



# PA152: Efektivní využívání DB

## 1. Úvod

Vlastislav Dohnal

# Poděkování

- Zdrojem materiálů tohoto předmětu jsou:
  - Přednášky CS245, CS345, CS345
    - Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom
    - Stanford University, California
  - Přednášky dřívější verze PA152 (podzim 2008)
    - Pavel Rychlý
    - Fakulta informatiky, Masarykova Univerzita

# Literatura

## ■ Knihy

### □ Database Systems Implementation

- Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom
- Prentice Hall, 2000
- Signatura knihovny D89

### □ Database Systems: The Complete Book

- Hector Garcia-Molina, Jeffrey D. Ullman, Jennifer Widom
- 2nd edition, Prentice Hall, 2009
- Signatura knihovny D147

# Požadavky pro ukončení **ZKOUŠKOU**

- Vypracování 3 domácích úloh
  - zadání pošlu e-mailem
  - každá hodnocená 0-5 body
  - odevzdání v termínu (pozdě  $\Rightarrow$  0 bodů)
  - samostatné vypracování (opis  $\Rightarrow$  0 bodů)
- Složení zkouškové písemky
  - 6 příkladů, každý za 6 bodů
- Hodnocení
  - Součet bodů z domácích úloh a zkouškové písemky
  - $A \geq 47$ ,  $B \geq 42$ ,  $C \geq 37$ ,  $D \geq 32$ ,  $E \geq 27$ ,  $F < 27$

# Požadavky pro ukončení **ZÁPOČTEM**

## ■ Vypracování 3 domácích úloh

- zadání pošlu e-mailem
- každá hodnocená 0-5 body
- odevzdání v termínu (pozdě  $\Rightarrow$  0 bodů)
- samostatné vypracování (opis  $\Rightarrow$  0 bodů)

## ■ Hodnocení

- $\geq 10$  bodů** z domácích úloh

## ■ Pozor:

- některé studijní obory mají PA152 jako povinný předmět, pak musíte mít zapsanu zkoušku

# Základní pojmy

## ■ „Databáze“

- „Programuje“ většina programátorů
- potřebuje každá firma
- je součástí většiny aplikací

## ■ Relační model

- Struktura – data v relacích (tabulkách)
- Operace – dotazování, modifikace
  - SQL, relační algebra

# Příklad implementace DB systému

- Implementovat databázi je snadné:
  - příkaz: relace → výsledek
- Relace jsou v souborech na disku
  - Relace R je v /usr/db/R

```
Miller # 123 # CS
Peterhansel # 522 # EE
:
```

# Příklad 2

- Seznam platných relací v *adresáři*
  - /usr/db/directory

```
R # name # STR # id # INT # dept STR ...  
R2 # C # STR # A # INT ...  
:  
:
```



# Příklad 3

- Dotaz v „SQL“

```
& select A,B  
from R,S  
where R.A = S.A and S.C > 100
```

<u>A</u>	<u>B</u>
123	CAR
522	CAT

&

# Příklad 4

```
select * from R where podmínka
```

## ■ Zpracování dotazu

1. přečti adresář (dictionary) a zjisti atributy relace R

2. čti soubor R a pro každý řádek:

- a. vyhodnot' podmínku

- b. pokud je platná, přidej do výsledku

# Příklad 5

`select A,B from R,S where podmínka`

## ■ Zpracování dotazu

1. Přečti adresář a získej atributy R a S

2. Čti soubor R a pro každý řádek:

a. Čti soubor S a pro každý řádek:

i. Spoj oba řádky (n-tice)

ii. Vyhodnot' podmínku

iii. Pokud je platná,

proved' projekci a přidej do výsledku

# Problémy implementace

## ■ Způsob ukládání

- Řádky formátovány pomocí oddělovačů

- Změna hodnoty vede ke změně v celém souboru

- Neúsporné ukládání

- Mazání je drahé

## ■ Vyhledávání je drahé

- Nejsou indexy

- Hledání podle primárního klíče je pomalé

- Vždy je nutné přečíst celou relaci

# Problémy implementace

- Žádné souběžné zpracování
- Žádná spolehlivost
  - Možná ztráta dat
  - Operace nemusí být dokončena
- Žádná bezpečnost
  - Přístup na úrovni systému souborů (filesystem)
  - Práva jsou příliš hrubozrnná
- Není API, není GUI

# Osnova kurzu

## ■ Úložiště dat

Hierarchie pamětí, RAID, výpadky, ...

