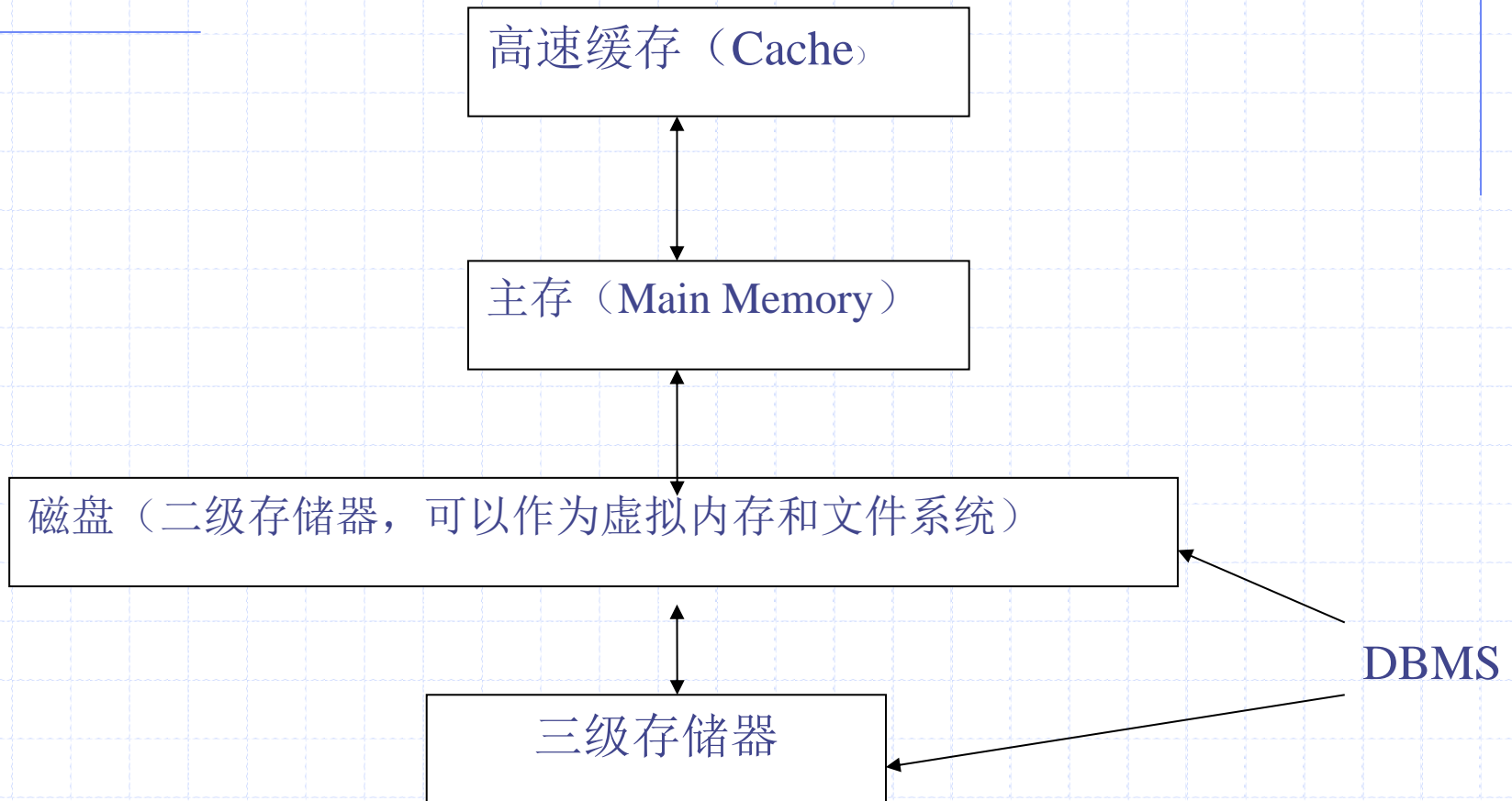# Database System Principles

chapter 2: Data storage

Outline

- Hardware: Disks
- Access Times
- Optimizations
- Other Topics:
  - Storage costs
