

Datová úložiště

Geometrie HDD, RAID

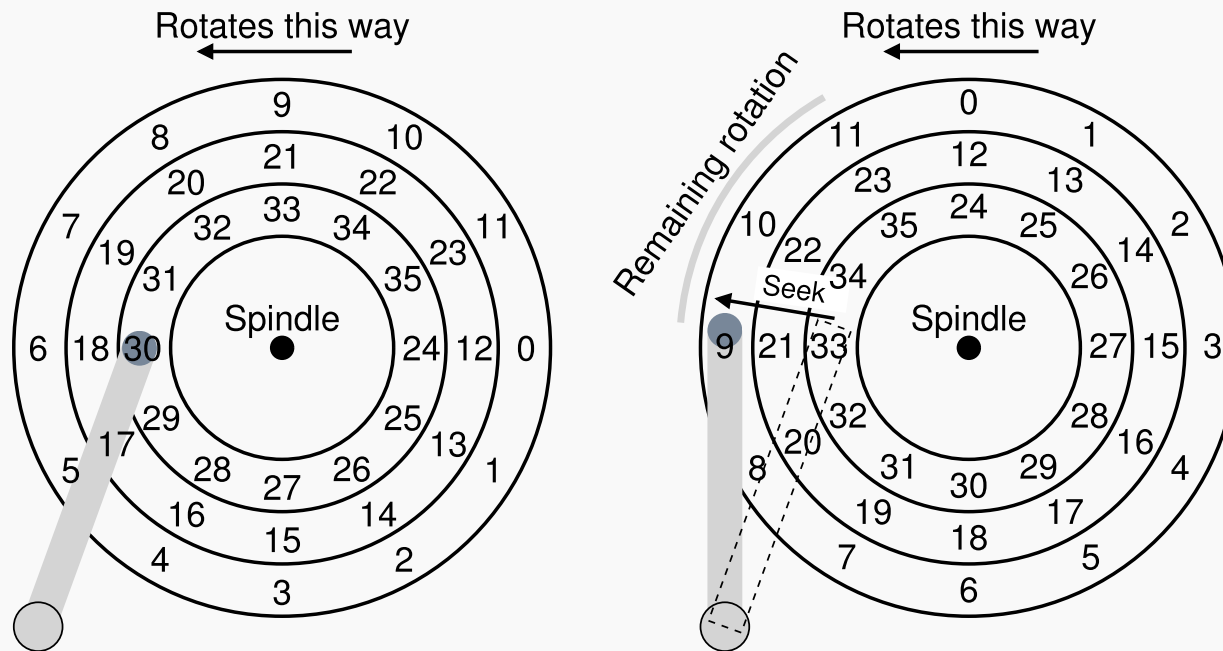
Milan Radojčić

SSPŠaG – Operační systémy

19. 10. 2026



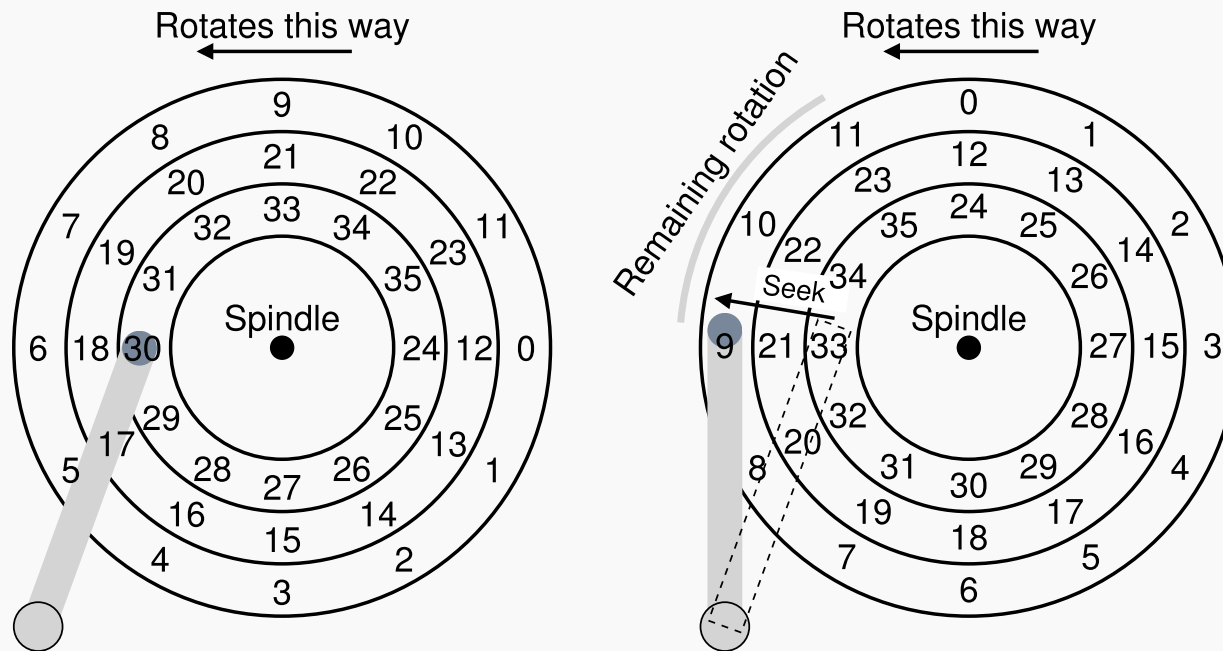
Geometrie HDD



- HDD se skládá z několika **ploten**, na jejich povrchu je speciální materiál do kterého se pomocí směru magnetizace ukládá informace.
