Storing Data: Disks and Files

Iztok Savnik, FAMNIT

Slides & Textbook

- Textbook:
  - Raghu Ramakrishnan, Johannes Gehrke, *Database Management Systems, McGraw-Hill, 3rd ed., 2007.*
- *Slides:*
  - *From „Cow Book“: R.Ramakrishnan, http://pages.cs.wisc.edu/~dbbook/*

Disks and Files

- DBMS stores information on (“hard”) disks.
- This has major implications for DBMS design!
  - READ: transfer data from disk to main memory (RAM).
  - WRITE: transfer data from RAM to disk.
  - Both are high-cost operations, relative to in-memory operations, so must be planned carefully!

Why Not Store Everything in Main Memory?

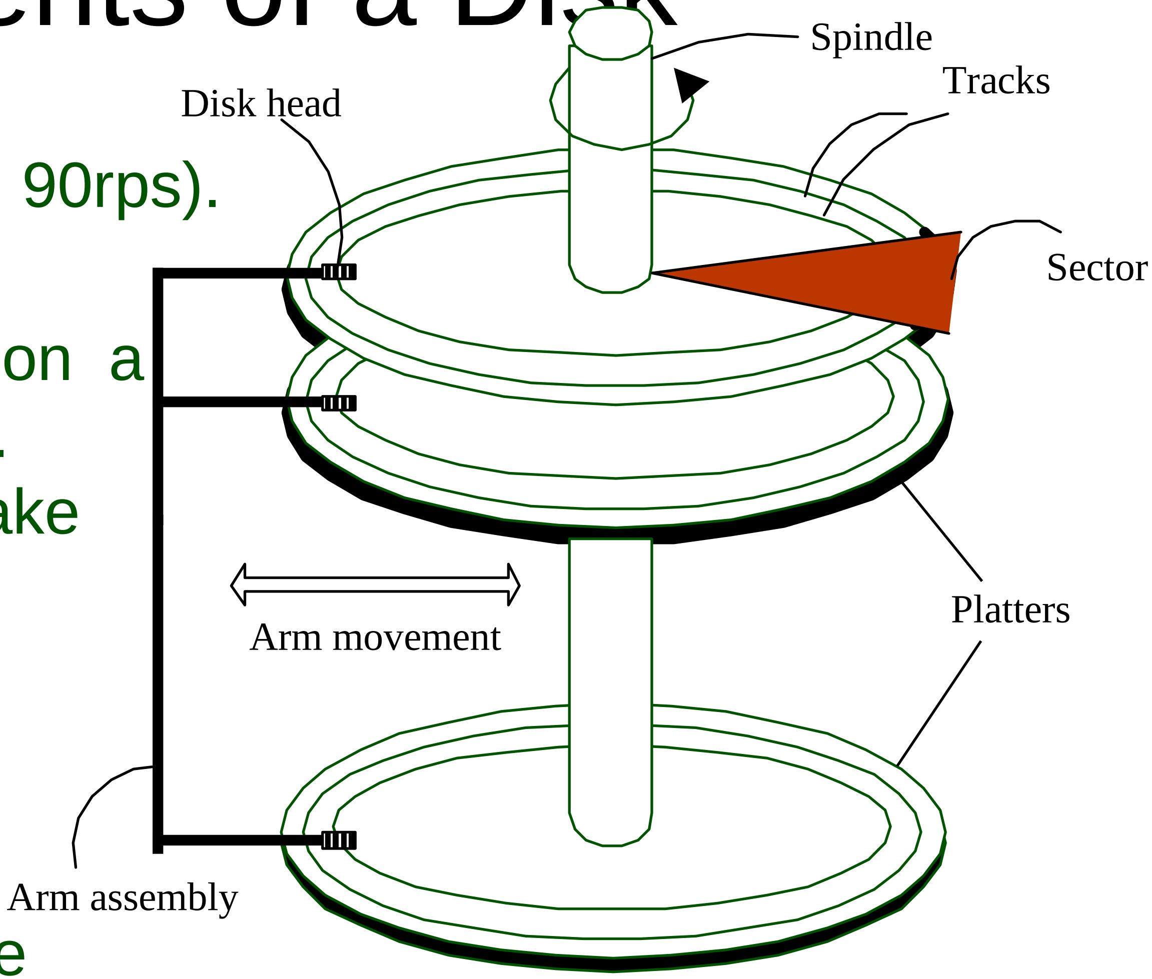
- *Costs too much*. 100 eur will buy you either 16GB RAM or 4TB of disk today.
- *Main memory is volatile*. We want data to be saved between runs. (Obviously!)
- Typical storage hierarchy:
  - Main memory (RAM) for currently used data.
  - Disk for the main database (secondary storage).
  - Tapes for archiving older versions of the data (tertiary storage).

Disks

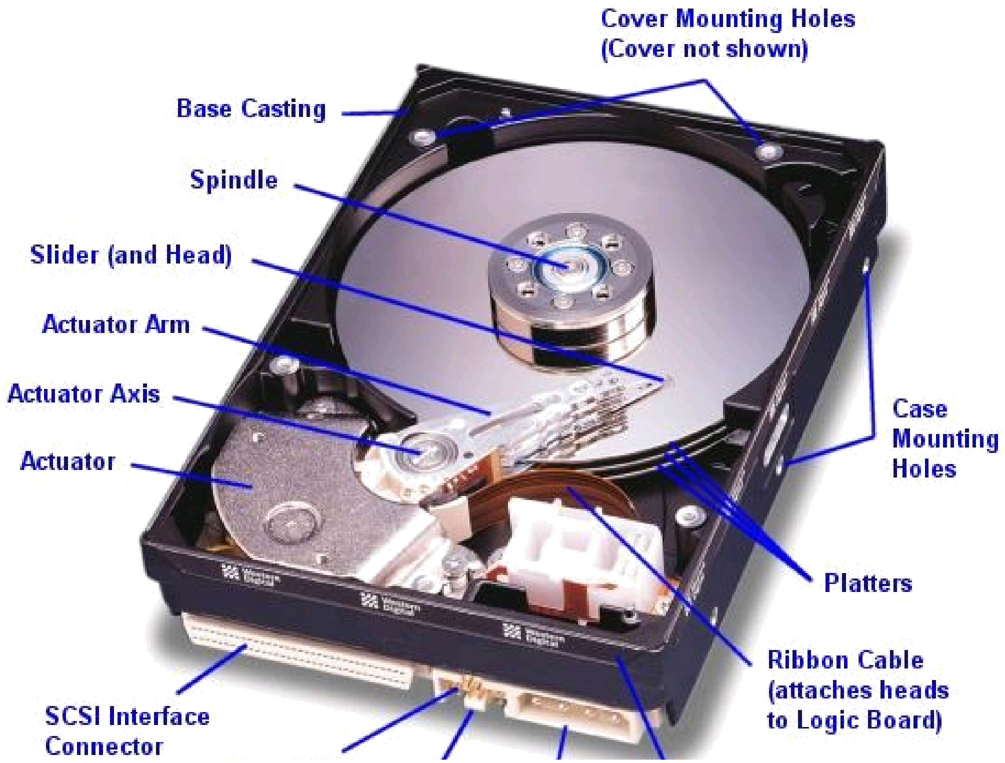
- Secondary storage device of choice.
- Main advantage over tapes: *random access* vs. *sequential*.
- Data is stored and retrieved in units called *disk blocks* or *pages*.
- Unlike RAM, time to retrieve a disk page varies depending upon location on disk.
  - Therefore, relative placement of pages on disk has major impact on DBMS performance!

Components of a Disk

- The platters spin (say, 90rps).
- The arm assembly is moved in or out to position a head on a desired track. Tracks under heads make a *cylinder* (imaginary!).
- Only one head reads/writes at any one time.
- *Block size* is a multiple of *sector size* (which is fixed).

