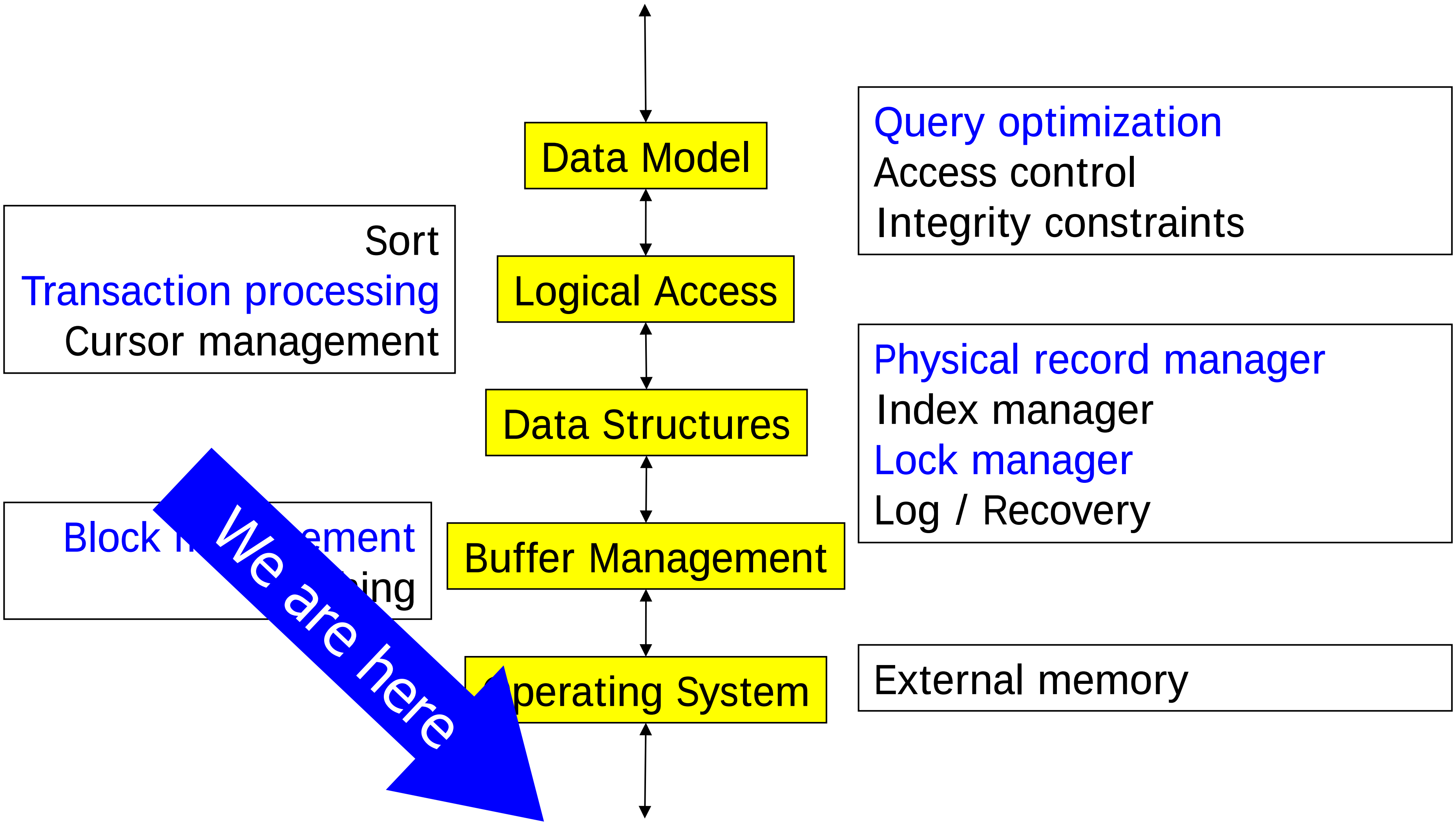
Datenbanksysteme II: Storage, Discs, and Raid