- Ke čtení se používají speciální **hlavičky**, ty se můžou pohybovat směrem od středu.
- Celý disk se točí a proto se mohou hlavičky dostat všude.



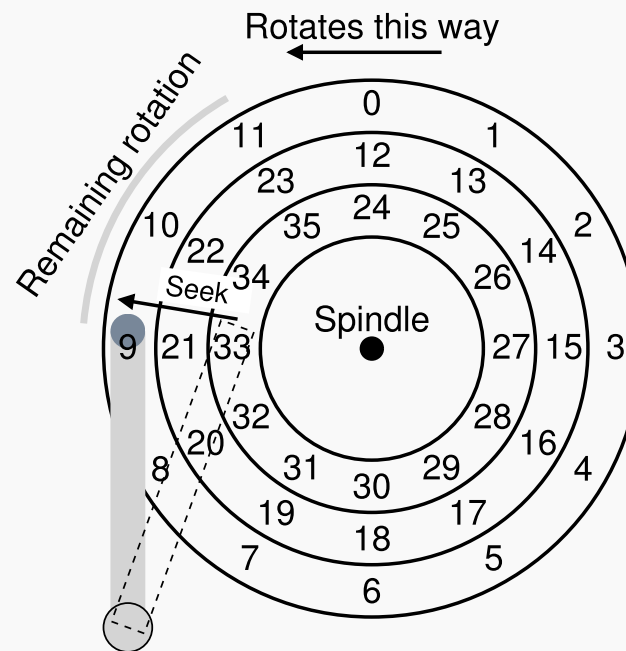
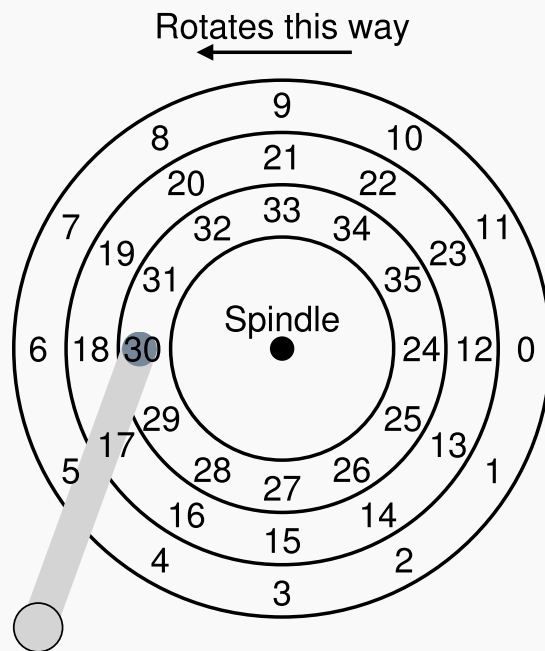
Geometrie HDD