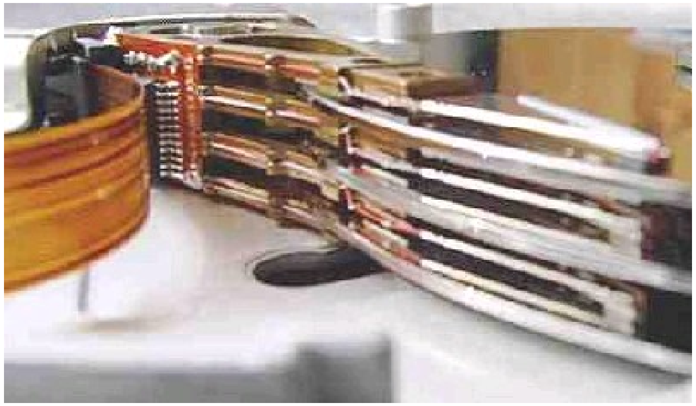


Hard Disk Drives (HDDs)

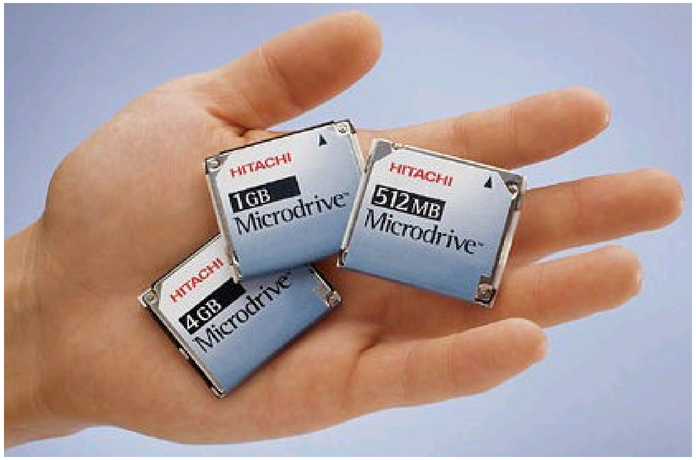


Western Digital Drive
http://www.storagereview.com/guide/

IBM Personal Computer/AT (1986)
- 30 MB hard disk - $500
- 30-40ms seek time
- 0.7-1 MB/s (est.)



Read/Write Head Side View



IBM/Hitachi Microdrive

Accessing a Disk Page

- Time to access (read/write) a disk block:
  - *seek time* (moving arms to position disk head on track)
  - *rotational delay* (waiting for block to rotate under head)
  - *transfer time* (actually moving data to/from disk surface)
- Seek time and rotational delay dominate.
  - Seek time varies from about 1 to 20msec
  - Rotational delay varies from 0 to 10msec
  - Transfer rate is about 1msec per 4KB page
- Key to lower I/O cost: reduce seek/rotation delays! Hardware vs. software solutions?

Barracuda®

The Power of One

Seagate®

Specifications	3TB[1]	2TB[1]	1.5TB[1]	1TB[1]	750GB[1]	500GB[1]	320GB[1]	250GB[1]
Model Number	ST3000DM001	ST2000DM001	ST1500DM003	ST1000DM003	ST750DM003	ST500DM002[2]	ST320DM000[2]	ST250DM000[2]
Interface Options	SATA 6Gb/s NCQ	SATA 6Gb/s NCQ	SATA 6Gb/s NCQ	SATA 6Gb/s NCQ	SATA 6Gb/s NCQ	SATA 6Gb/s NCQ	SATA 6Gb/s NCQ	SATA 6Gb/s NCQ
Performance								
Spindle Speed (RPM)	7200	7200	7200	7200	7200	7200	7200	7200
Cache, Multisegmented (MB)	64	64	64	64	64	16	16	16
SATA Transfer Rates Supported (Gb/s)	6.0/3.0/1.5	6.0/3.0/1.5	6.0/3.0/1.5	6.0/3.0/1.5	6.0/3.0/1.5	6.0/3.0/1.5	6.0/3.0/1.5	6.0/3.0/1.5
Seek Average, Read (ms)	<8.5	<8.5	<8.5	<8.5	<8.5	<11	<11	<11
Seek Average, Write (ms)	<9.5	<9.5	<9.5	<9.5	<9.5	<12	<12	<12
Average Data Rate, Read/Write (MB/s)	156	156	156	156	156	125	125	125
Max Sustained Data Rate, OD Read (MB/s)	210	210	210	210	210	144	144	144
Configuration/Organization								
Heads/Disks	6/3	6/3	4/2	2/1	2/1	2/1	2/1	1/1
Bytes per Sector	4096	4096	4096	4096	4096	4096 or 512[2]	4096 or 512[2]	4096 or 512[2]

WD Red™ Pro

Specifications

Model Number[4]	WD221KFGX	WD201KFGX	WD181KFGX	WD161KFGX	WD141KFGX	WD121KFBX
Formatted capacity[1]	22TB	20TB	18TB	16TB	14TB	12TB
Recording technology	CMR	CMR	CMR	CMR	CMR	CMR
Interface	SATA 6 Gb/s	SATA 6 Gb/s	SATA 6 Gb/s	SATA 6 Gb/s	SATA 6 Gb/s	SATA 6 Gb/s
Form factor	3.5-inch	3.5-inch	3.5-inch	3.5-inch	3.5-inch	3.5-inch
Native command queuing	Yes	Yes	Yes	Yes	Yes	Yes
OptiNAND™ technology	Yes	Yes	No	No	No	No
Advanced Format (AF)	Yes	Yes	Yes	Yes	Yes	Yes
RoHS compliant[5]	Yes	Yes	Yes	Yes	Yes	Yes
Performance						
Interface speed (max)	6 Gb/s	6 Gb/s	6 Gb/s	6 Gb/s	6 Gb/s	6 Gb/s
Internal transfer rate[6]	265 MB/s	268 MB/s	272 MB/s	259 MB/s	255 MB/s	240 MB/s
Cache (MB)[1]	512	512	512	512	512	256
RPM	7200	7200	7200	7200	7200	7200
Reliability/Data Integrity						
Load/unload cycles[7]	600,000	600,000	600,000	600,000	600,000	600,000
Non-recoverable errors per bits read	<10 in 10^{14}	<10 in 10^{14}	<10 in 10^{14}	<10 in 10^{14}	<10 in 10^{14}	<10 in 10^{14}
MTBF (hours)[8]	1,000,000	1,000,000	1,000,000	1,000,000	1,000,000	1,000,000
Workload rate (TB/year)[2]	300	300	300	300	300	300
Limited warranty (years)[3]	5	5	5	5	5	5

- And RAM?
  - What are the differences to HD?
- DDR4
  - 12-15ns latency
  - 12-15 GB/s transfer rate
- DDR5
  - The same latency (to DDR4)
  - 38-50 GB/s transfer rate

excepted to be like this

Standard name	Memory clock (MHz)	I/O bus clock (MHz)	Data rate (MT/s)	Module name	Peak trans-fer rate (MB/s)	Timings CL-tRCD-tRP	CAS latency (ns)
DDR4-1600J* DDR4-1600K DDR4-1600L	200	800	1600	PC4-12800	12800	10-10-10 11-11-11 12-12-12	12.5 13.75 15
DDR4-1866L* DDR4-1866M DDR4-1866N	233.33	933.33	1866.67	PC4-14900	14933.33	12-12-12 13-13-13 14-14-14	12.857 13.929 15
DDR4-2133N* DDR4-2133P DDR4-2133R	266.67	1066.67	2133.33	PC4-17000	17066.67	14-14-14 15-15-15 16-16-16	13.125 14.063 15
DDR4-2400P* DDR4-2400R DDR4-2400T DDR4-2400U	300	1200	2400	PC4-19200	19200	15-15-15 16-16-16 17-17-17 18-18-18	12.5 13.32 14.16 15
DDR4-2666T DDR4-2666U DDR4-2666V DDR4-2666W	333.33	1333.33	2666.67	PC4-21333	21333.33	17-17-17 18-18-18 19-19-19 20-20-20	12.75 13.50 14.25 15
DDR4-2933V DDR4-2933W DDR4-2933Y DDR4-2933AA	366.67	1466.67	2933.33	PC4-23466	23466.67	19-19-19 20-20-20 21-21-21 22-22-22	12.96 13.64 14.32 15
DDR4-3200W DDR4-3200AA DDR4-3200AC	400	1600	3200	PC4-25600	25600	20-20-20 22-22-22 24-24-24	12.5 13.75 15