## ■ Struktura ukládání dat

Záznamy, bloky, ...

## ■ Indexování

Stromy, hašování, ...

## ■ Zpracování dotazů

Odhad ceny, způsoby spojování relací, ...

## ■ Optimalizace dotazů

Vytváření indexů, problémy pohledů, rozdělování relací, ...

# Osnova kurzu

- **Optimalizace databáze**

Úpravy relačního schéma, monitorování db, ...

- **Transakční zpracování**

Souběžné zpracování, zamykání, logování, uváznutí, ...

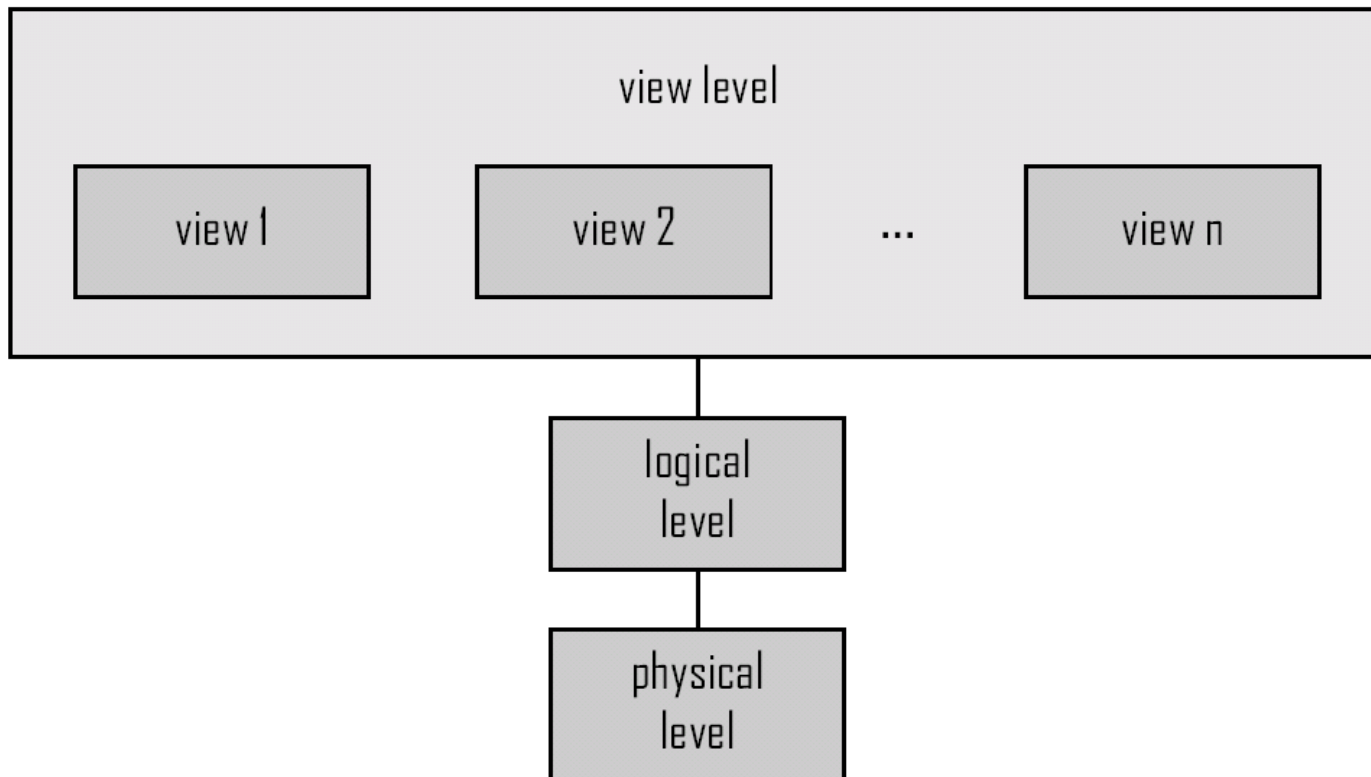
- **Bezpečnost**

Práva, ochrana dat, ...