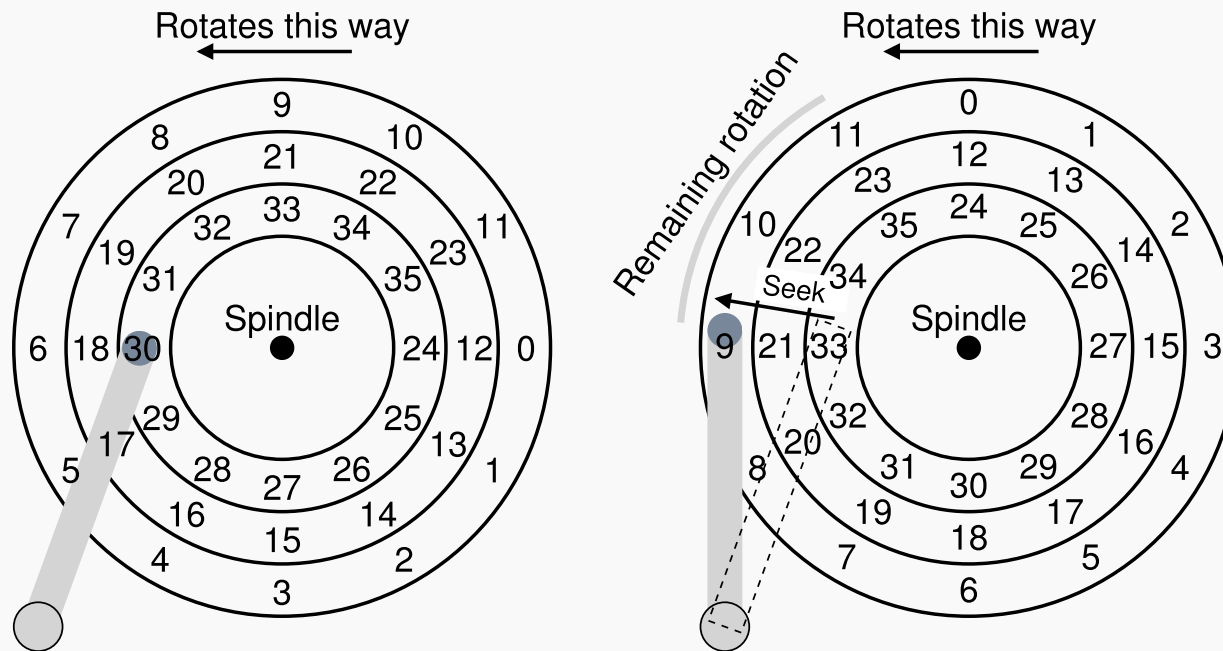


- Jednomu kruhu na povrchu plotny říkáme **stopa** (*track*).
 - Protože můžeme mít více ploten, tak všem stopám na všech plotnách se stejným poloměrem říkáme **cylindr**.
- Nejmenší adresovatelná jednotka se nazývá **sektor** nebo **blok** a má (většinou) 4KB.



Na čem závisí rychlost zápisu?





1. Doba vystavení (*seek time*)

- Čtecí hlavička se musí dostat nad správnou stopu.

2. Rotační zpoždění (*rotational delay*)

- Čtecí hlavička musí počkat, než se disk otočí a dostane se nad správný sektor.



Vsuvka: Testování rychlosti úložiště

```
1  fio --name=read_test \  
2      --filename=/dev/sdX \  
3      --rw=read \  
4      --bs=4k \  
5      --size=1G \  
6      --numjobs=1 \  
7      --iodepth=16 \  
8      --direct=1 \  
9      --runtime=60 \  
10     --time_based \  
11     --group_reporting
```

- Provádí sekvenční čtení, pro náhodné čtení lze použít `--rw=randread`.