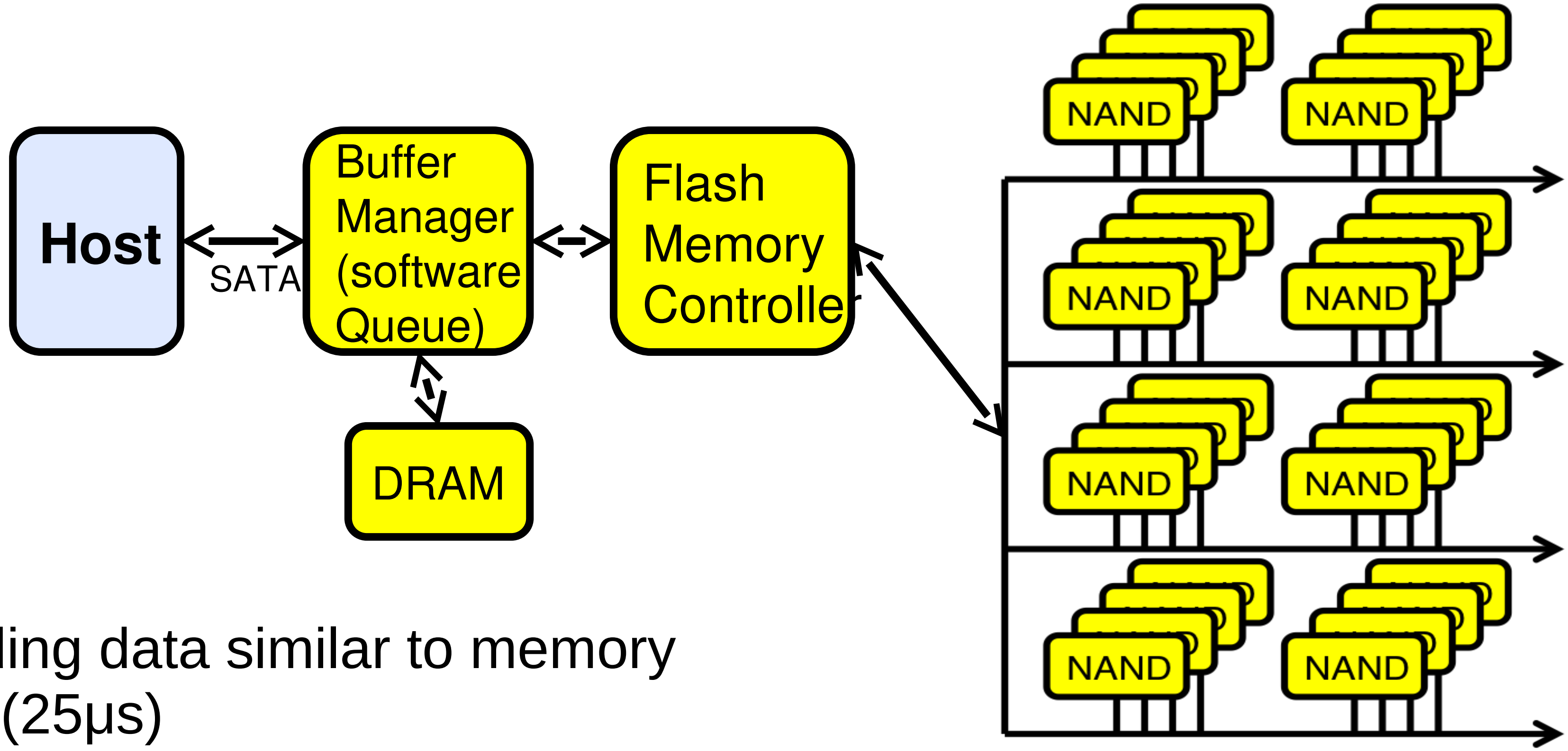
Solid State Disks (SSDs)

- Flash memory
  - Electronic non-volatile computer memory storage medium that can be electrically erased and reprogrammed.
  - Two main types of flash memory, NOR flash and NAND flash.
- Invented at Toshiba in 1980 and is based on EEPROM technology.
  - EPROMs had to be erased completely before they could be rewritten.
  - NAND flash memory can be erased, written, and read in blocks (or pages); much smaller than the entire device.
- NAND flash architecture
  - Hierarchical structure: strings, pages, blocks, planes and a die.
    - String is 32-128 NAND cells
    - It still is EEPROM: block first cleared then we can reprogram the pages!

Solid State Disks (SSDs)

- 2009 – Use NAND Multi-Level Cell (2-bit/cell) flash memory
  - Sector (4 KB page) addressable, but stores 4-64 “pages” per memory block
- No moving parts (no rotate/seek motors)
  - Eliminates seek and rotational delay (0.1-0.2ms access time)
  - Very low power and lightweight

SSD Architecture – Reads

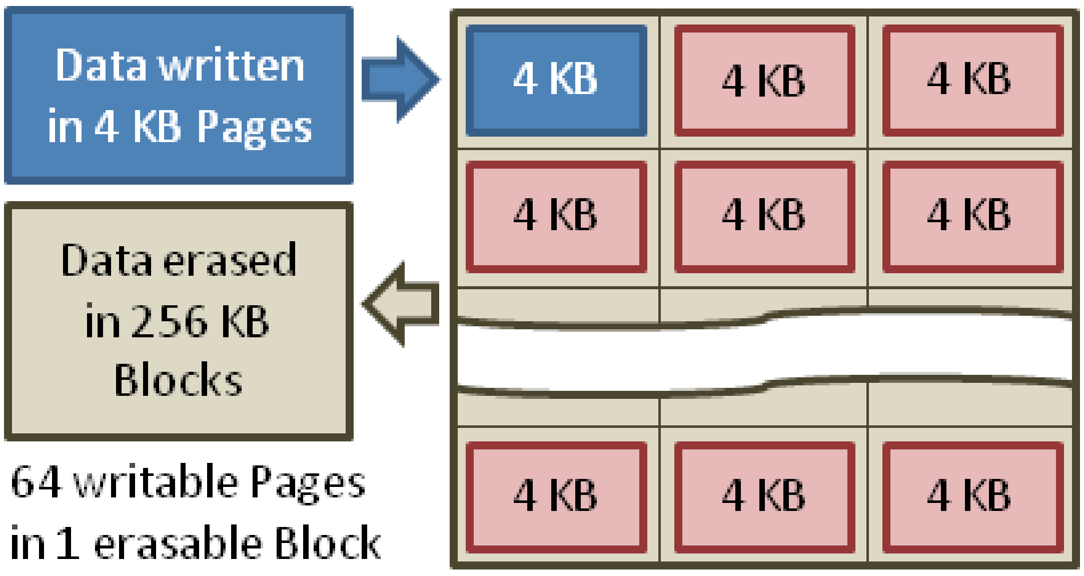


Reading data similar to memory read (25μs)

- No seek or rotational latency
- Transfer time: transfer a 4KB page
  - SATA: 300-600MB/s => ~4 x103 b / 400 x 106 bps => 10 μs
- Latency = Queuing Time + Controller time + Xfer Time
- Highest Bandwidth: Sequential OR Random reads

SSD Architecture – Writes (I)

- Writing data is complex! (~200μs – 1.7ms)
  - Erasing a block takes ~1.5ms
  - Controller maintains pool of empty blocks by coalescing used pages (read, erase, write), also reserves some % of capacity.



Typical NAND Flash Pages and Blocks

Development of SSD

- Important numbers (2020-21)
  - Seq. read/write
  - IOPS
  - Access time!

SSD evolution

Parameter	Started with	Developed to	Improvement
Capacity	20 MB (Sandisk, 1991)	100 TB (Enterprise Nimbus Data DC100, 2018) (As of 2020 Up to 8 TB available for consumers)[16]	5-million-to-one[17] (400,000-to-one[17])
Sequential read speed	49.3 MB/s (Samsung MCAQE32G5APP-0XA, 2007)[18]	15 GB/s (Gigabyte demonstration, 2019) (As of 2020 up to 6.795 GB/s available for consumers)[19]	304.25-to-one[20] (138-to-one)[21]
Sequential write speed	80 MB/s (Samsung enterprise SSD, 2008)[22][23]	15.200 GB/s (Gigabyte demonstration, 2019) (As of 2020 up to 4.397 GB/s available for consumers)[19]	190-to-one[24] (55-to-one)[25]
IOPS	79 (Samsung MCAQE32G5APP-0XA, 2007)[18]	2,500,000 (Enterprise Micron X100, 2019) (As of 2020 up to 736,270 read IOPS and 702,210 write IOPS available for consumers)[19]	31,645.56-to-one[26] (Consumer: read IOPS: 9,319.87-to-one,[27] write IOPS: 8,888.73-to-one)[28]
Access time (in milliseconds, ms)	0.5 (Samsung MCAQE32G5APP-0XA, 2007)[18]	0.045 read, 0.013 write (lowest values, WD Black SN850 1TB, 2020)[29][19]	Read:11-to-one,[30] Write: 38-to-one[31]
Price	US$50,000 per gigabyte (Sandisk, 1991)[32]	US$0.10 per gigabyte (Crucial MX500, July 2020)[33]	555,555-to-one[34]

