BC = \frac{file\_size}{block\_size} = \frac{1 \bullet 10^9}{4 \bullet 10^3} = 250{,}000$$

We compute how long does it take to transfer a single block, i.e., we compute the *block transfer time* (btt)*

$$btt = \frac{2 \bullet r}{TC} \bullet block\_size = \frac{2 \bullet 4.17}{0.3} \bullet 0.004$$

$$btt = 0.11\ ms$$

IBM Deskstar HDD	
Tack capacity	0.3 MB
Latency	4.17 ms
Block size	0.004 MB
Average seek time	8.5 ms

Finally, we combine all together

- $read\_time = BC \bullet (s + r + btt)$
- $read\_time = 250{,}000 \bullet (8.5 + 4.17 + 0.11)$
- $read\_time \approx 3{,}195\ s \approx 53\ m$

* It is important to realize that we use TC that is somewhere between the estimates we got before

Exercise 1.4

Solve example 1.3 having track capacity (TC) estimate based on latency and media transfer rate (MTR; see exercise 1.1)

$$btt = \frac{2 \bullet r}{TC} \bullet block\_size$$

- You can also use MTR to compute btt directly
  - Reminder: Media transfer rate uses bits not bytes as a unit (1B = 8b)

$$btt = \frac{block\_size}{MTR}$$

- Try it yourself: Usage of MTR and usage of TC computed from MTR have the same result

IBM Deskstar HDD

Block size	0.004 MB
Media transfer rate	448 Mb/s
Latency	4.17 ms

Example 1.5: Reading Sequential Data From the HDD

- In this case, blocks are adjacent
- Once again, file size is 1 GB and block size is 4 kB

- We can use sustained transfer rate (STR) since it equals to $MTR + head\_switch\_time + track\_to\_track\_time$
  - But let's assume that the STR is unknown to us

- First, we need to determine *number of tracks* n_T that the file occupies

$$n_T = \frac{file\_size}{TC} = \frac{1 \bullet 10^9}{0.3 \bullet 10^6} = 3333.3$$

- We compute *number of cylinders* n_C

$$n_C = \frac{n_T}{data\_heads} = \frac{3333.3}{10} = 333.3$$

IBM Deskstar HDD

Track capacity	0.3 MB
Data heads	10

Example 1.5 (Continued)

- Now, we can compute the *read time as the summation of* several times:
  - *Move heads* to the initial cylinder $(s + r)$
  - *Read blocks* $(2 \bullet r \bullet n_T)$
  - *Number of head switches* $(n_C \bullet (data\_heads - 1) \bullet head\_switch\_time)$
    - I.e., for each cylinder we have to do $data\_heads - 1$ switches
  - *Time to move between adjacent cylinders* $(n_C \bullet track\_to\_track\_time)$
    - Note that we assume the best possible positioning for block

IBM Deskstar HDD	
Average seek time	8.5 ms
Latency	4.17 ms
Data heads	10
Head switch time	1 ms
Track-to-track	1.2 ms

- $t_{read} = (s + r) + (2 \bullet r \bullet n_T) + (n_C \bullet (data\_heads - 1) \bullet head\_switch\_time) + (n_C \bullet track\_to\_track\_time)$
- $t_{read} = (8.5 + 4.17) + (2 \bullet 4.17 \bullet 3333.3) + (333.3 \bullet (10 - 9) \bullet 1) + (333.3 \bullet 1.2)$
- $t_{read} = 31\ s$