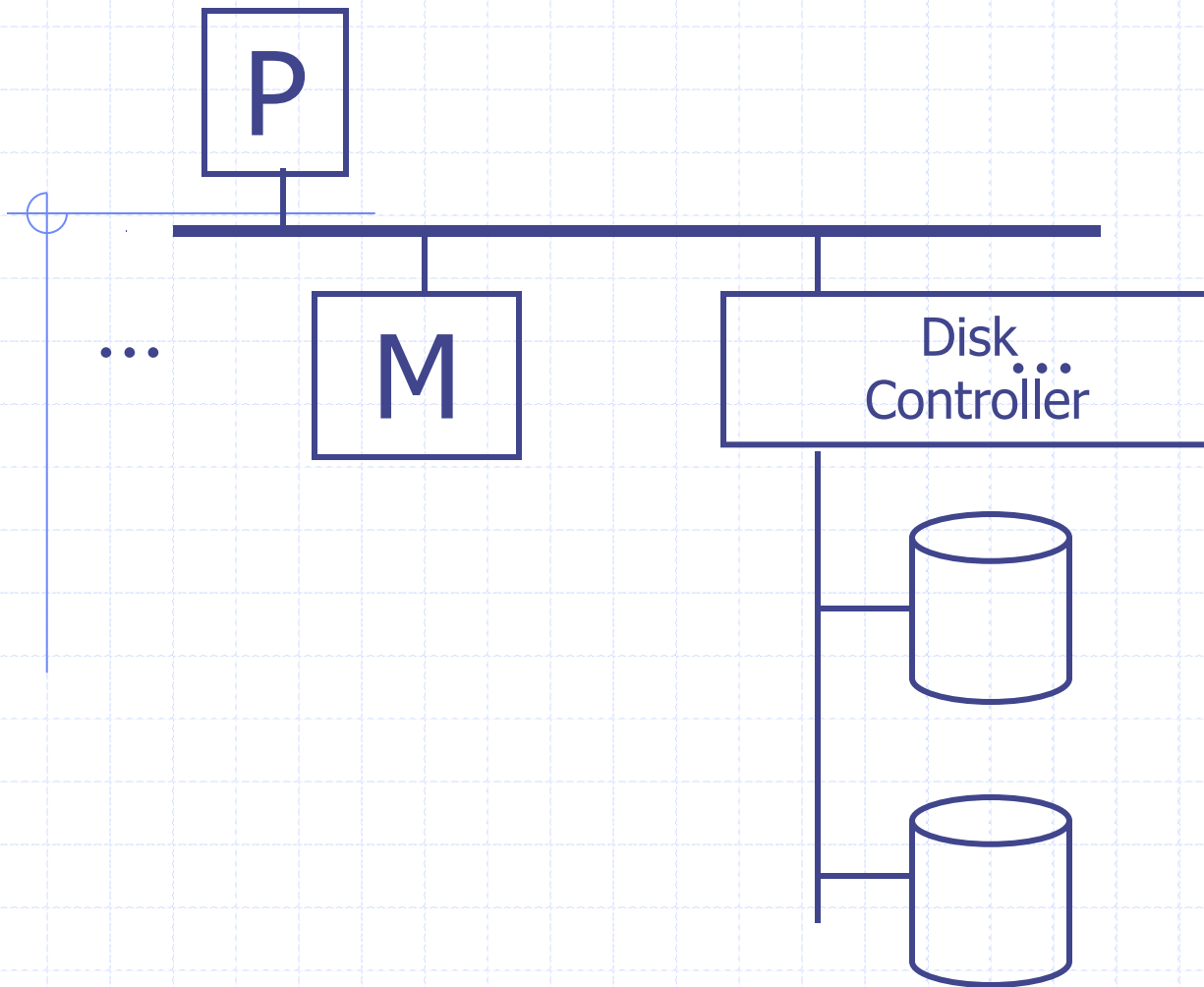 - Using secondary storage
  - Disk failures

Hardware

DBMS

Data Storage

高速缓存（Cache）

主存（Main Memory）

磁盘（二级存储器，可以作为虚拟内存和文件系统）

三级存储器

DBMS

P

... M

Disk
Controller ...