Some “Current” 3.5in SSDs

- Seagate Nytro SSD: 15TB (2017)
  - Dual 12Gb/s interface
  - Seq reads 860MB/s
  - Seq writes 920MB/s
  - Random Reads (IOPS): 102K
  - Random Writes (IOPS): 15K
  - Price (Amazon): $6325 ($0.41/GB)
- Nimbus SSD: 100TB (2019)
  - Dual port: 12Gb/s interface
  - Seq reads/writes: 500MB/s
  - Random Read Ops (IOPS): 100K
  - Unlimited writes for 5 years!
  - Price: ~ $50K? ($0.50/GB)

some of the best disks

Seagate Nytro SSD (2022)

Specifications	Nytro 5550H 15 mm — Mixed Use		
Capacity	6.4TB	3.2TB	1.6TB
Standard Model[1]	XP6400LE70005	XP3200LE70005	XP1600LE70005
SED Model[1]	XP6400LE70015	XP3200LE70015	XP1600LE70015
FIPS 140-3/Common Criteria Model[1]	XP6400LE70025	XP3200LE70025	XP1600LE70025
Features			
Interface	PCIe® Gen4 ×4 NVMe	PCIe® Gen4 ×4 NVMe	PCIe® Gen4 ×4 NVMe
NAND Flash Type	3D eTLC	3D eTLC	3D eTLC
Form Factor	2.5 in × 15mm	2.5 in × 15mm	2.5 in × 15mm
Performance			
Sequential Read (MB/s) Sustained, 128 KB[2]	7,400	7,400	7,400
Sequential Write (MB/s) Sustained, 128 KB[2]	7,200	6,900	4,300
Random Read (IOPS) Sustained, 4 KB QD64[3]	1,700,000	1,700,000	1,700,000
Random Write (IOPS) Sustained, 4 KB QD64[3]	470,000	470,000	315,000
Average Read Latency (µs), 4 KB QD1	75	75	75
Average Write Latency (µs), 4 KB QD1	12	12	12

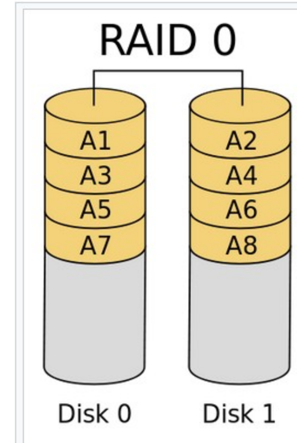
Nimbus ExaDrive DC (2022)

	EDDCT016	EDDCT032	EDDCT050	EDDCT100	EDDCS016	EDDCS032	EDDCS050	EDDCS100
Basics								
Capacity	16 TB	32 TB	50 TB	100 TB	16 TB	32 TB	50 TB	100 TB
Interface	SATA-3 (6.0 Gbps)				SAS-2 dual-port (for HA)			
Form Factor	3.5" (LFF)							
Reliability								
Endurance	Unlimited DWPD for 5 years							
MTBF (hours)	2.5 million hours							
Limited Warranty	5 years							
Performance								
Latency	0.1 ms	0.1 ms	0.1 ms	0.05 ms	0.2 ms	0.2 ms	0.2 ms	0.15 ms
Random Read (4 KB)	97K IOps	97K IOps	97K IOps	114K IOps	50K IOps	50K IOps	50K IOps	52K IOps
Random Write (4 KB)	91K IOps	91K IOps	91K IOps	106K IOps	25K IOps	25K IOps	25K IOps	26K IOps
Sequential Read	500 MBps	500 MBps	500 MBps	500 MBps	450 MBps	450 MBps	450 MBps	450 MBps
Sequential Write	460 MBps	460 MBps	460 MBps	460 MBps	260 MBps	260 MBps	260 MBps	260 MBps
Power								
Active Read Power	12.1 W	12.2 W	12.1 W	15.2 W	12.1 W	12.2 W	12.1 W	15.2 W
Active Write Power	13.1 W	13.2 W	13.8 W	16.8 W	13.1 W	13.2 W	13.8 W	16.8 W
Idle Power	6.8 W	7.2 W	7.2 W	11.1 W	7.0 W	7.4 W	7.4 W	11.3 W
Active Read Power / TB	0.76 W	0.38 W	0.24 W	0.15 W	0.76 W	0.38 W	0.24 W	0.15 W
Active Write Power / TB	0.82 W	0.41 W	0.28 W	0.17 W	0.82 W	0.41 W	0.28 W	0.17 W
Idle Power / TB	0.43 W	0.23 W	0.14 W	0.11 W	0.44 W	0.23 W	0.14 W	0.11 W

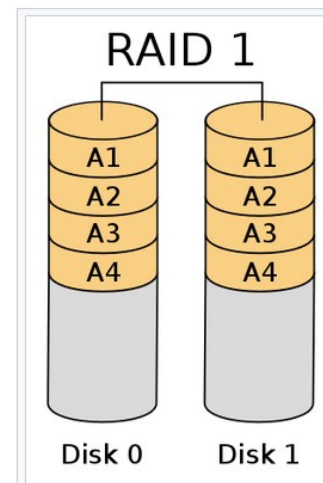
RAID

- Disk Array: Arrangement of several disks that gives abstraction of a single, large disk.
- Goals: Increase performance and reliability.
- Two main techniques:
  - Data striping: Data is partitioned; size of a partition is called the striping unit. Partitions are distributed over several disks.
  - Redundancy: More disks => more failures. Redundant information allows reconstruction of data if a disk fails.

RAID levels



- Level 0: No redundancy
  - Data distributed in strips
  - No redundancy, parity
  - Speed is the only reason

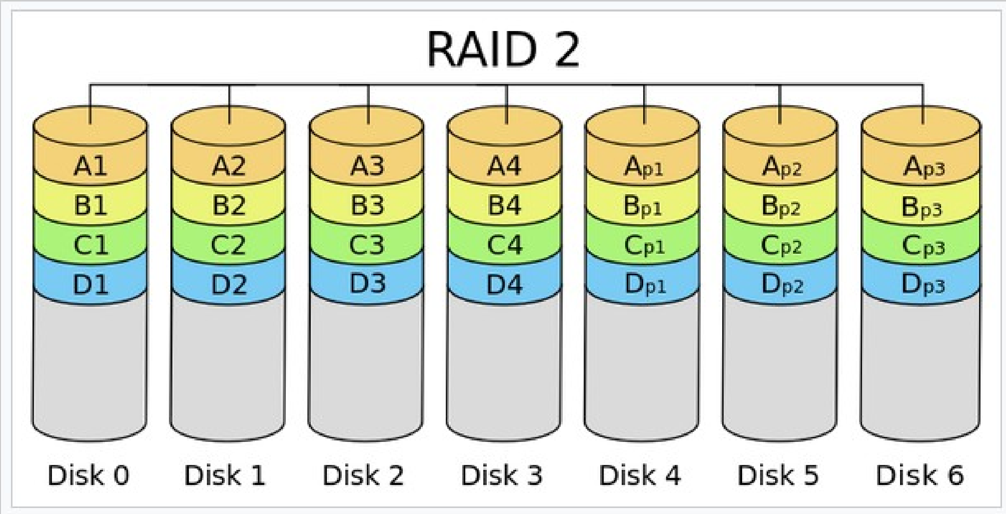


- Level 1: Mirrored (two identical copies)
  - Each disk has a mirror image (check disk)
  - Parallel reads, a write involves two disks.
  - Max.transfer rate = transfer rate of one disk
    - N mirrored disks => N times access to one