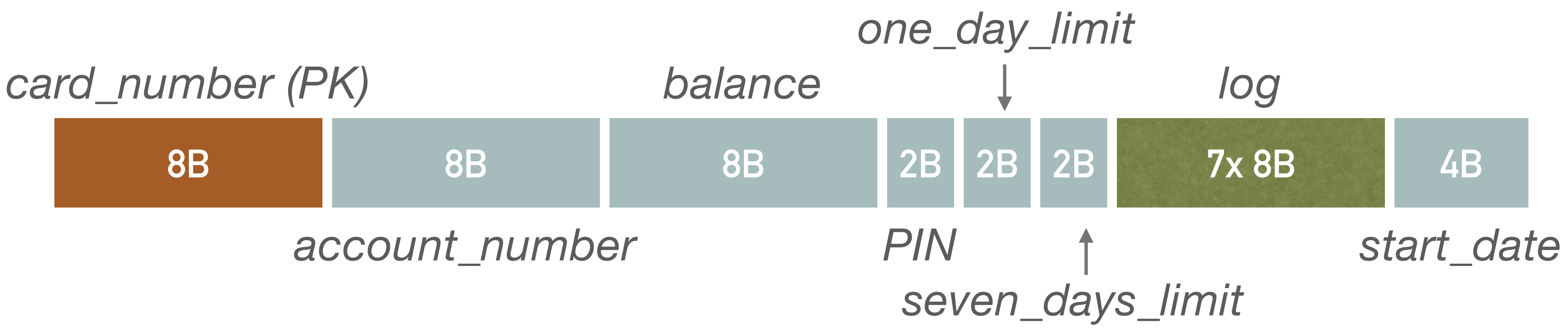
Example 1.6: Bank Withdrawals

Design a record structure for a credit card system managing 5,000,000 cards

- The system should allow a defined amount of money to be withdrawn when a card is inserted
- The withdrawal should identify the relevant DB record, i.e., the account associated with that card, and check the daily and weekly limits on withdrawals

- The log records withdrawals for the last 7 days and the start date is the information when the first recorded withdrawal was made
- To test the limit for the last 7 days, we simply check what date is the last log entry (from the start date)

Record structure:



Example 1.6 (Continued)

Estimate time required for a single withdrawal

- The withdrawal needs to find the record and write it to the log
- Consider a situation where we have an *index-sequential file*, i.e., data stored sequentially with an index to a primary key built over the primary file

First, determine how many records fits the size of one block

- We define block size 4 kB, pointer size 4 B (needed to calculate index blocking factor)
- Record size $R = 128B$ (rounded to nearest power of 2)

$$b = \frac{B}{R}$$

$$b = \frac{4 \bullet 2^{10}}{128} = 32$$

Example 1.6 (Continued)

Second, determine *blocking factor* for the index R_I

- We need $N = 5{,}000{,}000 \div 32 = 156{,}250$ blocks to store records of all the accounts
  - The number of blocks is also the number of index sheets
- We need to know how many index records (key-pointer pairs) can fit in the index block, i.e., the *blocking factor* (b) for the index R_I

$$R_I = 8 + 4 = 12$$

$$b = \frac{B}{R_I} = \frac{4 \bullet 2^{10}}{12} = 341$$

Example 1.6 (Continued)

Third, the height of the tree (h) is calculated

$$h = \lceil \log_{R_I} N \rceil = \lceil \log_{341} 156{,}250 \rceil = 3$$

- The root of the index tree is always stored in memory (it is 1 page)
  - Therefore, 3 disk accesses are needed to read the record (2 index levels and 1 data file blocks)
  - However, in this particular case, tree-level 2 has only 2 pages
    - 2 pages can address $2 \bullet 341 \bullet 341$ pages, which is more pages than the primary file has
  - Hence, we can keep the second level of the index straight in memory, and then we only need to access the disk twice

Finally, the time it takes to load the record is*

$$T = 2 \bullet (s + r + btt) + 2 \bullet r + btt$$

$$T = 2 \bullet (8.5 + 4.17 + 0.11) + 2 \bullet 4.17 + 0.11$$

$$T = 34\ ms$$

- If a record is processed in one rotation of the disk, then after the time of one rotation ($2 \bullet r$) the modified data can be written to disk

* Twice because we go once to the index level 3 and once to the data file

Example 1.7: Bank Transactions per Day

- In 2007, the number of all transactions in the Czech Republic was approximately 800,000 per day
- Can our system handle such a number, assuming that we handle a quarter of all transactions in the country?
  - Assume that the load is not evenly distributed over the day and that half of all transactions are made at peak times
  - That is, 100,000 requests per hour go to our system

- In other words, how many request are we able to serve per hour?

$$n_T = \frac{60 \bullet 60 \bullet 1{,}000}{T} = \frac{60 \bullet 60 \bullet 1{,}000}{34} = 105{,}882$$

- $n_T > 100{,}000$, hence the system handles the workload