Typical
Computer

Secondary
Storage

Processor

 Fast, reduced instruction set, with cache
 Speed: 100 → 500 → 1000 MIPS
处理器的频率目前在3GHz到4GHz之间，能达到
10,000 -100,000MIPS数量级。Intel Core i7
4770k 处理器，3.9GHz频率，127,273 MIPS。

Memory

 Fast, slow, non-volatile, read-only,...
 Access time: 10^{-6} → 10^{-9} sec.
 1 μs → 1 ns

Notes: MIPS (million instructions per second)

Secondary storage

      - Disk: Floppy

                      Removable Packs

                      Optical, CD-ROM...

                      Arrays

      - Tape

机械硬盘读取速度有150M/s-300M/s，写速度在100M/s以内。

磁盘和文件

- ◆ DBMS把数据存放在硬盘上 ("hard") disks.
- ◆ DBMS的主要功能
  - 读READ: 把数据从磁盘读到内存 (RAM).
  - 写WRITE: 把数据从内存写到磁盘.
  - 相对于内存中的操作，读些操作都是很耗时的，在设计DBMS时，需要认真计划!

为什么不把所有的内容放在内存?

◈ 花销大.

◈ 内存是数据是挥发的. 掉电后，数据就丢失了

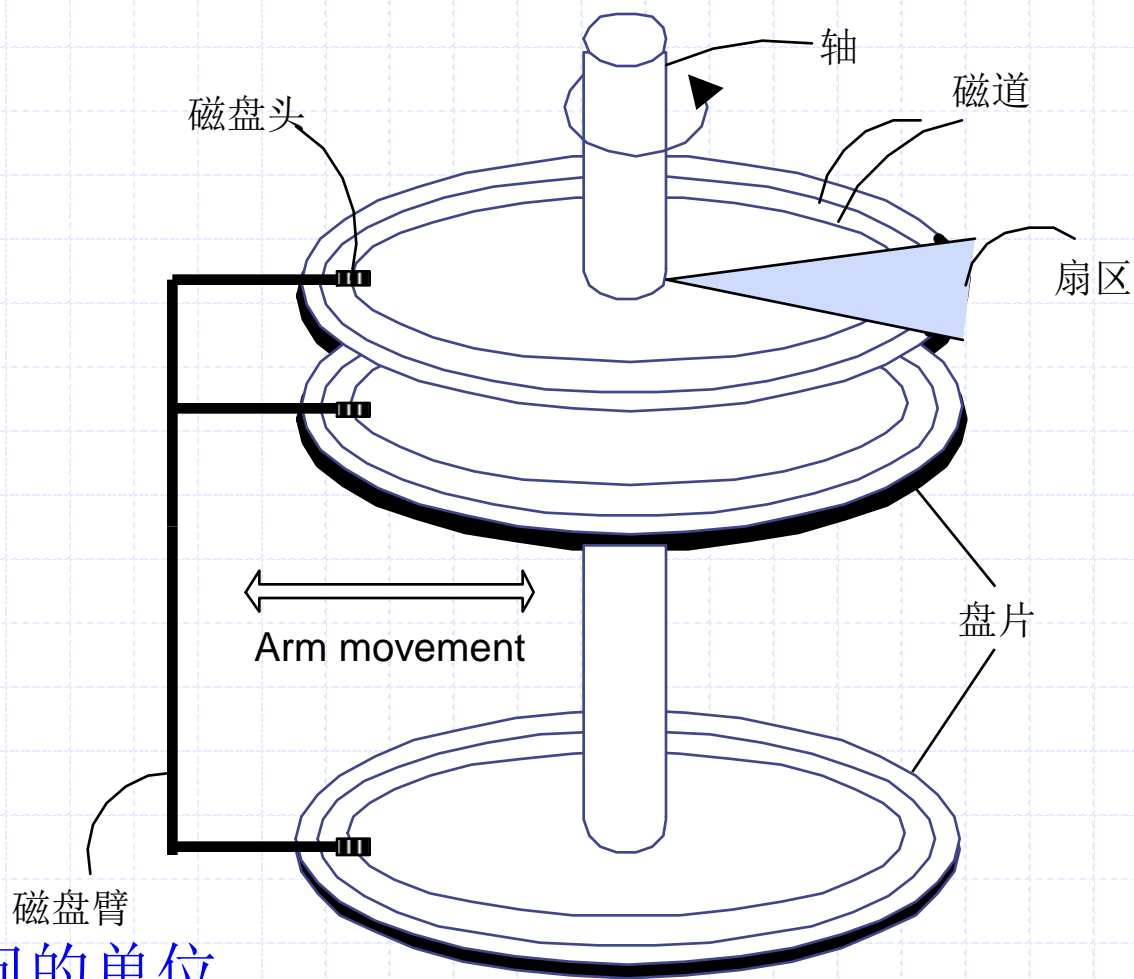◈ 典型的存储结构:

- 内存 (RAM)：临时使用的数据.

- 磁盘：二级存储.

- 磁带：数据库中数据的归档.

磁盘

- ◆ 二级存储设备.
- ◆ 磁盘**vs.**磁带：随机存取**vs.** 顺序读取.
- ◆ 存取的数据单元：磁盘块、页面*pages*.

磁盘的组成

v 盘片绕轴旋转.

v 一次只能读或写

v *扇区大小是固定的.*访问的单位
为磁盘块，磁盘块可由多个扇区
组成

磁盘头

轴

磁道

扇区

Arm movement

盘片

磁盘臂

存取一个页面

◆ 读写时间:
- *寻道时间**seek time**
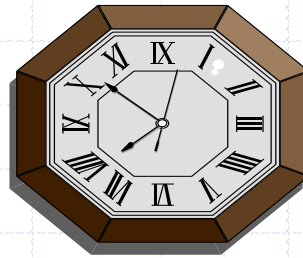- *旋转延迟**rotational delay** (waiting for block to rotate under head)
- *传输时间**transfer time**

◆ 寻道时间和旋转延迟占主要部分.
- 寻道时间大约 1微秒到 20msec
- 旋转延迟0微秒到 10msec
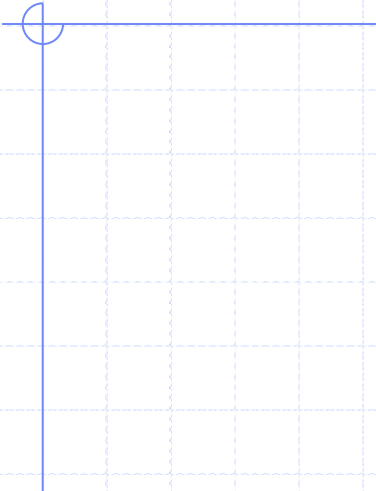- 传输时间每 4KB大约1msec

◆ **I/O**的主要影响因素: 减少寻道和旋转延迟!

Disk Access Time

I want
block X →

?

→ block x
in memory

Time =   Seek Time +
         Rotational Delay +
         Transfer Time +
         Other

