NDBI007: PRACTICAL CLASS 1

HARD DISK DRIVE

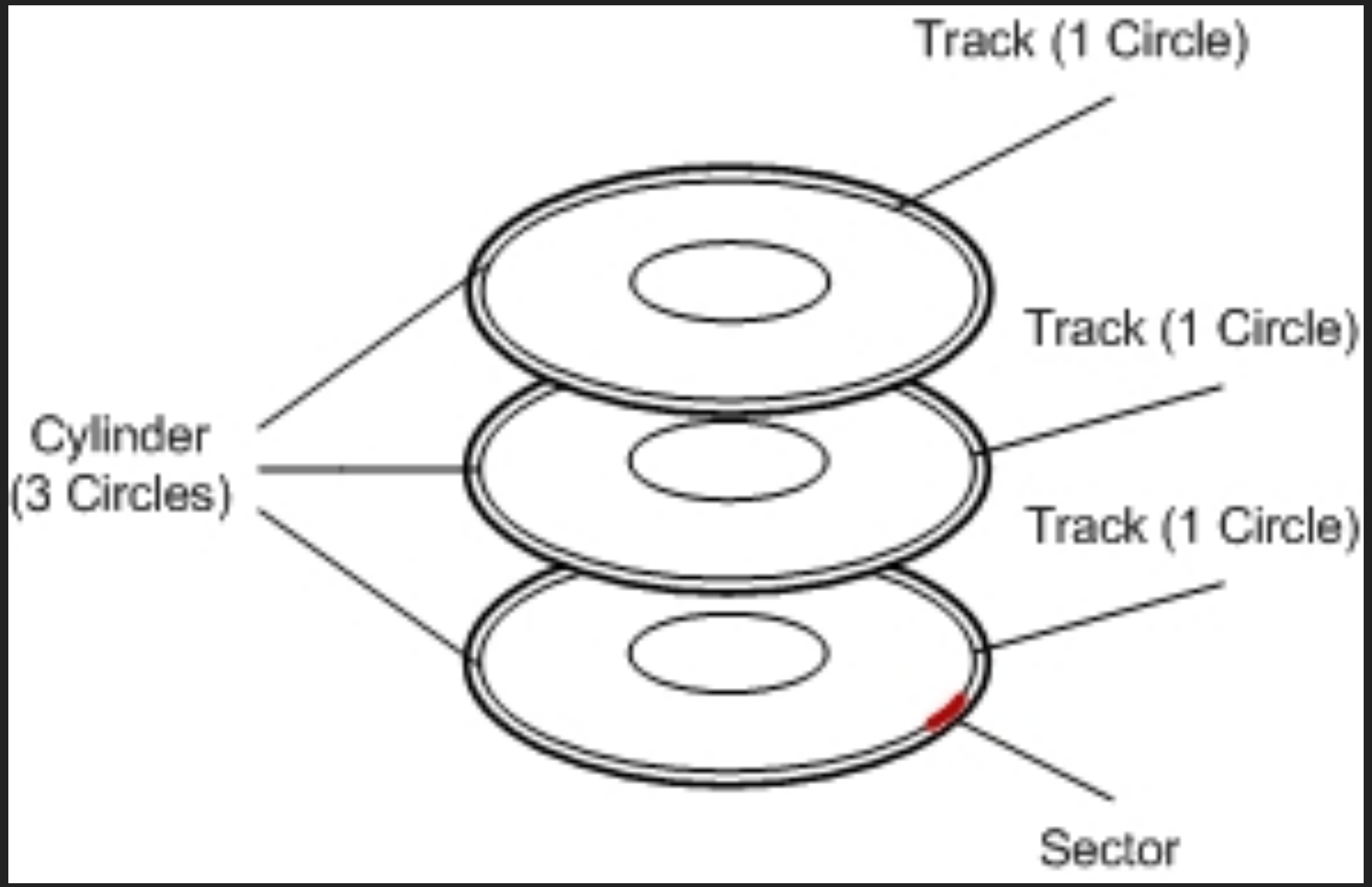
Inspired by NDBI007 practical class materials created by Petr Škoda; Tutor: Pavel Koupil; October 13th 2021

IMPORTANT TERMS

- Latency r
    - $r = \frac{1}{rotational\_speed}$
    - Single rotation is equal to $2 \bullet r$
- Seek Time s
    - 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

DISK STRUCTURE

- Disk structure
  - The surface of platters is divided into `tracks`
  - Track is divided into `sectors`
  - The set of all tracks with the same diameter form a `cylinder`
- Zoned 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



https://www.snia.org/education/storage_networking_primer/stor_devices/data_structure

TRACK CAPACITY (TC)

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

- User cylinders

$$TC = \frac{capacity}{data\_heads \bullet user\_cylinders} = \frac{75 \bullet 10^9}{10 \bullet 27724} \approx 0.28\ MB$$

- 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 sector per track is not constant