# Databázový systém

- DBMS (Database Management System)

- Architektura:





# Hlavní součásti databázového systému

## ■ Storage Manager

- správa bloků na disku
- správa vyrovnávací paměti

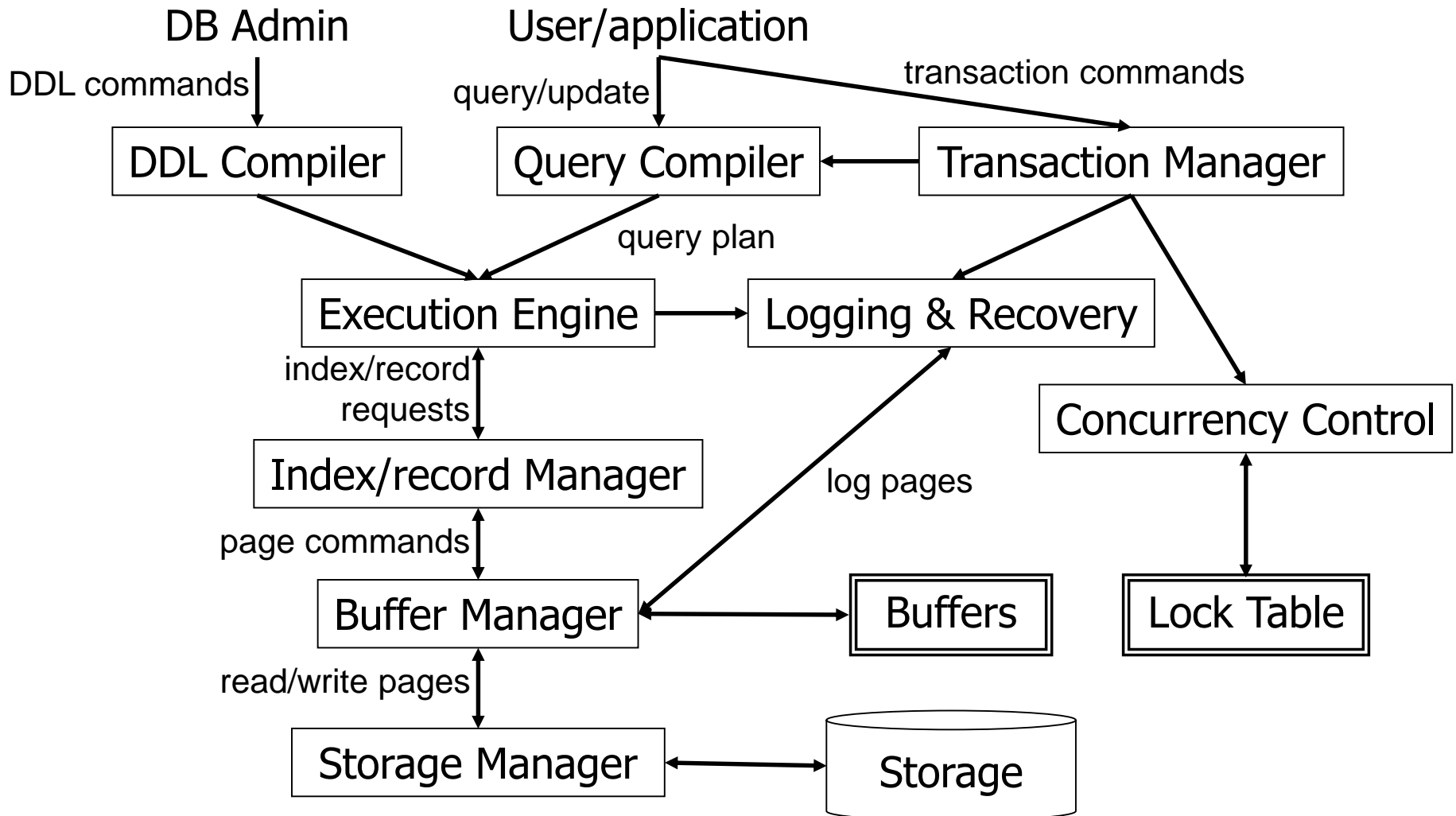
## ■ Query Processor

- překlad dotazu, optimalizace
- vyhodnocení dotazu

## ■ Transaction Manager

- atomičnost, izolovanost a trvalost transakcí

# Části databázového systému



# Hierarchie pamětí



## ■ Primární

vyrovnávací (cache)