Rotational Delay

Head Here

Block I Want
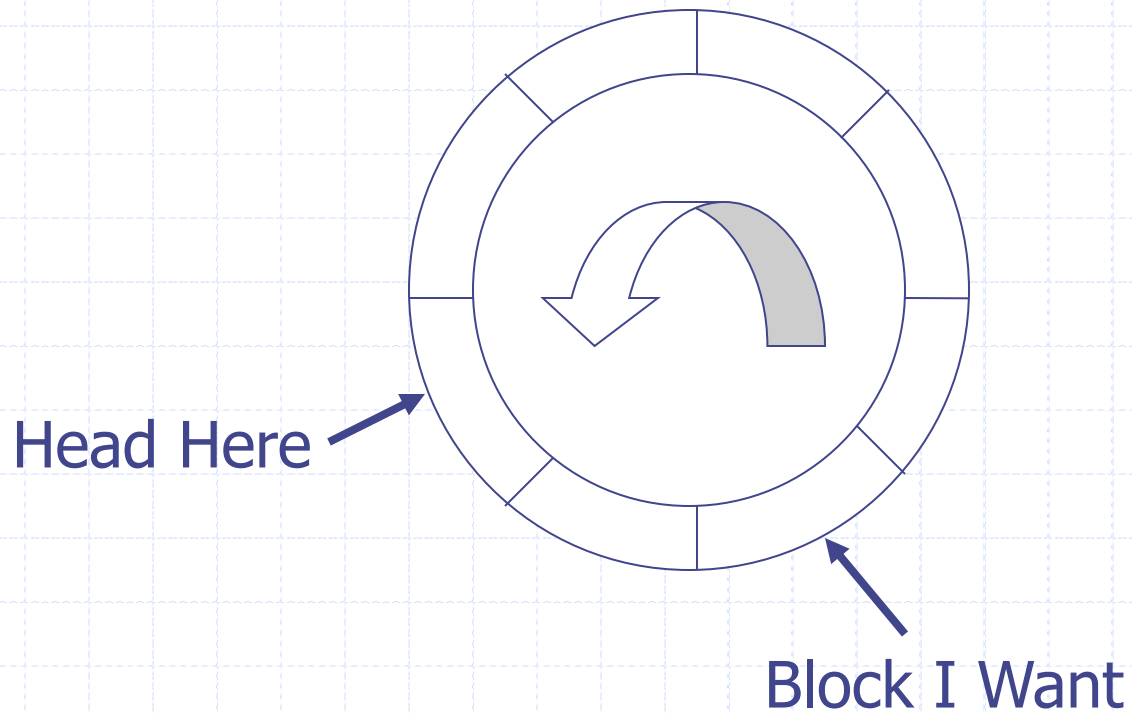
Other Delays

- CPU time to issue I/O
- Contention for controller
- Contention for bus, memory

"Typical" Value: 0

Block Address:

- Physical Device
- Cylinder #
- Surface #
- Sector

Outline

- Hardware: Disks
- Access Times
- Optimizations
- Other Topics
  - Storage Costs
  - Using Secondary Storage
  - Disk Failures

here ⟸

Optimizations (in controller or O.S.)

- Cylinder-based Organization
- Disk Scheduling Algorithms
  - e.g., elevator algorithm
- <u>Double</u> Buffer
- Pre-fetch
- Mirrored Disks

磁盘数据的组织

- ◆ 相邻块:
  - blocks on same cylinder
  - blocks on same track
- ◆ 文件尽量在硬盘上为相邻块

Single Buffer Solution

(1) Read B1 $\rightarrow$ Buffer

(2) Process Data in Buffer

(3) Read B2 $\rightarrow$ Buffer

(4) Process Data in Buffer ...

Say   P = time to process/block

        R = time to read in 1 block
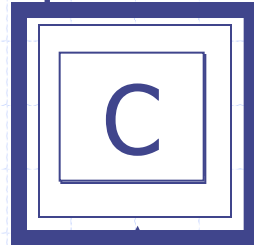
        n = # blocks

Single buffer time

                        = n(P+R)

Double Buffering

Memory:

process   process

$$C \quad\quad\quad\quad B$$

Disk:

		C	D	E	F	G	
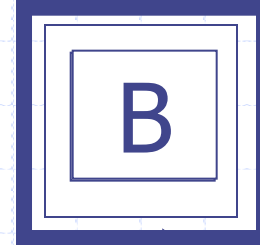
done done

Say $P \geq R$

P = Processing time/block
R = IO time/block
n = # blocks

<u>What is processing time?</u>

◈ Double buffering time   = R + nP

◈ Single buffering time   = n(R+P)

Block Size Selection?