Ulf Leser

Tasks



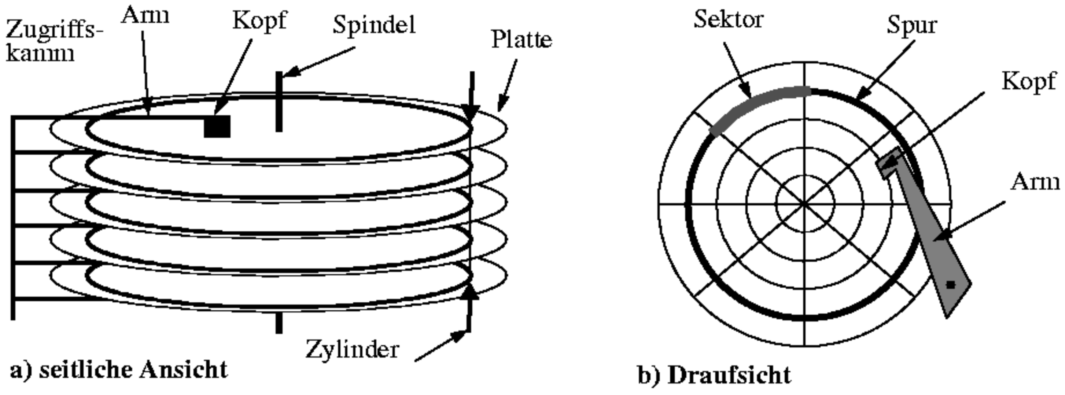
Data Model
Logical Access
Data Structures
Buffer Management
perating System
Query optimization
Access control
Integrity constraints
Sort
Transaction processing
Cursor management
Physical record manager
Index manager
Lock manager
Log / Recovery
Block
External memory
We are here

Content of this Lecture

- Discs
- RAID level
- Some guidelines

Magnetic Discs

- Preferred mass-storage since ~1970
  - Multiple rotating discs, each with a separate head
  - Discs: Tracks, sectors (blocks)
  - Formatting: Determining (fixed) block size
  - Blocks with constant size, tracks do not have constant number of blocks
- Blocks use error-correcting codes: Single bit errors can be corrected



Reading from Discs

- Seek time: t_s
  - 5-20ms: Move head to right track
- Latency time: t_r
  - 3-10ms: Wait for sector to rotate to head
  - On average: ½ rotation
  - Typical speed: 6000 - 10000 rotations / minute
- Reading blocks: At rotation speed
  - Beware caching within disc controller
- Transfer rate: u
  - Data volume read per time and put into main memory

Development

Exemplarische Entwicklung der Plattengeschwindigkeit über die Zeit

Kategorie	Jahr	Modell	Größe in GB	Drehzahl	Datenrate in MB/s	Spurwechsel	Latenz	mittlere Zugriffszeit
Server	1993	IBM 0662	1	5.400 min^{-1}	5	8,5 ms	5,6 ms	15,4 ms
Server	2002	Seagate Cheetah X15 36LP	18 – 36	15.000 min^{-1}	52 – 68	3,6 ms	2,0 ms	5,8 ms
Server	2007	Seagate Cheetah 15k.6	146 – 450	15.000 min^{-1}	112 –171	3,4 ms	2,0 ms	5,6 ms
Desktop	1989	Seagate ST296N	0,080	3.600 min^{-1}	0,5	28 ms	8,3 ms	40 ms
Desktop	1993	Seagate Marathon 235	0,064 – 0,210	3.450 min^{-1}		16 ms	8,7 ms	24 ms
Desktop	1998	Seagate Medalist 2510–10240	2,5 – 10	5.400 min^{-1}		10,5 ms	5,6 ms	16,3 ms
Desktop	2000	IBM Deskstar 75GXP	20 – 40	5.400 min^{-1}	32	9,5 ms	5,6 ms	15,3 ms
Desktop	2009	Seagate Barracuda 7200.12	160 –1.000	7.200 min^{-1}	125	8,5 ms	4,2 ms	12,9 ms
Notebook	1998	Hitachi DK238A	3,2 – 4,3	4.200 min^{-1}	8,7 – 13,5	12 ms	7,1 ms	19,3 ms
Notebook	2008	Seagate Momentus 5400.6	120 – 500	5.400 min^{-1}	39 – 83	14 ms	5,6 ms	18 ms