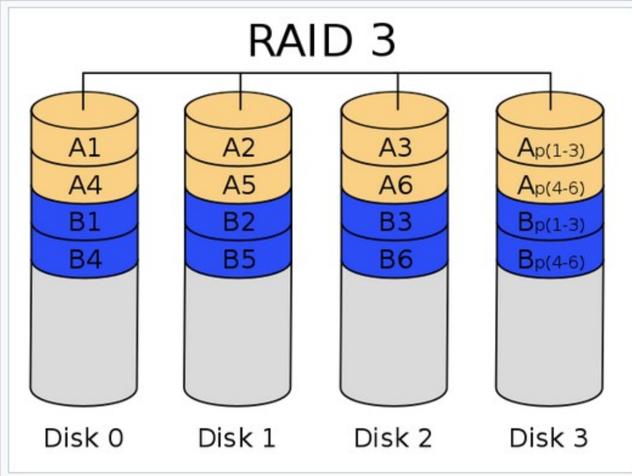
•

RAID levels

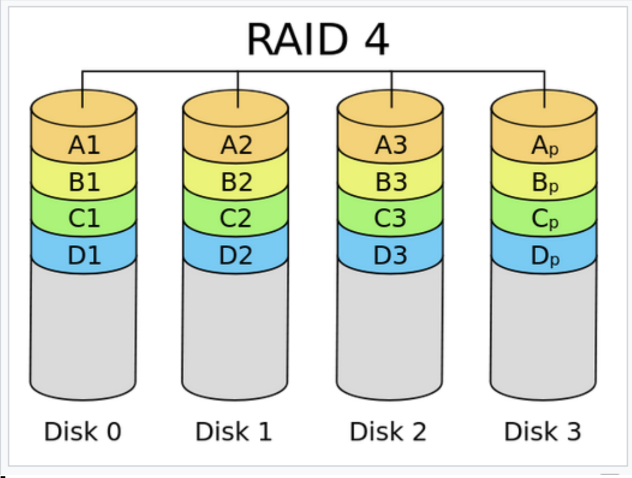
- Level 2 (0+1): Striping and Mirroring
  - Striping unit is 1 bit
  - Hamming code for error correction
  - Parallel reads, a write involves two disks.
  - Maximum transfer rate = aggregate bandwidth
  - Rearly used



RAID levels



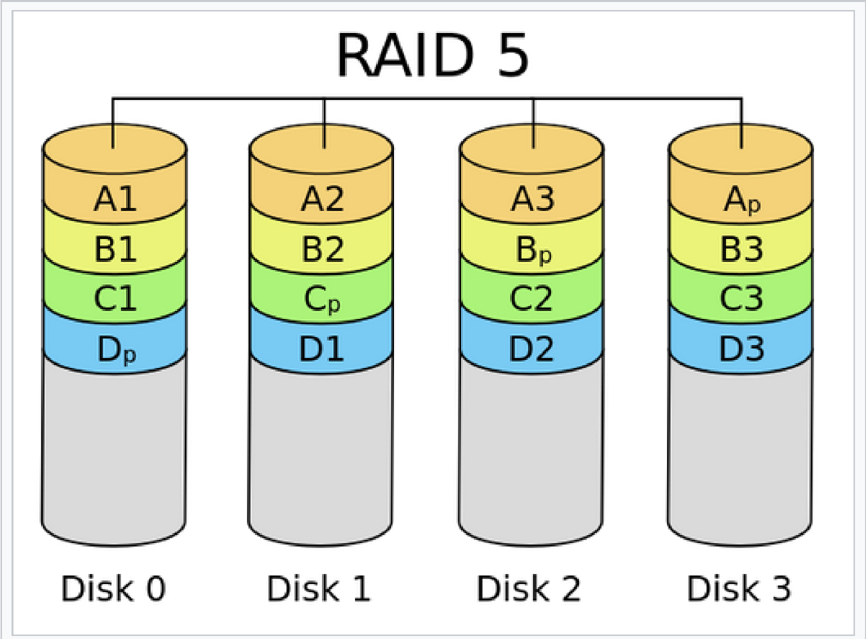
- Level 3: Bit-interleaved parity
  - Striping Unit: One byte.
  - One parity disk.
  - Each read and write request involves all disks
  - Disk array can process one request at a time



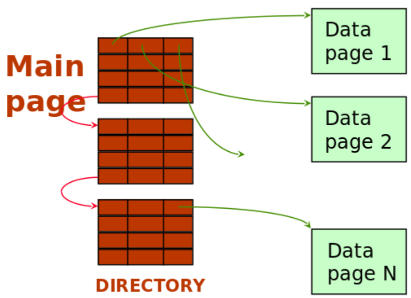
- Level 4: Block-interleaved parity
  - Striping Unit: One disk block.
  - One check disk.
  - Parallel reads possible for small requests
  - Large requests can utilize full bandwidth
  - Writes involve modified block and check disk

RAID levels

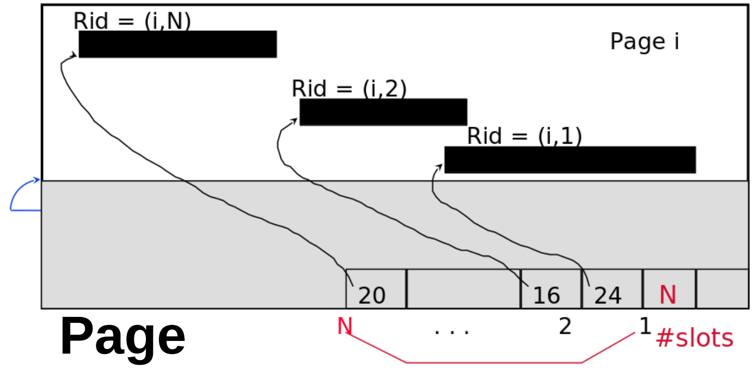
- Level 5: Block-Interleaved Distributed Parity
  - Similar to RAID Level 4, but parity blocks are distributed over all disks