- Big Block $\rightarrow$ Amortize I/O Cost

Unfortunately...

- Big Block $\Rightarrow$ Read in more useless stuff!

    and takes longer to read

Trend

- As memory prices drop, blocks get bigger …

Storage Cost

typical capacity (bytes)

10^{15}

10^{13}

10^{11}

10^{9}

10^{7}

10^{5}

10^{3}

10^{-9} 10^{-6} 10^{-3} 10^{-0} 10^{3}

access time (sec)

cache

electronic main

electronic secondary

magnetic optical disks

online tape

nearline tape & optical disks

offline tape

from Gray & Reuter

Storage Cost

A log-log plot of dollars/MB (y-axis) versus access time (sec) (x-axis).

- cache — at 10^4 dollars/MB, near 10^{-9} sec
- electronic main — near 10^2 dollars/MB
- electronic secondary
- magnetic optical disks — near 10^0 dollars/MB, near 10^{-3} sec
- online tape — near 10^2 dollars/MB
- nearline tape & optical disks — near 10^0 dollars/MB
- offline tape — near 10^{-2} dollars/MB, near 10^3 sec

y-axis: dollars/MB — 10^4, 10^2, 10^0, 10^{-2}, 10^{-4}

x-axis: access time (sec) — 10^{-9}, 10^{-6}, 10^{-3}, 10^{-0}, 10^3

Buffer Management缓冲区管理

来自高层次代码的页请求

缓冲池

磁盘页

空闲帧

MAIN MEMORY

DISK

DB

空闲帧的选择由
replacement policy替代策略决定
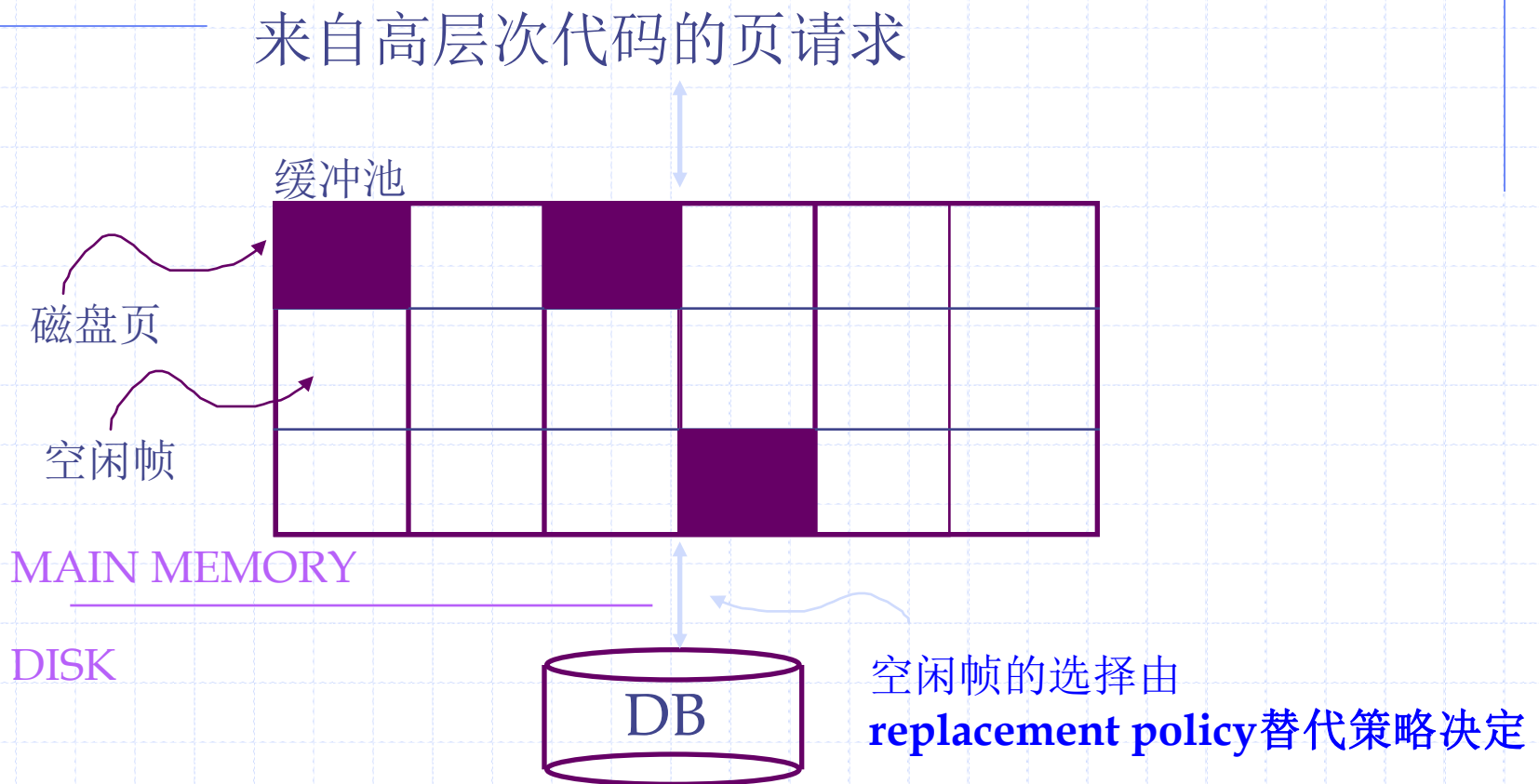
◈ *数据必须在主存中，才能被DBMS所处理!*
◈ *系统维护表 <frame#, pageid,pin-count,Dirty>.*