■ procesor

hlavní (operační)

■ RAM

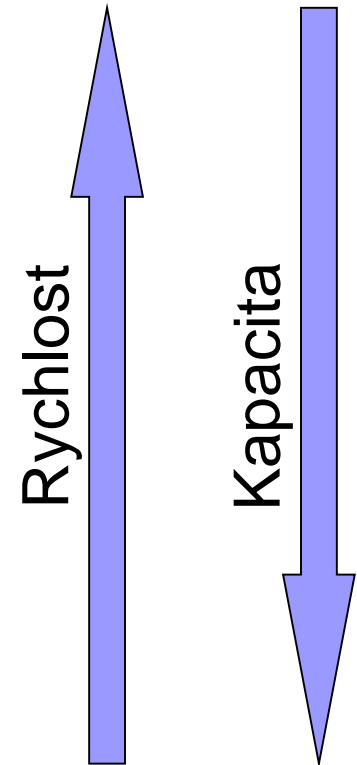
## ■ Sekundární

disk, flash

## ■ Terciární

záložní

■ pásy, optické disky



# Mooreův zákon

## ■ Počet tranzistorů

- Zdvojnásobení cca každé 2 roky

- Rychlost procesorů

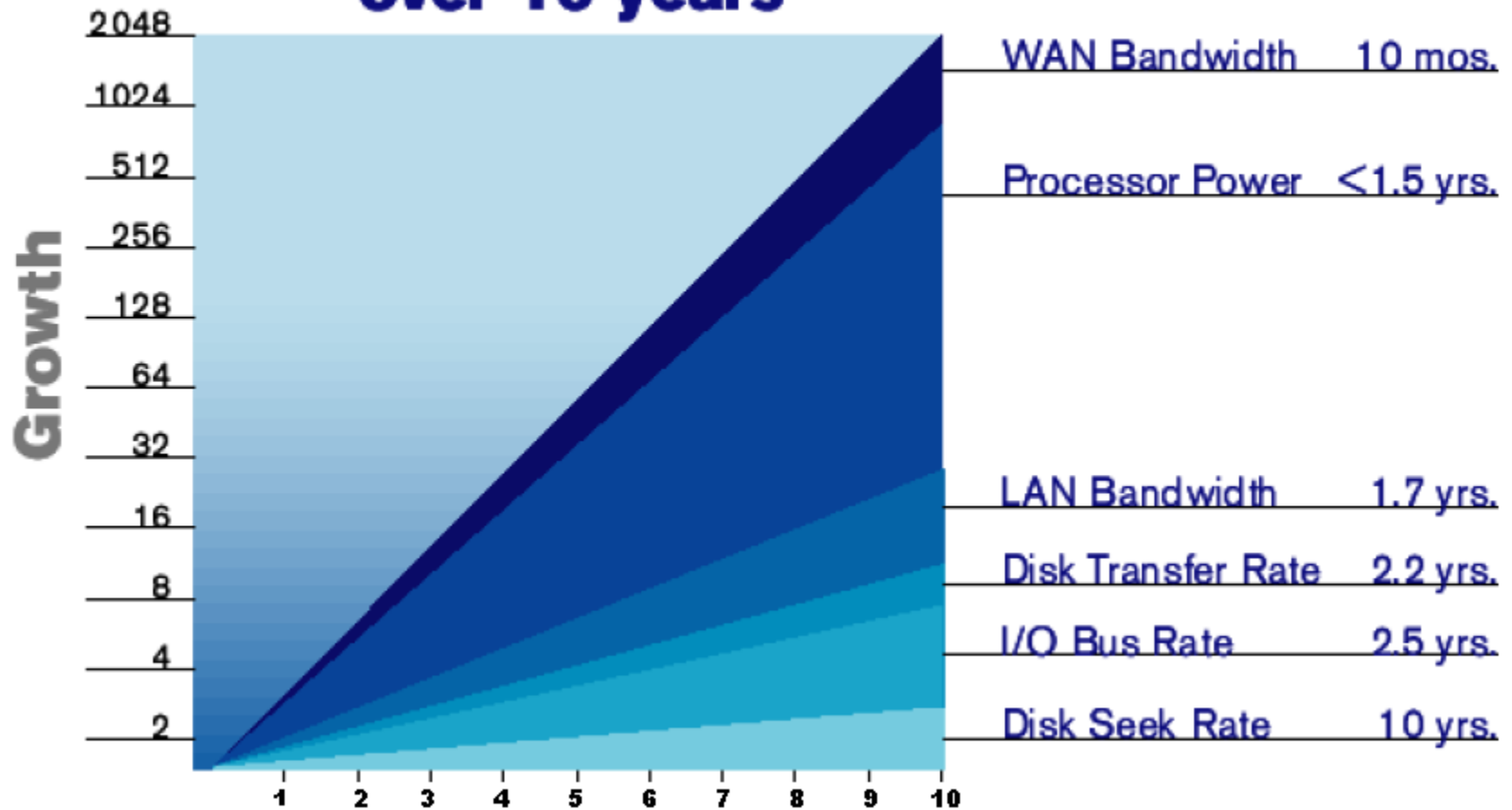
- Kapacita pamětí

- Kapacita disků (Kryder's Law)

## ■ Pozor: neplatí pro rychlost disku

# Mooreův zákon

## Technology Growth Rates over 10 