EXERCISE 1: ESTIMATE TRACK CAPACITY BASED ON R AND MTR

- Estimate track capacity based on `latency` (r) and `media transfer rate` (MTR)
  - Media transfer rate uses bits not bytes as unit (1B = 8b)
  - We use MTR (max) measured at the outer edge of the HDD
  - We use $2 \bullet r$ since we need the amount of time required to full rotation of plates
  - Transfer speed on outer edge is maximal, therefore the result is the upper bound

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

EXERCISE 2: ESTIMATE TRACK CAPACITY BASED ON SDR

- Estimate track capacity based on `sustained data rate` (SDR)
  - SDR is computed as the average transfer speed. Therefore, 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., $head\_switch\_time$ (value is not presented in data sheets, consider it to be $\pm 1\ ms$)

- 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., $track\_to\_track\_time$

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

EXAMPLE 1: READING FULLY FRAGMENTED FILE FROM THE HDD (SOLVED)

- 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:
    - Move heads to the right cylinder
    - Read a sector
    - Continue with 1 until the whole file is read

EXAMPLE 1: READING FULLY FRAGMENTED FILE FROM THE HDD (SOLVED)

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

$$BC = \frac{1 \bullet 10^9}{4 \bullet 10^3} = 250000$$

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

$$btt = \frac{2 \bullet r}{TC} \bullet block\_size = \frac{2 \bullet 4.17}{0.3} \bullet 0.004 = 0.11 \ ms$$

- Finally, we combine all together

$$read\_time = BC \bullet (s + r + btt) = 250000 \bullet (8.5 + 4.17 + 0.11) \approx 3195 \ s \approx 53 \ m$$

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

EXERCISE 3: READING FULLY FRAGMENTED FILE

- Solve previous example having TC estimate based on latency and media transfer rate MTR (see exercise 1)
  - You can also use MTR to compute btt directly

$$btt = \frac{block\_size}{\frac{MTR}{8}}$$

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

EXAMPLE 2: READING SEQUENTIAL DATA FROM THE HDD (SOLVED)

- 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 find out how many tracks the file occupies, i.e., number of tracks n_T

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

- We compute number of cylinders n_C

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

EXAMPLE 2: READING SEQUENTIAL DATA FROM THE HDD (SOLVED)

- 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. i.e., for each cylinder we have to do $data\_heads - 1$ switches, i.e., $(n_C \bullet (data\_heads - 1) \bullet head\_switch\_time)$
  - Time to move between adjacent cylinders, as we assume the best possible positioning for block, i.e., $(n_C \bullet track\_to\_track\_time)$

$$t_{read} = (s + r) + (2 \bullet r \bullet n_T) + (n_C \bullet (data\_heads - 1) \bullet head\_switch\_time) + (n_C \bullet track\_to\_track\_time)$$

$$t_{read} = (8.5 + 4.17) + (2 \bullet 4.17 \bullet 3333.3) + (333.3 \bullet (10 - 9) \bullet 1) + (333.3 \bullet 1.2) = 31 \ s$$

EXAMPLE 3: BANK WITHDRAWALS – RECORD STRUCTURE (SOLVED)

- 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:
  - card_number (8B), i.e., primary identifier (key)
  - account_number (8B)
  - balance (8B)
  - PIN (2B)
  - one_day_limit (2B)
  - seven_day_limit (2B)
  - log (7x8B)
  - start_date (4B)

EXAMPLE 4: BANK – TIME REQUIRED FOR SINGLE WITHDRAWAL (SOLVED)

- 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 sorted sequentially with an index to a primary key built over this primary file

- First, determine `how many records fits the size of one block`, i.e., $B = 4\ kB$
  - We define block size 4 kB, pointer size 4 B (needed to calculate index blocking factor)
  - Record size $R = 128\ B$ (rounded to the nearest power of 2)

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

EXAMPLE 4: BANK – TIME REQUIRED FOR SINGLE WITHDRAWAL (SOLVED)

- 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 for the index R_I

$R_I = 8\ B + 4\ B$ (we have 32 bit pointers)

$B = 4 \bullet 2^{10}\ B$

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

EXAMPLE 4: BANK – TIME REQUIRED FOR SINGLE WITHDRAWAL (SOLVED)

- Third, the height of the tree 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 block)
  - However, in the situation we are in our tree-level 2 has only 2 pages
    - 2 pages can address 2 * 341 * 341 pages, which is more pages than the primary file has
  - In such a situation, we can keep the second level of the index, i.e., 2 pages, straight in memory, and then we only need to touch the disk twice

- Then the time it takes to load the record*

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

$T = 2 \bullet (8.5 + 4.17 + 0.11) + 2 \bullet 4.17 + 0.11 = 34\ ms$

- If I can process a record in one rotation of the disk, then after the time of one rotation ($2r$) I can write the modified data back to disk

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

EXAMPLE 5: BANK – TRANSACTIONS PER DAY (SOLVED)

- In 2007, the number of all transactions in the Czech Republic per day was about 800,000
- 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

- That is, how many requests are we able to serve per hour?

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

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