DBMS memory hieararchy

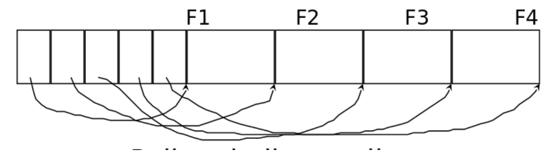


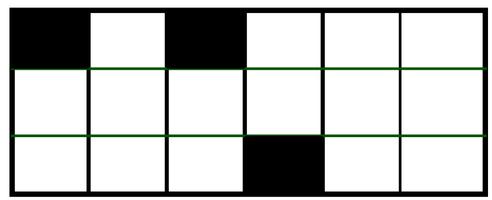
File



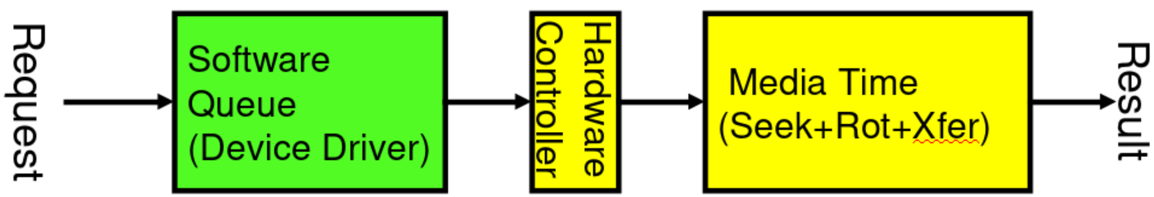
Page

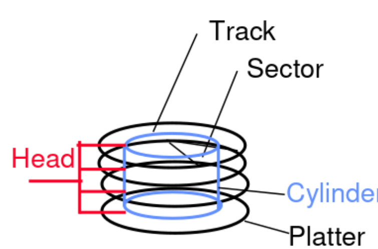
Record





Buffer pool





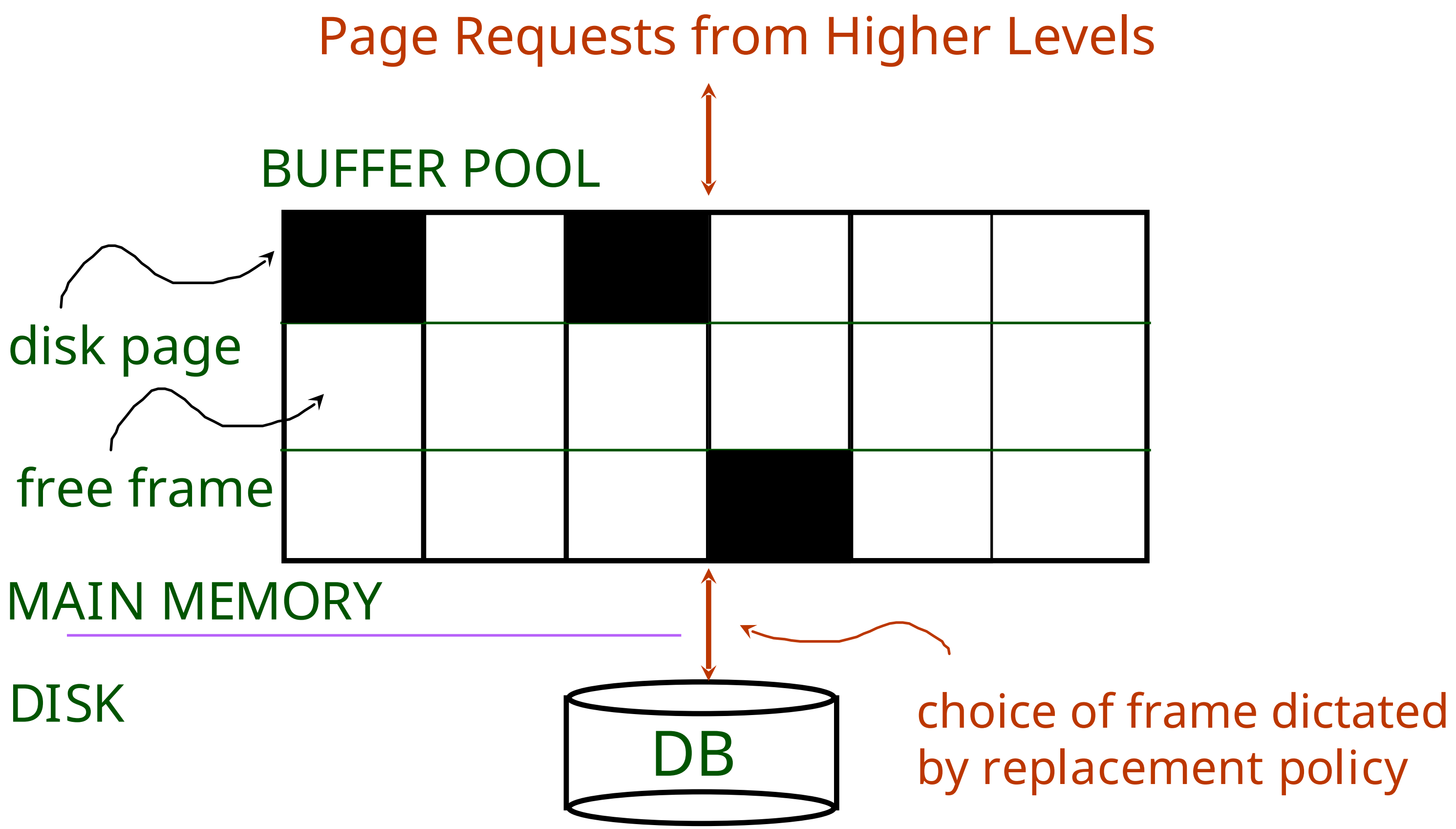
Arranging Pages on Disk

- *`Next'* block concept:
  - blocks on same track, followed by
  - blocks on same cylinder, followed by
  - blocks on adjacent cylinder
- Blocks in a file should be arranged sequentially on disk (by `next'), to minimize seek and rotational delay.
- For a sequential scan, *pre-fetching* several pages at a time is a big win!

Disk Space Management

- Lowest layer of DBMS software manages space on disk.
- Higher levels call upon this layer to:
  - allocate/de-allocate a page
  - read/write a page
- Request for a *sequence* of pages must be satisfied by allocating the pages sequentially on disk! Higher levels don't need to know how this is done, or how free space is managed.

Buffer Management in a DBMS



- *Data must be in RAM for DBMS to operate on it!*
- *Table of <frame#, pageid> pairs is maintained.*

When a Page is Requested ...

- If requested page is not in pool:
  - Choose a frame for *replacement*
  - If frame is dirty, write it to disk
  - Read requested page into chosen frame
- *Pin* the page and return its address.

☛ *If requests can be predicted (e.g., sequential scans) pages can be pre-fetched several pages at a time!*

More on Buffer Management

- Requestor of page must unpin it, and indicate whether page has been modified:
  - *dirty* bit is used for this.
- Page in pool may be requested many times,
  - a *pin count* is used. A page is a candidate for replacement iff *pin count* = 0.
- CC & recovery may entail additional I/O when a frame is chosen for replacement. (*Write-Ahead Log* protocol; more later.)

Buffer Replacement Policy

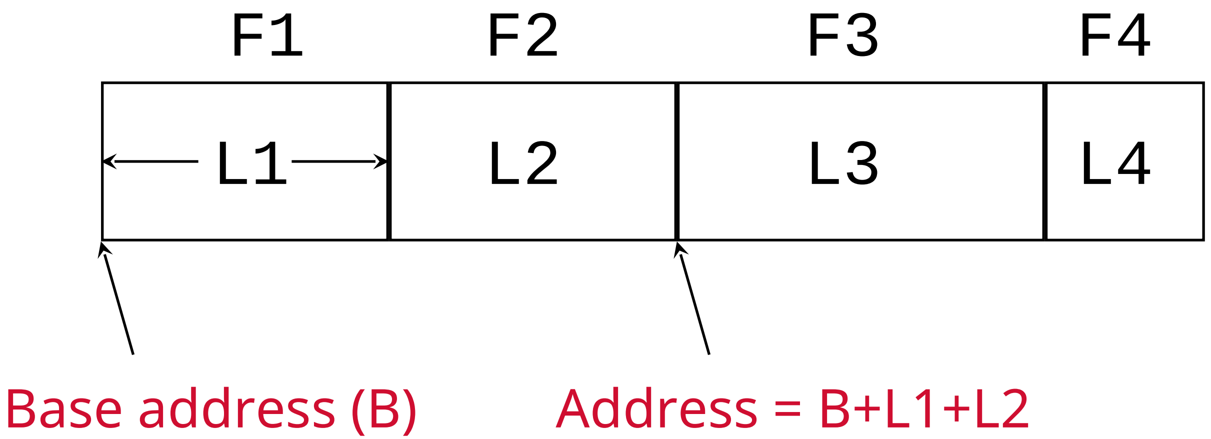
- Frame is chosen for replacement by a *replacement policy:*
  - Least-recently-used (LRU), Clock, MRU etc.
- Policy can have big impact on # of I/O's; depends on the *access pattern*.
- *Sequential flooding*: Nasty situation caused by LRU + repeated sequential scans.
  - # buffer frames < # pages in file means each page request causes an I/O. MRU much better in this situation (but not in all situations, of course).

DBMS vs. OS File System

OS does disk space & buffer mgmt: why not let OS manage these tasks?

- Differences in OS support: portability issues
- Some limitations, e.g., files can't span disks.
- Buffer management in DBMS requires ability to:
  - pin a page in buffer pool, force a page to disk (important for implementing CC & recovery),
  - adjust *replacement policy,* and pre-fetch pages based on access patterns in typical DB operations.