页面请求

◆ 当页面不在缓冲池中:
- 首先选择一个替换帧*replacement*
- 如果帧的**dirty**位是真,把该帧写到磁盘上
- 把请求的页读入替换帧

◆ *把替换帧的主存地址返回给申请者*

*如果可以预测请求（例如：顺序扫描），
*那么页面可以被预取*pre-fetched*!

缓冲区替换策略

◆ *替换策略replacement policy:*
- 最近最少使用策略Least-recently-used (LRU)（链表实现）
- 时钟替换Clock
- 最近经常使用策略MRU等.

Using secondary storage effectively (Sec. 2.3)

◆ Example: Sorting data on disk
◆ Conclusion:
  ■ I/O costs dominate
  ■ Design algorithms to reduce I/O

◆ **Merge Sort**
  ■ **Two-phase, Multiway merge sort**
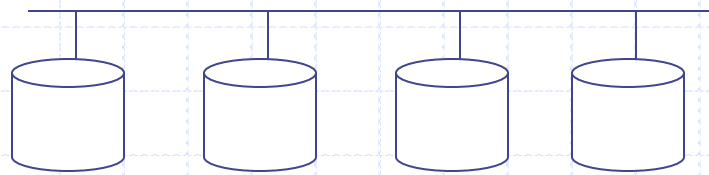
Disk Failures (Sec 2.5)

◆ <u>Coping with Disk Failures</u>

- Detection
  - ◆ e.g. Checksum

- Correction
  - $\Rightarrow$ Redundancy

At what level do we cope?

- ◆ Single Disk
  - ▪ e.g., Error Correcting Codes
- ◆ Disk Array

Logical ⟷ Physical

RAID

◈ 磁盘阵列: 把几个磁盘组织在一起的一种形式.

◈ 目标: 提高性能和可靠性.

◈ 两种主要技术:

  ■ 数据划分:数据给分成大小相等的段；不同数据段写在不同的磁盘上.

  ■ 冗余: 重复信息允许数据重现.

Summary

- Secondary storage, mainly disks
- I/O times
- I/Os should be avoided,
    especially random ones.....

Any question?

total 39