years



# Paměťová úložiště

## ■ Cache

- Nejrychlejší a nejdražší, závislé na napájení

## ■ RAM

- Rychlé – 10-100 ns ( $1 \text{ ns} = 10^{-9} \text{ s}$ )
- Příliš malé nebo drahé pro uložení celé databáze
- Závislé na napájení

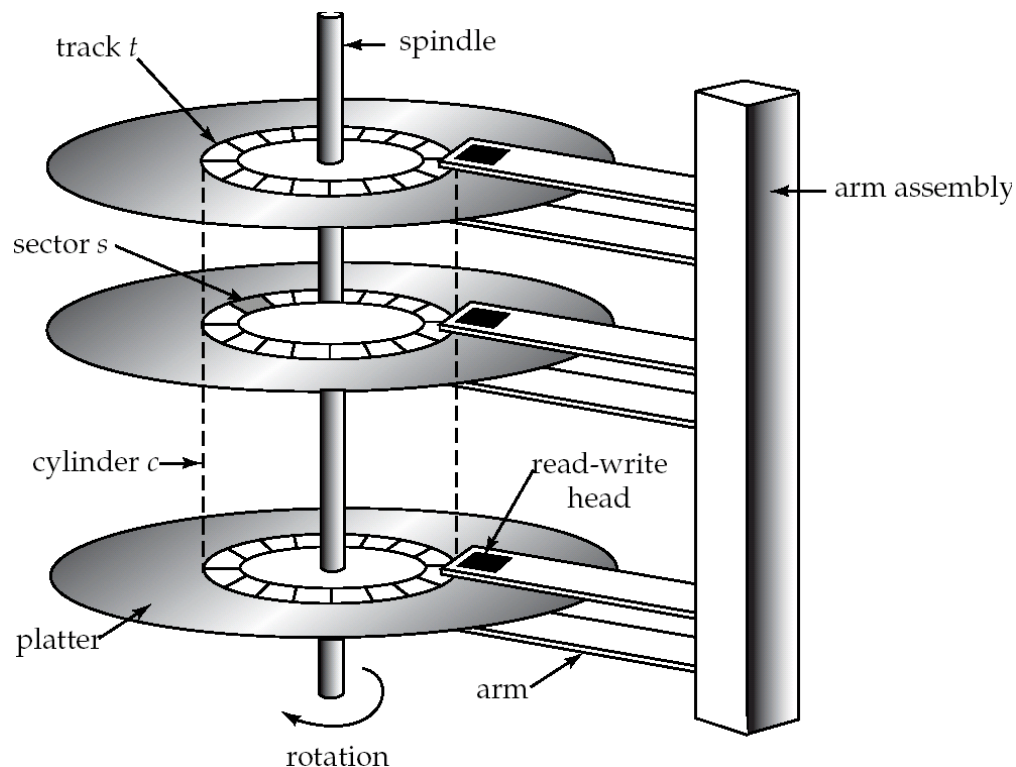
## ■ Flash

- Rychlé čtení, nezávislé na napájení
- Pomalý zápis – nejdříve smazat, pak zápis
  - Zapisuje se celá oblast (banka)
- Omezený počet zapisovacích cyklů