Quelle: Wikipedia

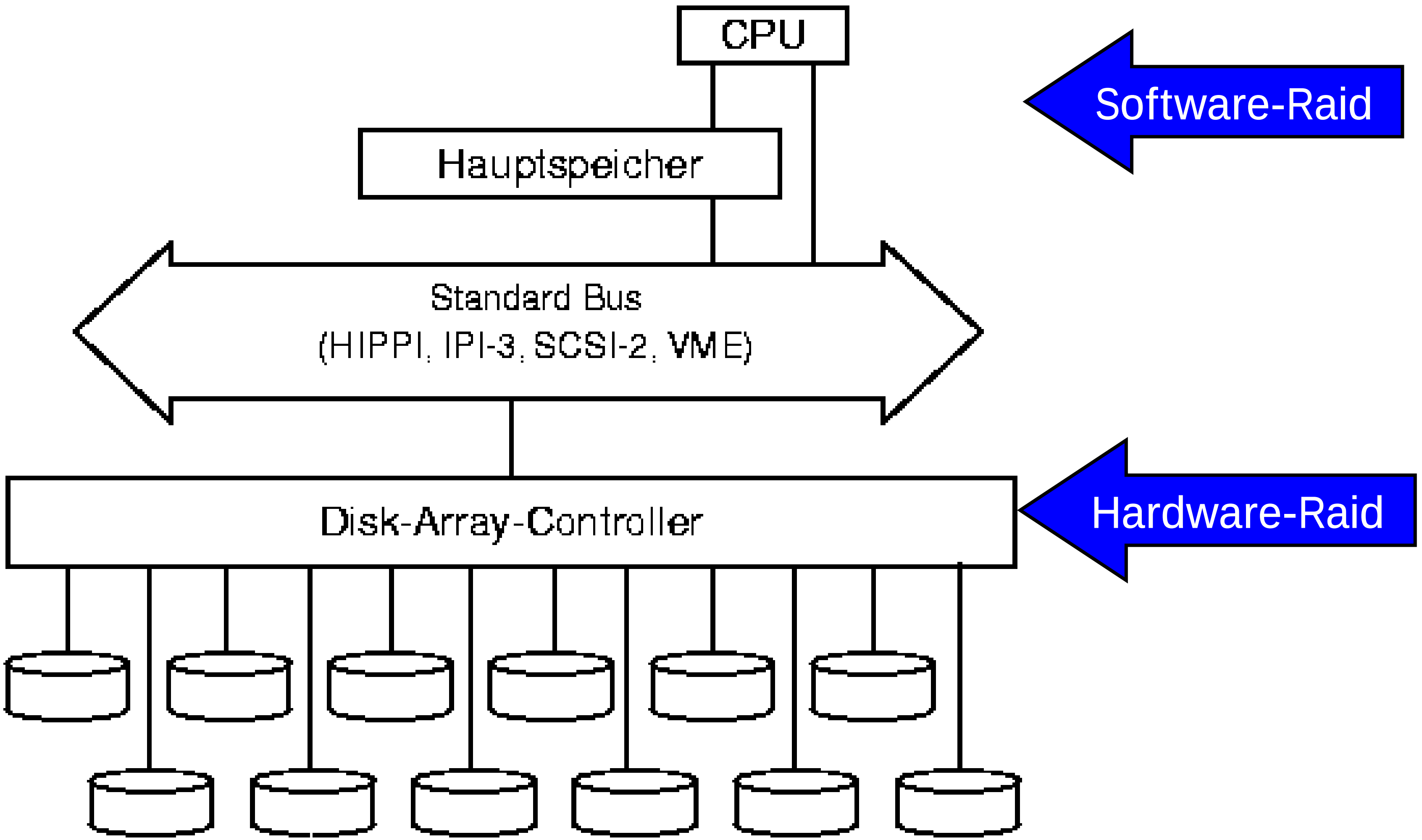
Random versus Sequential IO

- Task: Read 1000 blocks each 4KB (=4MB)
- Parameter: T_s= 5ms, T_r = 3ms, u = 15 MB/s
- Random I/O
  - For each block: seek + latency
  - t = 1000 * (5 ms + 3 ms) + 1000*4KB/15MB*1000 ms
  - t= 8000 ms + 300ms ~ 8s
- Sequential I/O
  - Once seek+latency
  - 5 ms + 3ms + 4MB/15MB*1000 ms
  - ➔ 8ms + 300 ms ~ 1/3 s
- One can read a lot sequentially before RA makes sense

How to get Faster?

- Fast IO is vital for an DBMS
  - Do not use SAN, NFS, HDFS, ...
- Parallelize storage access (read and write)
  - Distribute files over multiple disks
  - Needs proper in-between infrastructure: disc controller, memory access channels
- RAID: Redundant Array of Independent Discs
  - Or: „Redundant array of inexpensive discs“
  - Idea: Buy many yet cheap disks
    - In contrast to more expensive disk with faster rotations and less errors
  - Allows faster access (parallelization)
  - Allows higher fault tolerance (redundancy)
    - Which requires disks to be independent

