Database Access Methods

RAID

Motivation

- Data are *very* important
- Good hard drives (reliable, fast) are expensive
- Cheap hard drives lack certain qualities
- How we can more from inexpensive hardware -> RAID

Outline

- Redundancy
- RAID 0 - 6
- RAID +

RAID

- Redundant Arrays of Inexpensive (Independent) Disks
- Inexpensive
  - original motivation
  - alternative to high-capacity expensive disks
- Independent
  - present-day motivation
  - higher reliability – redundancy
  - higher bandwidth – parallelism

- Must be supported by the controller

RAID - MTTF

- Mean time to failure (MTTF) of a system is much lower than MTTF of an individual device
- Redundancy of information can help by storing multiple copies of data which are then used in case of failure

RAID - Mirroring

- keeps copies of a disk → each write is carried out on multiple disks
- data are read from one disk – if one goes awry, the backup disk can be utilized

RAID - Parity

Example with 3 disks (D1, D2, D3), one parity disk (DP) and one hot spare disk (HS) used for recovery purposes:

To calculate parity XOR operation can be used
D1: 00100101
D2: 11101001
D3: 10101101
DP: 01100001 (= D1 XOR D2 XOR D3)

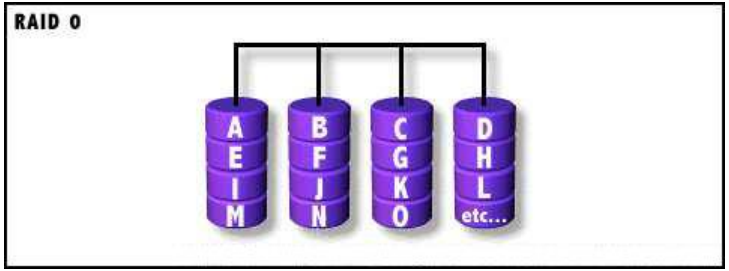
D2 breaks down

Now the original values of D2 can be obtained from the parity information from DP (D1 XOR D3 XOR DP) and can be written to HS which can serve as a new D2

Outline

- Redundancy
- RAID 0 - 6
- RAID +

RAID 0



- Stripping by blocks
- High performance
- Non-critical applications

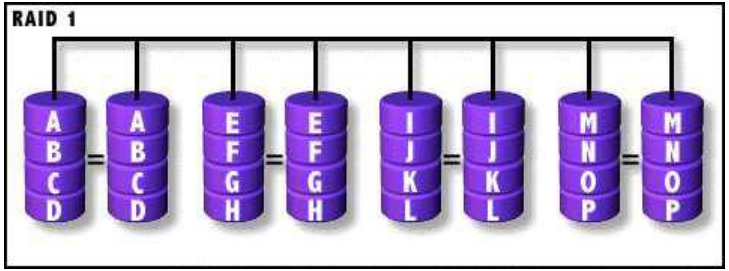
Advantages

- Easy to implement
- All storage capacity available
- Superior I/O performance

Disadvantages

- No fault-tolerance

RAID 1



- Disk mirroring
- Parallel writes, optionally parallel reads
- Data-critical applications (storing log files, accounting systems, …)

Advantages

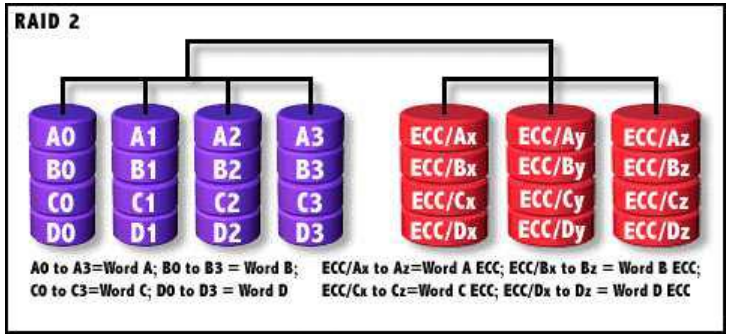
- Easy to implement
- I/O speed comparable to single disk
- In case of disk failure, data are only copied