# Diskové úložiště

## ■ Rotační disk

- Velká kapacita, nezávislý na napájení
- Čtení a zápis téměř stejně rychlé



# Rotační disk

- Přístupová doba (access time)
  - Čas mezi požadavkem na čtení/zápis a počátkem přenosu dat
- Data jsou blokována
  - atomická jednotka čtení je sektor/diskový blok
- Složky:
  - Vystavení hlaviček (seek time) – 4-10 ms
    - Přesun na správnou stopu disku
    - Average seek time =  $\frac{1}{2}$  nejhorší případ seek time
  - Rotační zpoždění – 4-11ms (5400-15k rpm)
    - Čas pro otočení disku na správný sektor
    - Average latency =  $\frac{1}{2}$  nejhorší případ latency



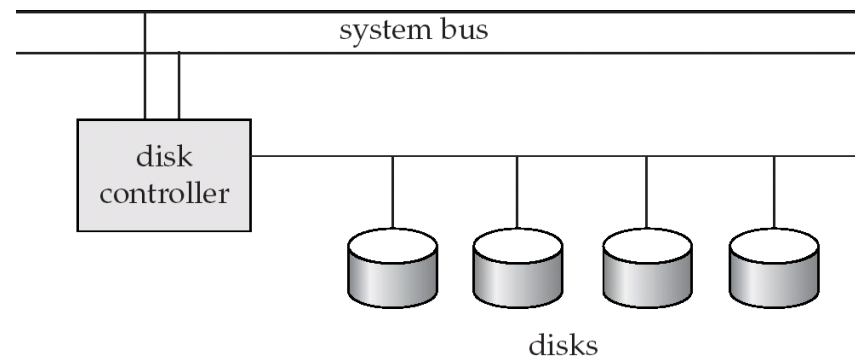
# Rotační disk (pokrač.)

- Přenosová rychlost

- Rychlost čtení/zápisu dat z/na disk
- 50-200MB/s, nižší pro vnitřní stopy

- Rychlost řadiče

- Více disků připojených na jeden řadič
- SATAIII (6Gb/s) – 600 MB/s
- SCSI Ultra 640 – 640 MB/s



# Rotační disk

## ■ Vlastnosti

- Náhodné čtení je pomalé
  - access time can be up to 20ms
- Sekvenční čtení je rychlé

## ■ Optimalizace přístupu v HW

- Cache
  - Buffery pro zápis, zálohovány baterií nebo flash
- Algoritmy pro minimalizaci pohybu hlavičky
  - Algoritmus „výtah“
  - Fungují pouze při velkém počtu požadavků současně

# Rotační disk

## ■ Příklad

- SATAII disk, 7200rpm, 100MB/s, 8.9ms seek
  - Avg seek time = 8.9ms
  - Avg latency =  $(1/(7200/60))*0.5=0.00417s=4.17ms$
- Čtení sektoru (512B) = 13.07ms + 4.88μs
- 10MB souvisle = 13.07ms + 100ms = 113ms
  - souvislé čtení obsahuje i
    - přesun na další stopu – změna povrchu / cylindru
    - toto zanedbáváme
- 10MB náhodně =  $20480*13.07488ms = 268s$

# Diskové operace

- Přístup po blocích (atomická jednotka)
  - Skupina sousedních sektorů
  - Typicky 4KB – 16KB
- Čtení bloku
- Zápis bloku
  - Zápis a ověření (otočení disku + čtení !)
- Modifikace bloku:
  - Čtení
  - Změna v paměti
  - Zápis a ověření

# Algoritmy v DB

- Pracují s bloky
- Minimalizují počet náhodně čtených bloků
- Náklady na čtení a zápis jsou stejné

# Databáze na FI

- MySQL, PostgreSQL, Oracle

- PostgreSQL <http://www.postgresql.org/>

- Zdarma i pro komerční aplikace

- Technické informace <http://www.fi.muni.cz/tech/unix/databaze.shtml>

- Podle návodu si vytvořte účet

- Pro připojení použijte pgAdmin <http://www.pgadmin.org/>

- Nebo phpPgAdmin <http://phpPgAdmin.sourceforge.net/>

- Dostupný na <https://mufin.fi.muni.cz/phpPgAdmin/>

- Zvolte server DB FI MUNI,

- databázi pgdb,

- schéma podle Vašeho loginu

- Nebo Adminer <https://mufin.fi.muni.cz/adminer/>