Architectures



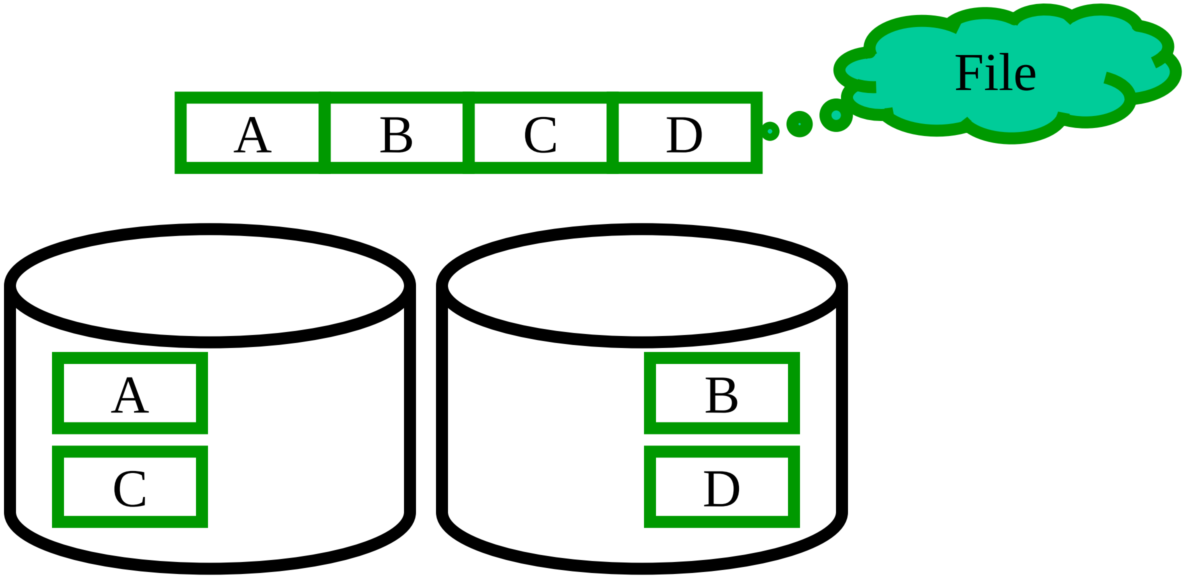
Measuring Fault Tolerance

- One disc: If a head crashes, disk is gone
- With n non-redundant disks
  - Let p be the average number of day until a disk crashes
    - When will a disk fail (one is enough for data loss)?
    - If bought at the same time - after ~p days – all “at once”
  - Let p be the probability per day that a disk crashes
    - What is the probability per day that at least one disk chrashes?
    - $1-(1-p)^n$
- If we introduce redundancy, probability of faults changes
- So does latency, read throughput, write throughput, and net space

Content of this Lecture

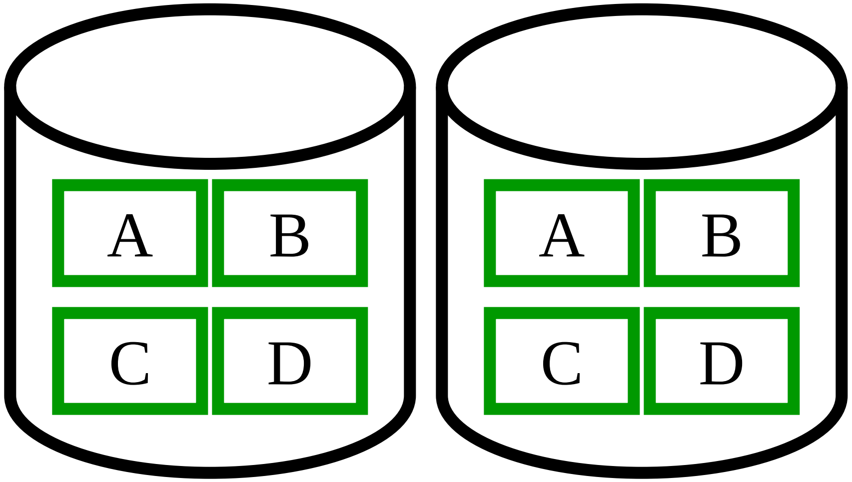
- Discs
- RAID level
- Some guidelines

RAID 0: Striping



- Doubled throughput for sequential file reads and writes
  - Assuming files being perfectly distributed
- Short files are not accelerated much
  - Seek+latency times dominate
- Decreased fault tolerance