- Mějme HDD s těmito parametry: 15000 RPM, průměrná doba vystavení 4ms, rychlost přenosu: 125 MB/s.
- Jak rychlý bude tento HDD když budeme uvažovat náhodný přístup k sektorům?
- Celkový čas na provedení I/O můžeme rozepsat jako

$$T_{I/O} = T_{\text{seek}} + T_{\text{rotation}} + T_{\text{transfer}}$$

- Potřebujeme získat z RPM čas na jednu rotaci v ms:

$$\frac{1 \cancel{\text{min.}}}{15000 \text{ rot.}} \cdot \frac{60 \cancel{\text{sek.}}}{1 \cancel{\text{min.}}} \cdot \frac{1000 \text{ ms}}{1 \cancel{\text{sek.}}} = 4 \frac{\text{ms}}{\text{rot.}}$$

- Máme tedy $T_{\text{seek}} = 4 \text{ ms}$, $T_{\text{rotation}} = 2 \text{ ms}$.



- Mějme HDD s těmito parametry: 15000 RPM, průměrná doba vystavení 4ms, rychlost přenosu: 125 MB/s.
- Jak rychlý bude tento HDD když budeme uvažovat náhodný přístup k sektorům?
- Čas na přenos můžeme získat jako

$$T_{\text{transfer}} = \frac{\text{velikost přenosu}}{\text{rychlost přenosu}}.$$

- Pro náš případ je to $T_{\text{transfer}} = \frac{4\text{KB}}{125\text{MB/s}} \approx 30\mu\text{s}$ – zanedbáme.



- Mějme HDD s těmito parametry: 15000 RPM, průměrná doba vystavení 4ms, rychlost přenosu: 125 MB/s.
- Jak rychlý bude tento HDD když budeme uvažovat náhodný přístup k sektorům?
- Celkem tedy $T_{I/O} = 6\text{ms}$, propustnost získáme jako

$$R_{I/O} = \frac{\text{velikost přenosu}}{T_{I/O}}.$$

- Tedy pro náš případ

$$R_{I/O} = \frac{4\text{KB}}{6\text{ms}} = \frac{1\text{MB}}{1024\text{KB}} \cdot \frac{4\text{KB}}{6\text{ms}} \cdot \frac{1000\text{ms}}{1\text{s}} \approx 0,65\text{MB/s!}$$



- Mějme HDD s těmito parametry: 15000 RPM, průměrná doba vystavení 4ms, rychlost přenosu: 125 MB/s.
- Jak se změní situace když budeme uvažovat sekvenční přístup např. ke 100 MB?
- T_{seek} a T_{rotation} budou stejné, potřebujeme přepočítat T_{transfer} :

$$T_{\text{transfer}} = \frac{100\text{MB}}{125\text{MB/s}} = 800\text{ms}.$$

- Tedy $T_{\text{I/O}} = 806\text{ms}$ a $R_{\text{I/O}} = \frac{100\text{MB}}{806\text{ms}} \cdot \frac{1000\text{ms}}{1\text{s}} \approx 124.$



- Zkratka pro Redundant Array of Inexpensive Disks.^{<1>}
- Jsou to různé techniky, jak spojit více *fyzických disků* do jednoho **logického**.
- Co můžou být různé plusy a mínusy? Bude nás zajímat
 - **kapacita,**
 - **spolehlivost a**
 - **výkon.**

^{<1>}A Case for Redundant Arrays of Inexpensive Disks (RAID) – Patterson, Gibson, Katz 1988



RAID 0

- RAIDu 0 se říká **striping**.
- Data jsou rozprostřena přes disky.

Disk 0	Disk 1	Disk 2	Disk 3
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Ukázka RAID 0

- Není to opravdový RAID — chybí redundance.