Disadvantages

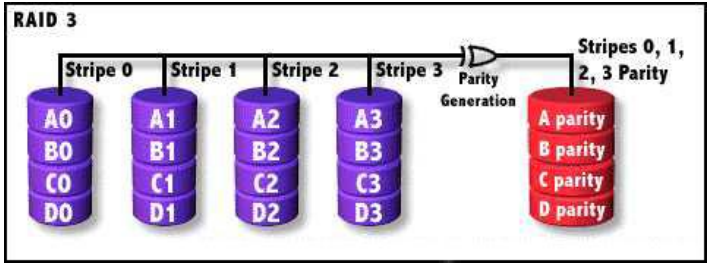
- 50% redundancy

RAID 2

- Bit-level striping
- Hamming code parity (error correction)
- Rarely used
- Synchronized rotations



RAID 3



- Byte-level striping
- One parity disk (XOR) - performance bottleneck
- Rarely used

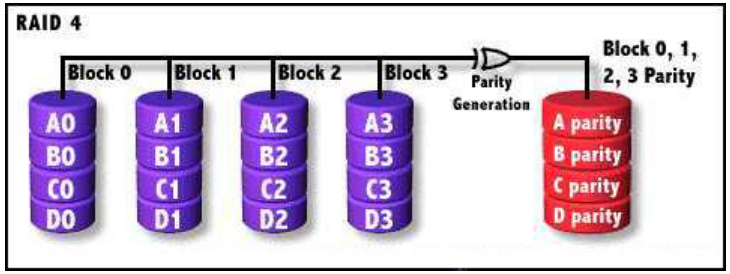
Advantages

- High-throughput for large I/O

Disadvantages

- At least 3 disks
- Slow for small I/O operations
- I/O requires activity on every disk

RAID 4



- Block-level striping
- One parity disk (XOR) - performance bottleneck
- Rarely used

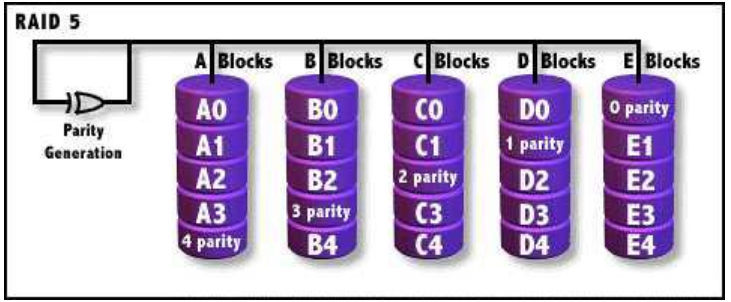
Advantages

- I/O requests can be carried out in parallel

Disadvantages

- Lot of small write operations can be problematic
- Complex controller design

RAID 5



- Block-level striping
- Distributed parity
- Most common
- Reads do not check the parity block (too expensive)
- Can handle single disk failure

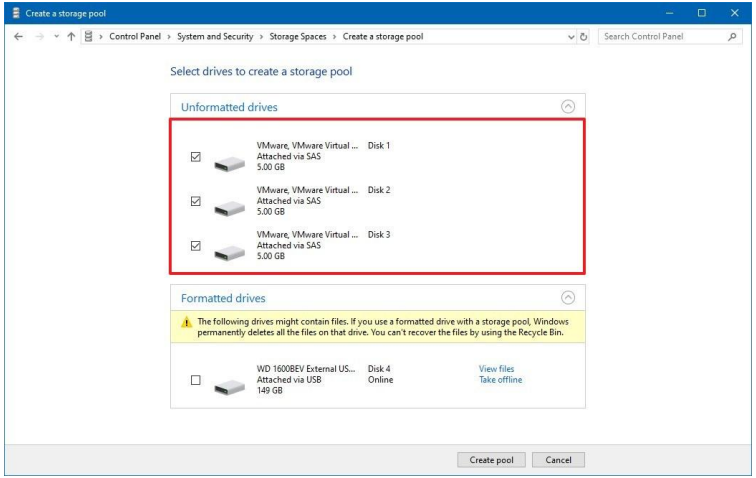
Advantages

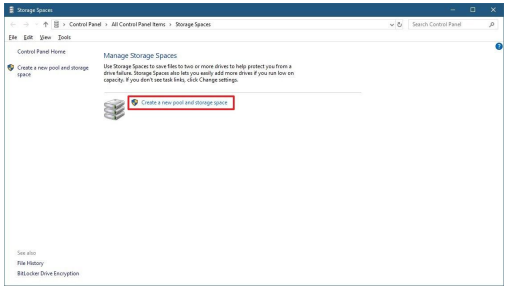
- High-throughput read operation
- Good aggregate transfer rate