RAID 1: Mirroring

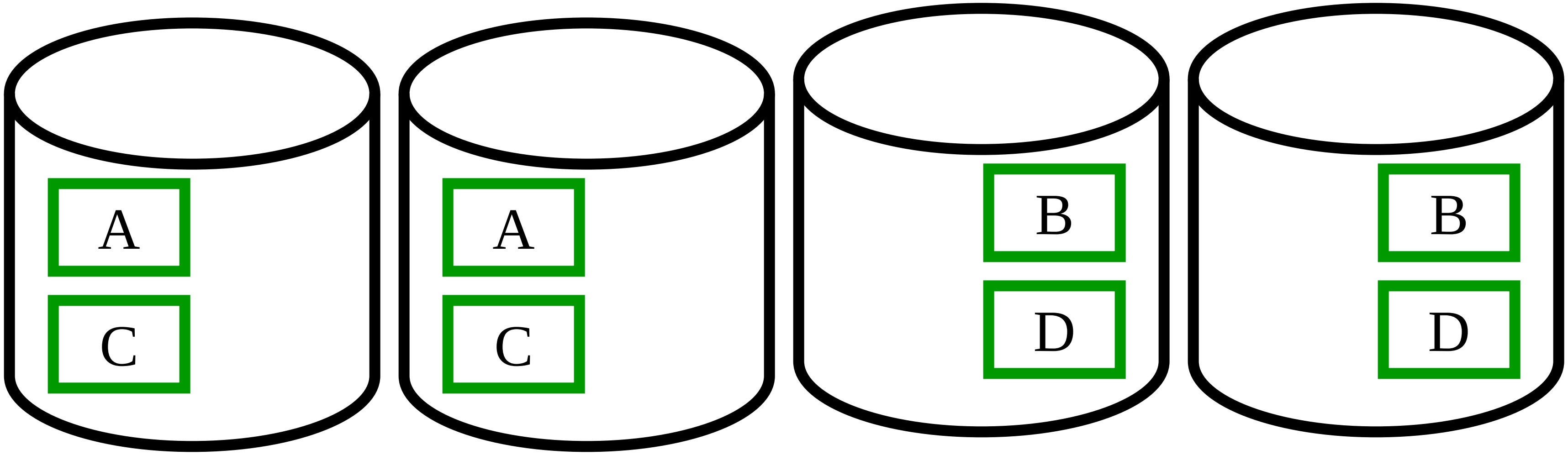


- 50% space lost
- Doubled throughput for sequential file reads
- Writes are not accelerated
- Single block read might be slightly better
  - Read form both disks, faster disk wins
- Increased fault tolerance

RAID0 versus RAID1

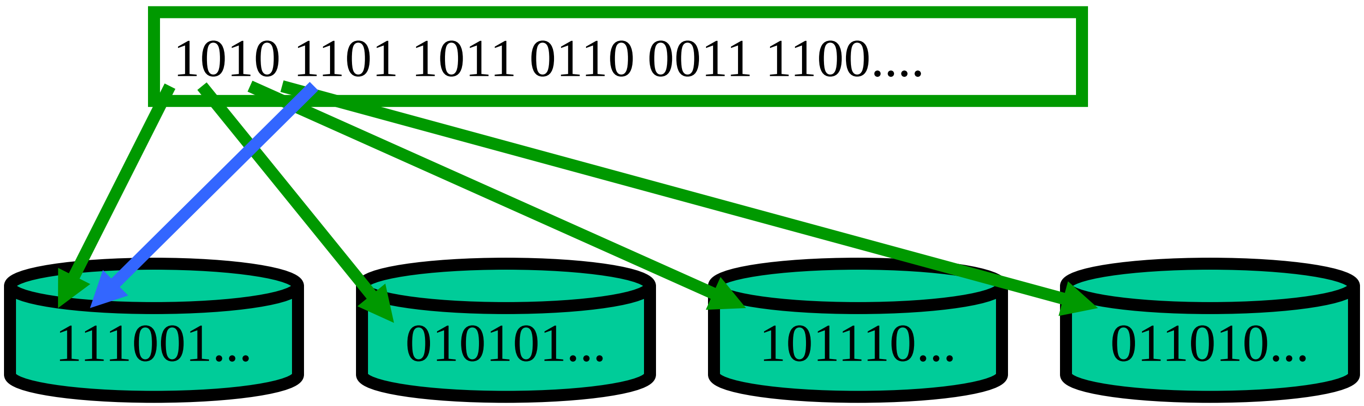
- Some concepts
  - MTTF = Mean time to failure
  - MTTDL = Mean time to data loss (fatal crash)
    - Data needs to be restored from backup
- Example: MTTF = 3650 days
  - RAID0 with 2 disks bought at arbitrary points in time
    - $MTTDL_1$ = 3650/2 = 1825 days
  - RAID1 with 2 disks bought at arbitrary points in time
    - $MTTDL_2 = MTTDL_1 * MTTDL_1$ ~ 9.000 years
  - Assuming statistical independence of events (disks)
  - But: Shared room (fire, flood), shared power (outage), shared building (earthquake), shared age, ...