RAID 0: velikost chunků

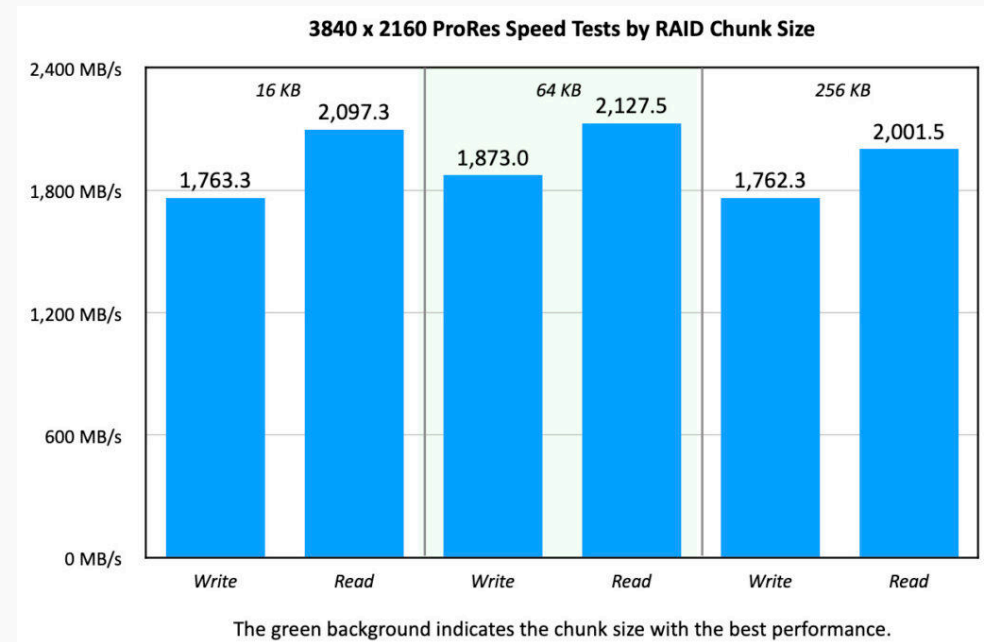
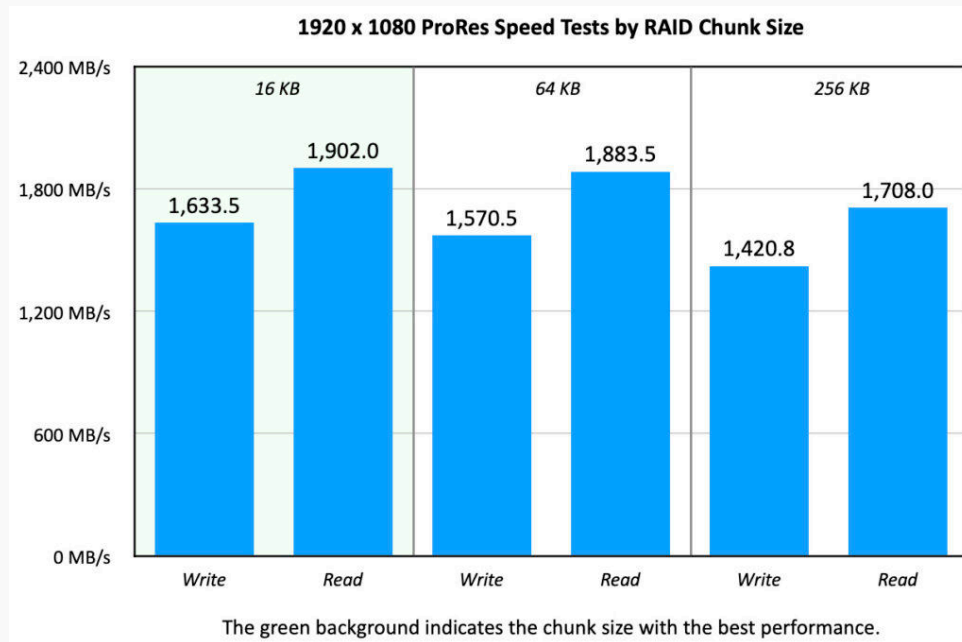
Disk 0	Disk 1	Disk 2	Disk 3
0	2	4	6
1	3	5	7
8	10	12	14
9	11	13	15

Ukázka RAID 0 s velikostí chunku 2

Co nám ovlivňuje velikost chunku?



RAID 0: velikost chunků



Porovnání různých velikostí chunků^{<2>}

- Většinou se nastavuje větší velikost chunku (např. 64KB), my budeme uvažovat jen 1 blok (4KB).