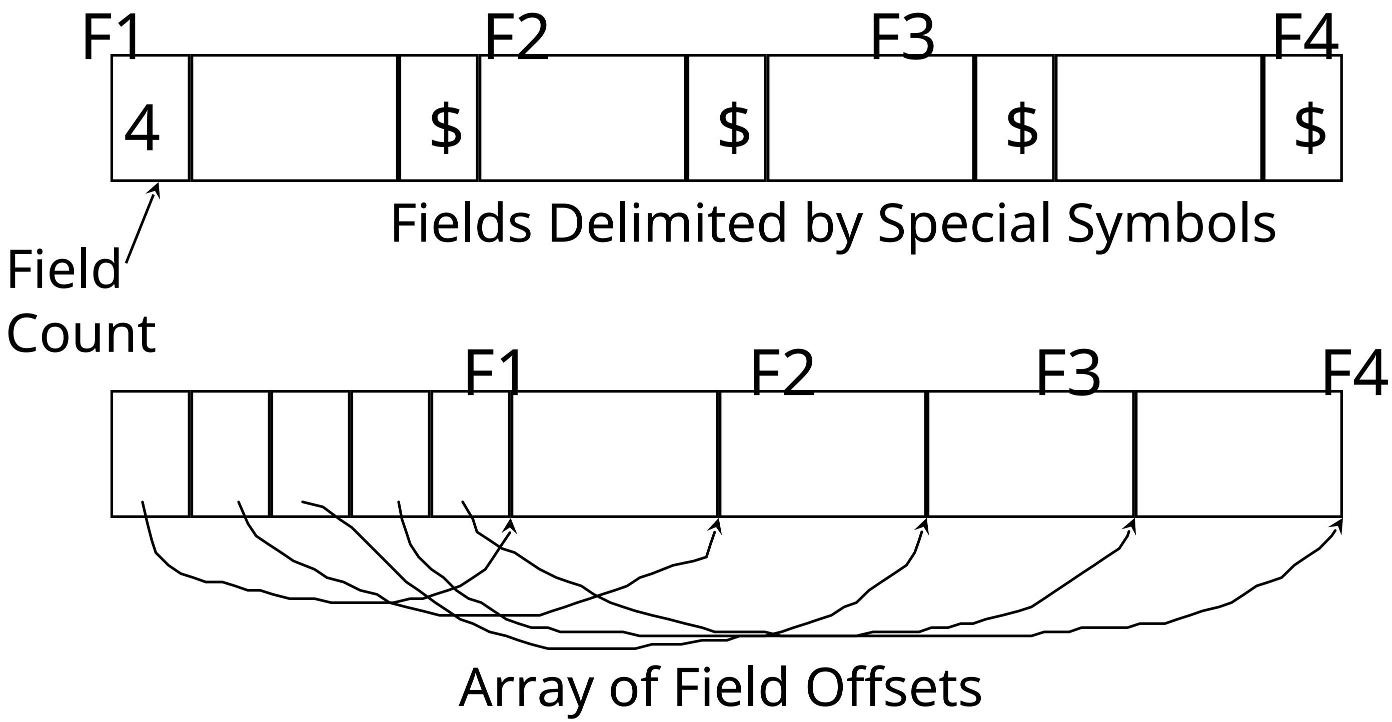
Record Formats: Fixed Length



- Information about field types same for all records in a file; stored in *system catalogs.*
- Finding *i'th* field does not require scan of record.

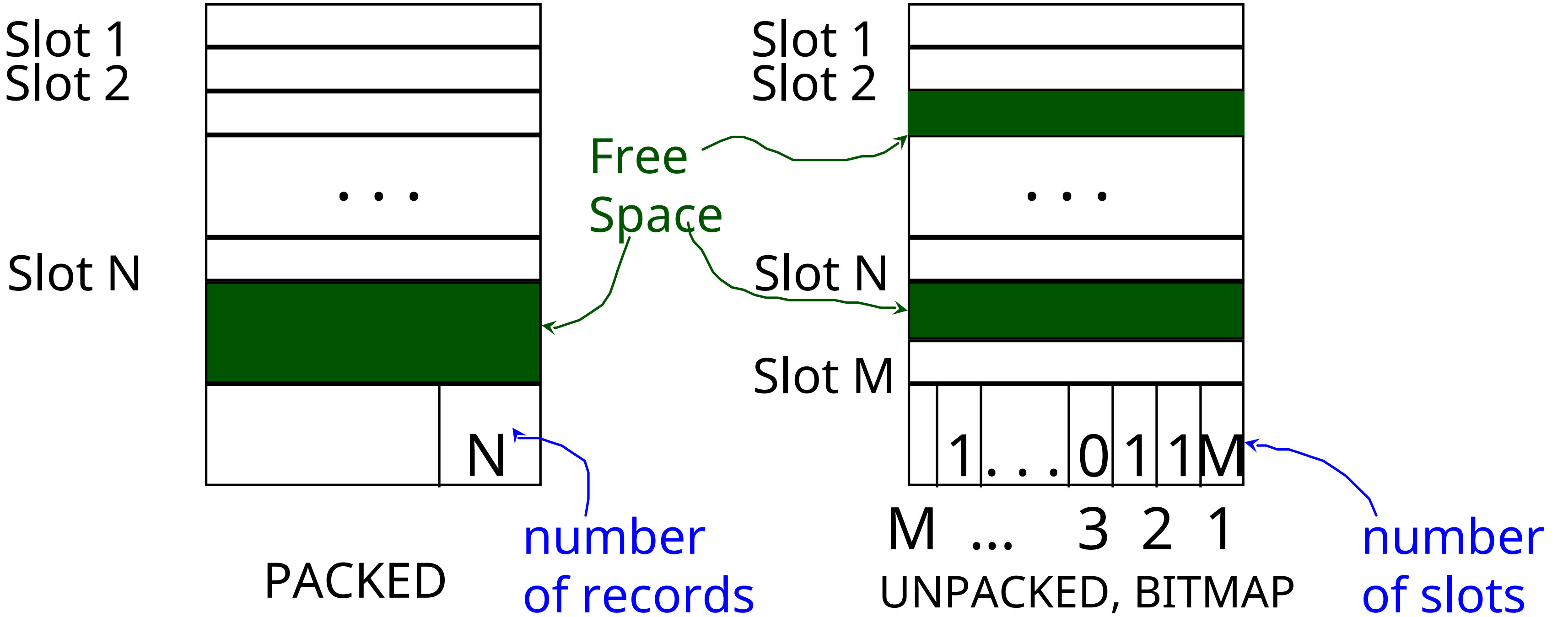
Record Formats: Variable Length

- Two alternative formats (# fields is fixed):



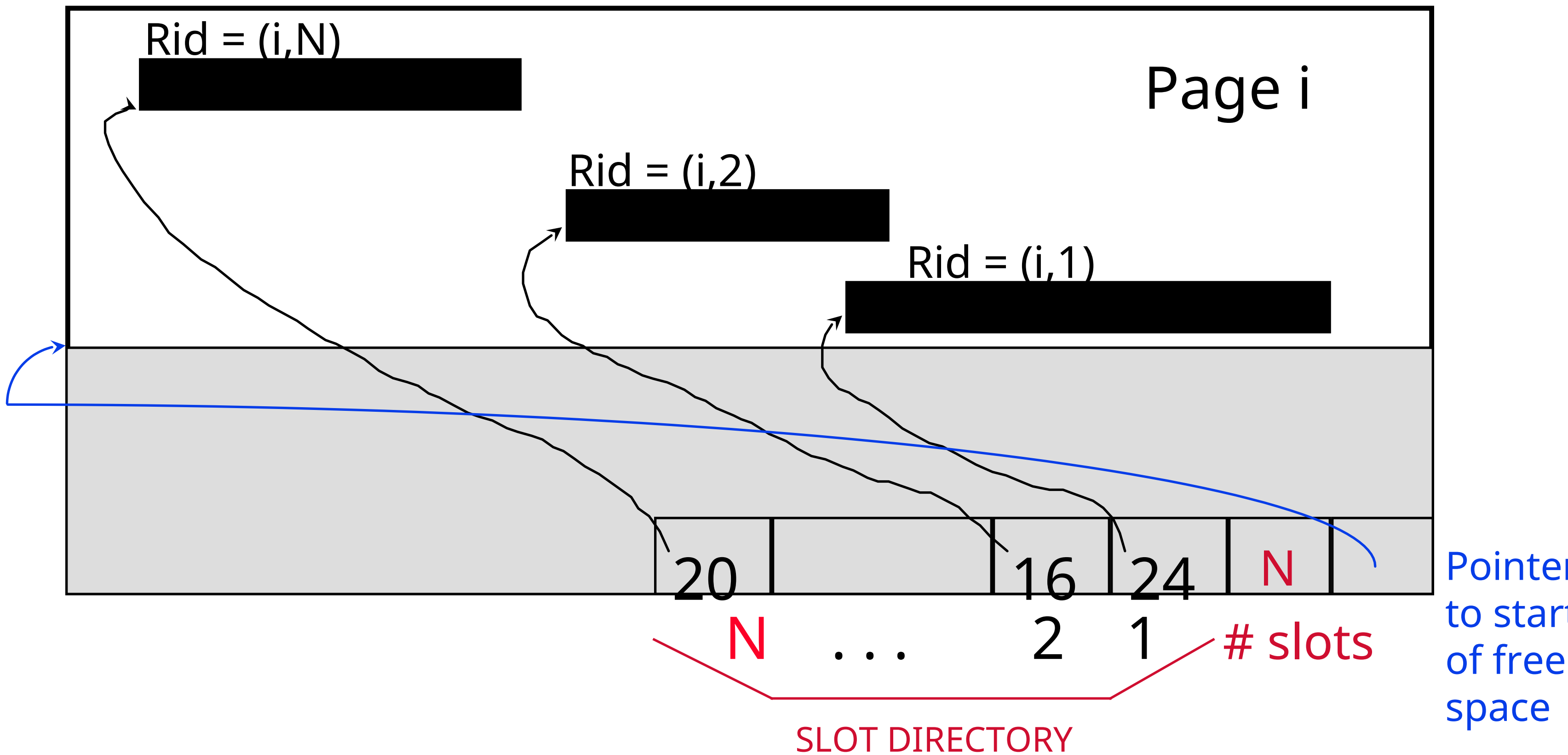
☛ Second offers direct access to i'th field, efficient storage of *nulls* (special *don't know* value); small directory overhead.