RAID 0+1: Striping and Mirroring



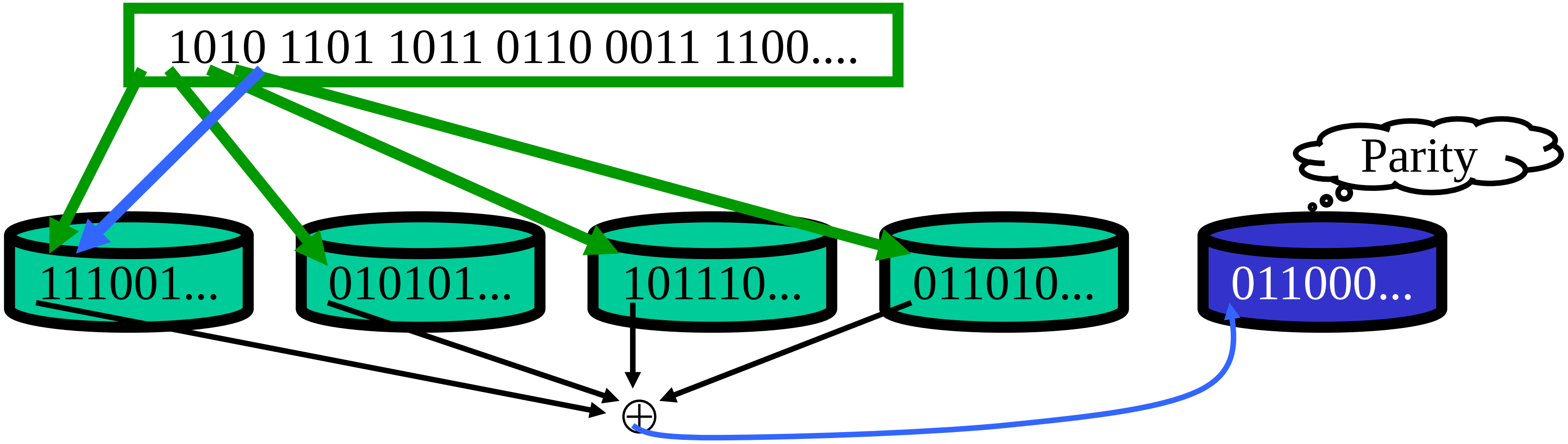
- Quadruple speed for sequential read
- Doubled speed for sequential writes
- 50% space loss
- Increased fault tolerance

RAID 2: Striping Bits (not Blocks)



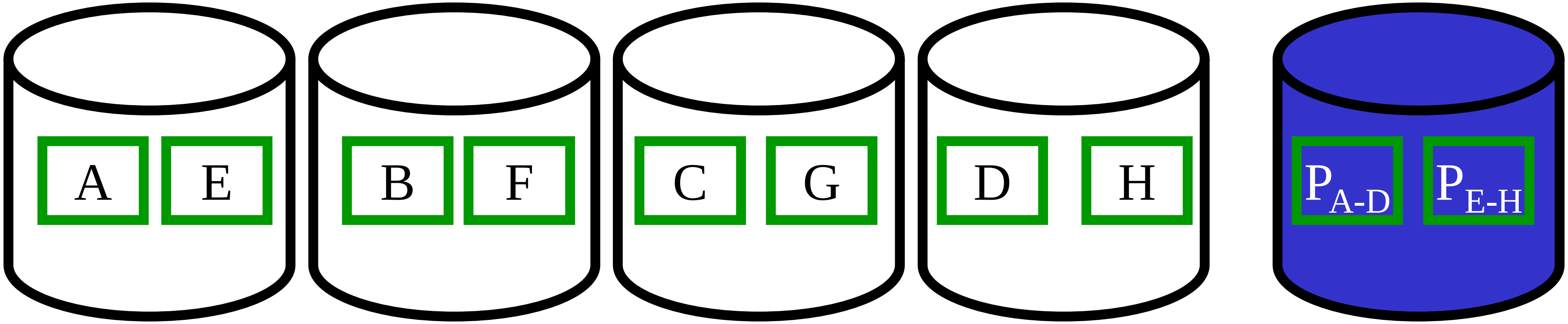
- On block devices, no advantage compared to RAID0
  - Reading a byte is as expensive as reading a block
- But more complex management
  - OS / DBs cache blocks, not parts of blocks
- Practically irrelevant