^{<2>} <https://larryjordan.com/articles/explaining-raid-chunk-size-and-which-to-pick-for-media/>



- Předpokládáme, že máme N (stejných) disků, každý s B bloky.
 - Dále budeme analyzovat sekvenční a náhodné přístupy k datům.
 - Předpokládáme, že sekvenční throughput jednoho disku je S a náhodný R .
- Kapacita je $N \cdot B$.
 - Sekvenční throughput je $N \cdot S$.
 - Náhodný throughput je $N \cdot R$.
 - Žádná spolehlivost — výpadek libovolného disku znamená ztrátu dat.



RAID 1: mirroring

- Data jsou zrcadlena na druhou polovinu disků.

Disk 0	Disk 1
0	0
1	1
2	2
3	3

Ukázka RAID 1

Disk 0	Disk 1	Disk 2	Disk 3
0	0	1	1
2	2	3	3
4	4	5	5
6	6	7	7

Ukázka RAID 1+0



- Kapacita je $(N \cdot B)/2$.
- Sekvenční throughput je $N \cdot \frac{S}{2}$.
- Throughput náhodného čtení je $N \cdot R$.
- Throughput náhodného zápisu je $N \cdot \frac{R}{2}$.
- Může tolerovat výpadek 1 libovolného disku.^{<3>}

^{<3>} Očividně lze tolerovat výpadek až $N/2$ disků, ale ne libovolných.



RAID 4

- Použitím **parity** můžeme získat redundanci s vyšší kapacitou.

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
0	1	2	3	P_0
4	5	6	7	P_1
8	9	10	11	P_2
12	13	14	15	P_3

Ukázka RAID 4

- Pro výpočet parity můžeme použít XOR, např. $P_0 = D_0 \oplus D_1 \oplus D_2 \oplus D_3$.

Proč tohle funguje?

Při výpadku disku 1 můžeme získat první blok jako $D_1 = D_0 \oplus P_0 \oplus D_2 \oplus D_3$.



- Kapacita je $(N - 1) \cdot B$.
- Throughput sekvenčního přístupu je $(N - 1) \cdot S$.^{<4>}
- Throughput náhodného čtení je $(N - 1) \cdot R$.
- Throughput náhodných zápisů je $R/2$.^{<5>}

^{<4>}Předpokládáme, že se děje „full-stripe write“ — v našem příkladu se zapíše do všech disků 0-3 a poté se zapíše parita.

^{<5>}Tady se právě nemůžou stát „full-stripe write“. Předpokládáme subtraktivní počítání parity viz další slide.



- Jeden způsob jak počítat paritu, je přechíst disky 0-3, vyxorovat je a výsledek uložit do disku 4.
 - Tomuhle se někdy říká aditivní metoda.
- Existuje i jiný způsob, pro nová data C_{new} platí $P_{\text{new}} = C_{\text{old}} \oplus C_{\text{new}} \oplus P_{\text{old}}$.
 - Tomuhle se říká subtraktivní metoda.
 - Tady jde vidět proč náhodné zapisování má throughput $R/2$ — bottleneck je paritní disk!

Jak vyřešit tento problém s bottleneckem?



- Řešením předchozího problému je **rotování parity přes všechny disky**.

Disk 0	Disk 1	Disk 2	Disk 3	Disk 4
0	1	2	3	<i>P0</i>
5	6	7	<i>P1</i>	4
10	11	<i>P2</i>	8	9
15	<i>P3</i>	12	13	14
<i>P4</i>	16	17	18	19

Ukázka RAID 5



- Kapacita je $(N - 1) \cdot B$.
- Throughput sekvenčního přístupu je $(N - 1) \cdot S$.
- Throughput náhodného čtení je $N \cdot R$!
- Throughput náhodných zápisů je $(N/4) \cdot R!$ ^{<6>}

^{<6>} Pokud počítáme paritu subtraktivně, tak na 1 zápis potřebujeme 4 I/O operace, proto $N/4$.



- Kapitoly *Hard Disk Drives*^{<7>} a *Redundant Arrays of Inexpensive Disks*^{<8>} z *Operating Systems: Three Easy Pieces*.

^{<7>} <https://pages.cs.wisc.edu/~remzi/OSTEP/file-disks.pdf>

^{<8>} <https://pages.cs.wisc.edu/~remzi/OSTEP/file-raid.pdf>