Page Formats: Fixed Length Records



- *Record id = <page id, slot #>. In first alternative, moving records for free space management changes rid; may not be acceptable.*

Page Formats: Variable Length Records



- *Can move records on page without changing rid; so, attractive for fixed-length records too.*

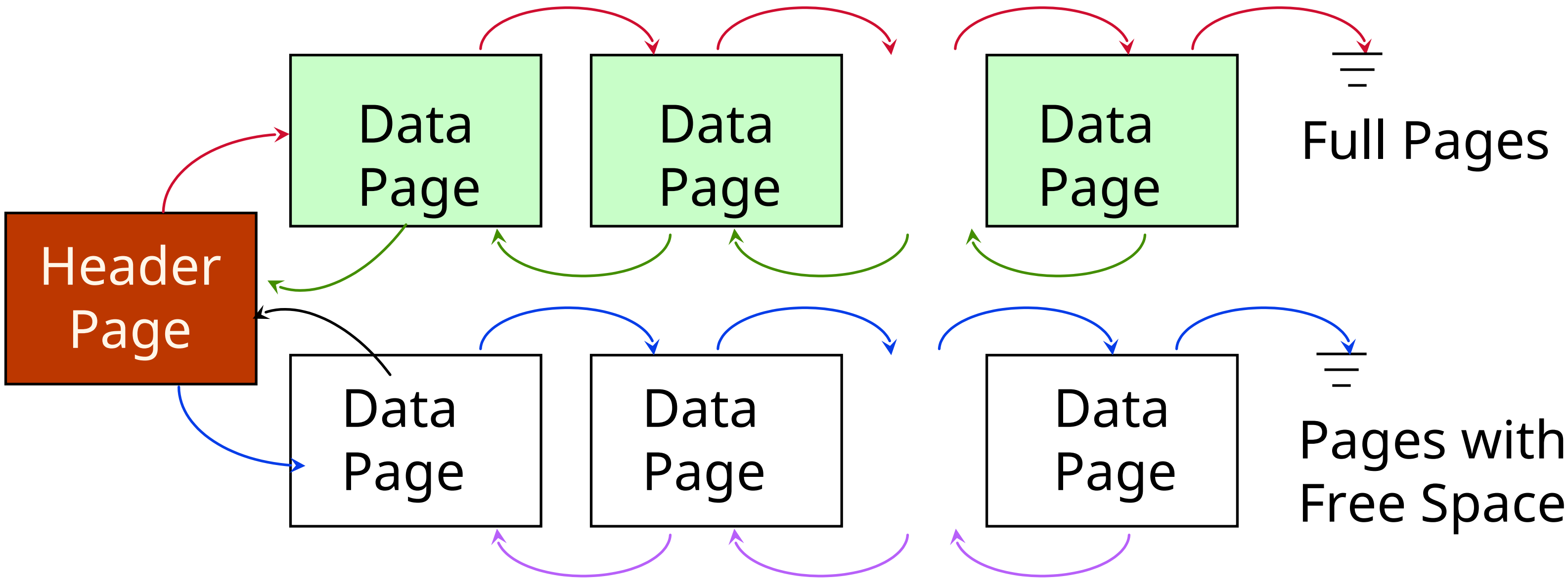
Files of Records

- Page or block is OK when doing I/O, but higher levels of DBMS operate on *records*, and *files of records*.
- FILE: A collection of pages, each containing a collection of records. Must support:
  - insert/delete/modify record
  - read a particular record (specified using *record id*)
  - scan all records (possibly with some conditions on the records to be retrieved)

Unordered (Heap) Files

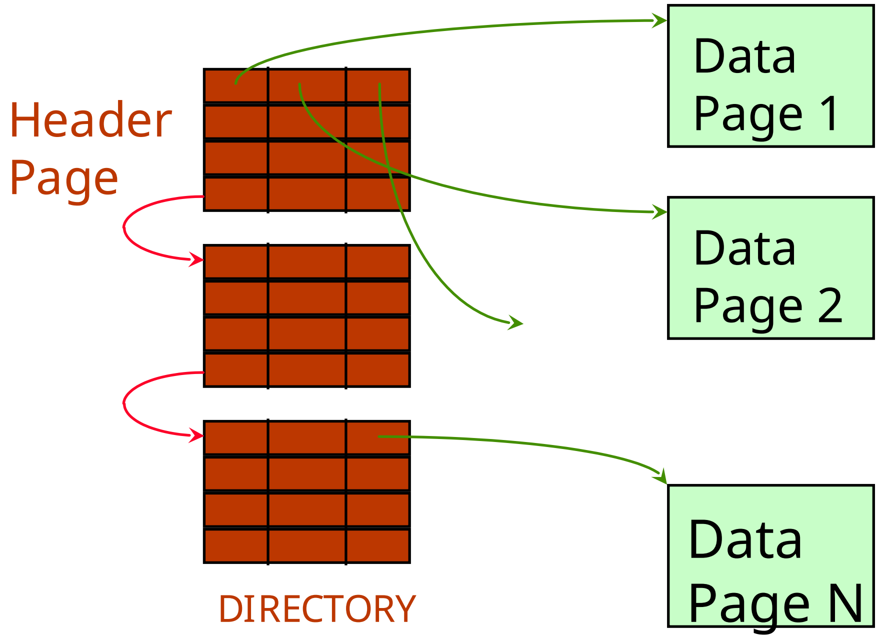
- Simplest file structure contains records in no particular order.
- As file grows and shrinks, disk pages are allocated and de-allocated.
- To support record level operations, we must:
  - keep track of the *pages* in a file
  - keep track of *free space* on pages
  - keep track of the *records* on a page
- There are many alternatives for keeping track of this.

Heap File Implemented as a List



- The header page id and Heap file name must be stored someplace.
- Each page contains 2 `pointers' plus data.

Heap File Using a Page Directory



- The entry for a page can include the number of free bytes on the page.
- The directory is a collection of pages; linked list implementation is just one alternative.
  - *Much smaller than linked list of all HF pages*!

System Catalogs

- For each index:
  - structure (e.g., B+ tree) and search key fields
- For each relation:
  - name, file name, file structure (e.g., Heap file)
  - attribute name and type, for each attribute
  - index name, for each index
  - integrity constraints
- For each view:
  - view name and definition
- Plus statistics, authorization, buffer pool size, etc.

☛ *Catalogs are themselves stored as relations*!

Attr_Cat(attr_name, rel_name, type, position)

attr_name	rel_name	type	position
attr_name	Attribute_Cat	string	1
rel_name	Attribute_Cat	string	2
type	Attribute_Cat	string	3
position	Attribute_Cat	integer	4
sid	Students	string	1
name	Students	string	2
login	Students	string	3
age	Students	integer	4
gpa	Students	real	5
fid	Faculty	string	1
fname	Faculty	string	2
sal	Faculty	real	3

Summary

- Disks provide cheap, non-volatile storage.
  - Random access, but cost depends on location of page on disk; important to arrange data sequentially to minimize *seek* and *rotation* delays.
- Buffer manager brings pages into RAM.
  - Page stays in RAM until released by requestor.
  - Written to disk when frame chosen for replacement (which is sometime after requestor releases the page).
  - Choice of frame to replace based on *replacement policy.*
  - Tries to *pre-fetch* several pages at a time.

Summary (Contd.)

- DBMS vs. OS File Support
  - DBMS needs features not found in many OS’s, e.g., forcing a page to disk, controlling the order of page writes to disk, files spanning disks, ability to control pre-fetching and page replacement policy based on predictable access patterns, etc.
- Variable length record format with field offset directory offers support for direct access to i’th field and null values.
- Slotted page format supports variable length records and allows records to move on page.

Summary (Contd.)

- File layer keeps track of pages in a file, and supports abstraction of a collection of records.
  - Pages with free space identified using linked list or directory structure (similar to how pages in file are kept track of).
- Indexes support efficient retrieval of records based on the values in some fields.
- Catalog relations store information about relations, indexes and views. (*Information that is common to all records in a given collection.*)