RAID 3: RAID2 + Parity



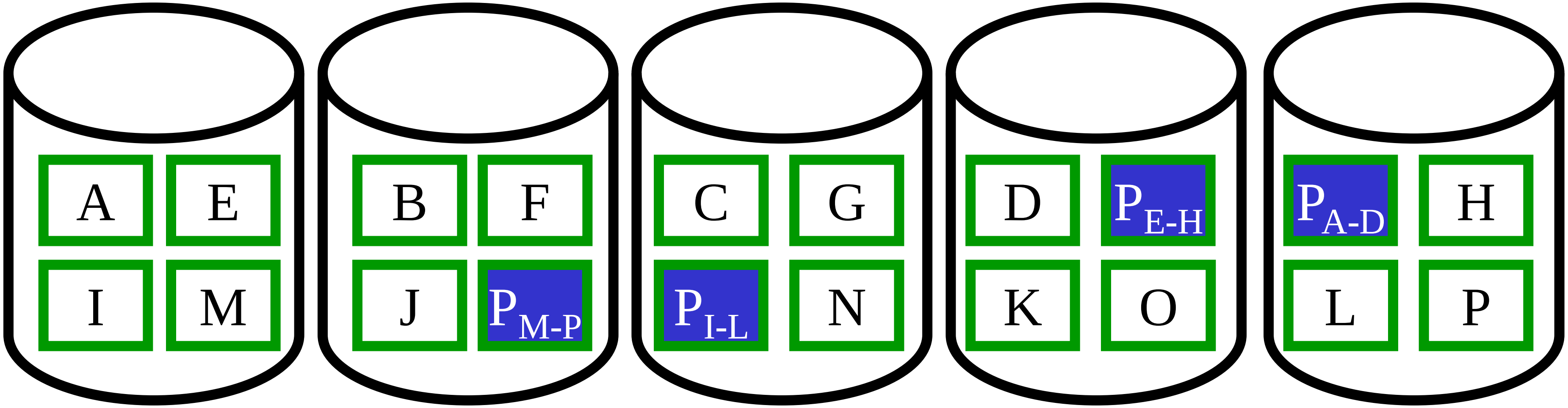
- Parity: bit-wise XOR of bits at each position
- Increased fault tolerance: One disk crash can be tolerated
  - Crashed data can be restored from other disks
  - At much better space utilization than RAID1
- (n-1) times faster throughput for sequential reads
- Writes may become even slower
  - If multiple processes write, parity disk becomes bottleneck

RAID 4: Block Striping + Parity



- Similar to RAID 3
- Easier management
- Parity still potential bottleneck
  - Writes must by synchronized: Write A,B,C,D,P_{A-D}, then B,F, ...
  - Difficult if multiple processes perform disk accesses
- Practically irrelevant

RAID 5: RAID4 with distributed Parity



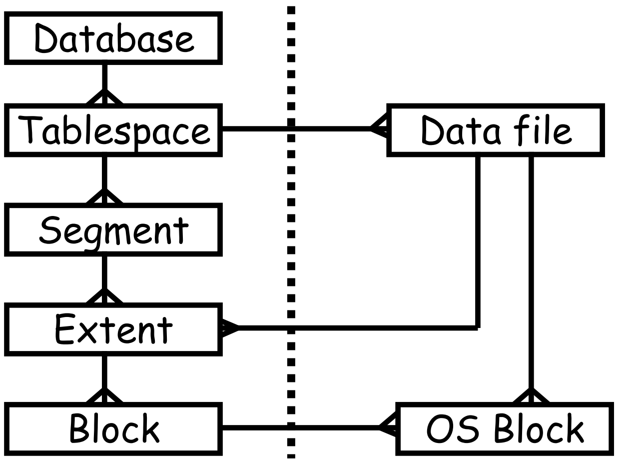
- Parity blocks are evenly spread over disks
- Writes not slowed down any more
- Many benefits
  - Much faster reads
  - Writes not affected
  - Not much space wasted
  - Disk crash can be masked

Summary

	0	1	0+1	2	3	4	5
Striping blockweise	√		√			√	√
Striping bitweise				√	√		
Kopie		√	√				
Parität				√	√	√	√
Parität dediz. Platte					√	√	
Parität verteilt							√
Erkennen mehrerer Fehler							

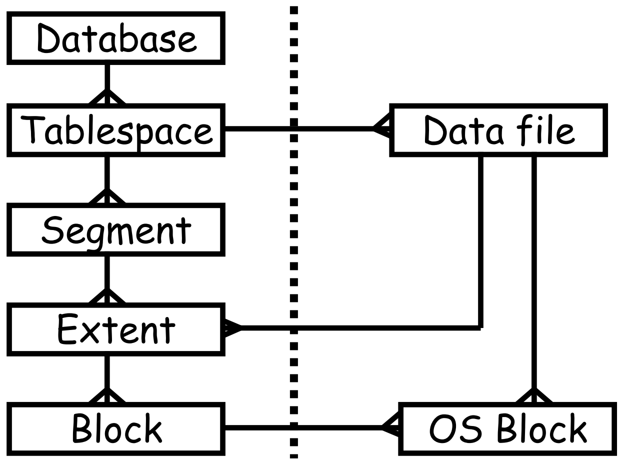
- Further RAID Level defined, e.g.: 6=5+1, ...
- Typical scenarios
  - Increase write speed needs striping (e.g. RAID 0)
  - RAID1: Simple, fast, safe, but needs lots of space
  - RAID5: More complex, safe, fast, requires more space, requires at last three disks

Oracle: Options without RAID



- Parallelization by distributing tablespaces
  - System tablespace on separate disk
  - Or: Tablespace-managed data dict.
  - Separate tablespaces for data / index
  - Separate disk for REDO Logs
- Parallelization by distributing one tablespace
- Parallelization by distributing a single table
  - Distribution of extends
  - Partitioning – value-based distribution of data
    - All sales prior to 2005 on one disk, all sales this year on another disk
    - One disk for sales in 2005, 2004, 2003, ...

Interference with RAID



- File layout and RAID interfere
- Multi-file distributed tablespace will not help if all files are RAID-distributed over the same physical disks
  - Mount points are not physical disks any more
- Proper design needs to consider both to prevent advantage-cancelling effects
- Note: Parallel reads must be consumed on upper levels – parallel memory access, parallel processing units

Some guidelines (Oracle handbooks)

- „Tsps should stripe *over at least as many devices as CPUs*"
- "You should stripe tablespaces for tables, indexes, rollback segments, and temporary tablespaces. You must also spread the devices over controllers, I/O channels, and internal buses"
  - Queries can run in parallel (inter-query parallelization)
  - Single disk is bottleneck – multiple processors become useless
  - Ideally, each disk becomes a "feed" for one processor (thread)
- Disadvantages
  - No simple backup of tablespace by file copying
  - Increased failure rate – use redundant RAID levels
  - Recovery of a disk might stop operations (all disks are involved)

Guidelines 2

- „In high-update OLTP systems, the redo logs are write-intensive. Moving the redo log files to disks that are separate from other disks and from archived redo log files has ... benefits ...“
  - Every transaction generates REDO information
  - REDO is written in batches before commit, data blocks are written sporadically by db-writer
  - Both should not interfere (too many seeks)
    - Hence: Put REDO log files away from data files
    - Disk crash can only effect REDO or data files
  - Redo data is extremely important (rollback, roll-forward)
    - Hence: Spread REDO data redundantly over many disks
    - By system (RAID) or by database (REDO groups)
  - REDO disks are good places to invest in RAID10

Typical Bottlenecks

- Temporary tablespace – used especially for large SORTS
  - And sorting is everywhere – sort-merge join, group by, order by, distinct, …
  - Receives many concurrent accesses from many processes
  - Hot spot – fast reads, fast writes, but failure is not critical
  - RAID0
- System tablespace
  - Holds data dictionary – important for everything
  - Required all the time – logs, latches, system log data, …
  - Especially logs can be a bottleneck
  - RAID1
- REDO log files
  - See last slide

Oracle flexible architecture (OFA)

The directory structure would look similar to this:

```
C:\oracle                                  --First logical drive
     \ora92                                --Oracle home
          \bin                             --Subtree for Oracle binaries
          \network                         --Subtree for Oracle Net
          \...
     \admin                                --Subtree for database administration files
          \prod                            --Subtree for prod database administration files
               \adhoc                      --Ad hoc SQL scripts
               \adump                      --Audit files
               \bdump                      --Background process trace files
               \cdump                      --Core dump files
               \create                     --Database creation files
               \exp                        --Database export files
               \pfile                      --Initialization parameter file
               \udump                      --User SQL trace files

F:\oracle                                  --Second logical drive (two physical drives, striped)
     \oradata                              --Subtree for Oracle database files
          \prod                            --Subtree for prod database files
               redo01.log                  --Redo log file group one, member one
               redo02.log                  --Redo log file group two, member one
               redo03.log                  --Redo log file group three, member one

G:\oracle                                  --Third logical drive (RAID level 5 configuration)
     \oradata                              --Subtree for Oracle database files
          \prod                            --Subtree for prod database files
               control01.ctl               --Control file 1
               indx01.dbf                  --Index tablespace datafile
               rbs01.dbf                   --Rollback tablespace datafile
               system01.dbf                --System tablespace datafile
               temp01.dbf                  --Temporary tablespace datafile
               users01.dbf                 --Users tablespace datafile

H:\oracle                                  --Fourth logical drive
     \oradata                              --Subtree for Oracle database files
```

OFA - Quote

- "The minimum configuration consists of seven data areas, either disks, striped sets, RAID sets, … The more heads you have moving at one time, the faster your database will be."
  - AREA1: Oracle executables and user areas, a control file, the SYSTEM tablespace, redo logs
  - AREA2: Data-data files, a control file, tool-data files, redo logs
  - AREA3: Index-data files, a control file, redo logs
  - AREA4: Rollback segment-data files
  - AREA5: Archive log files
  - AREA6: Export Files
  - AREA7: Backup Staging