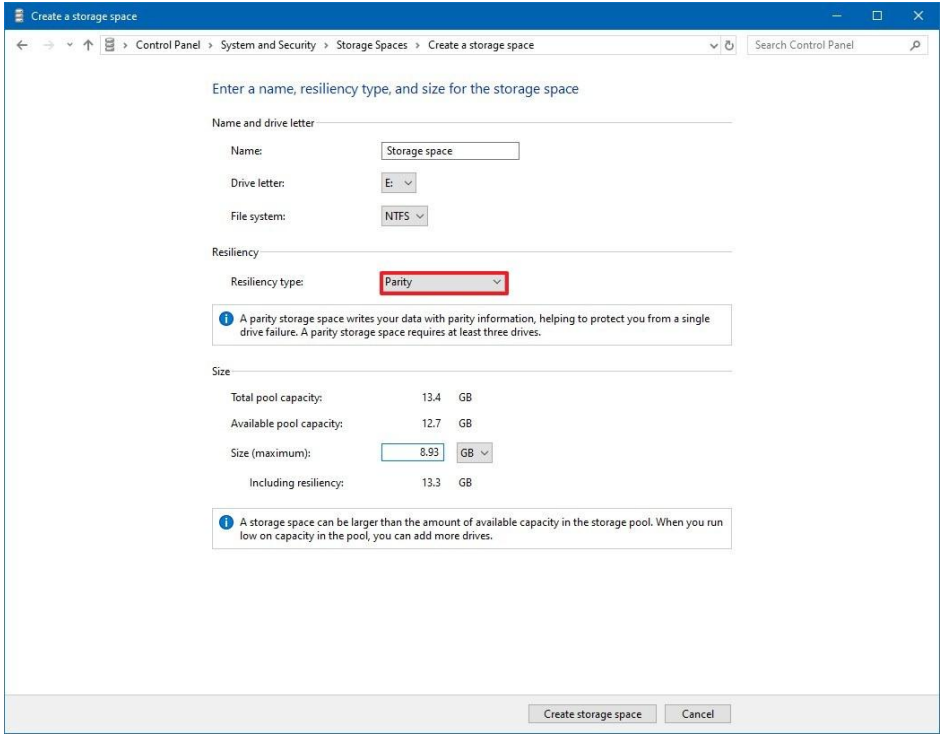
Disadvantages

- Write operation is slower (parity computation)
- At least 3 disks

RAID on Windows

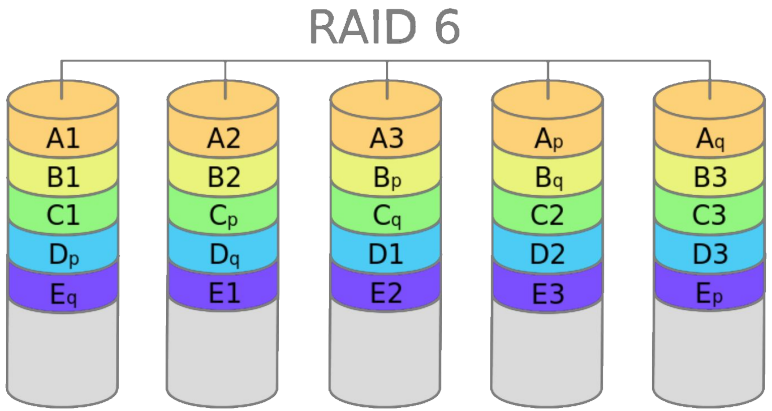






RAID 6

- Block-level striping
- 2 distributed parity blocks
- RAID 5 for enterprise



Advantages

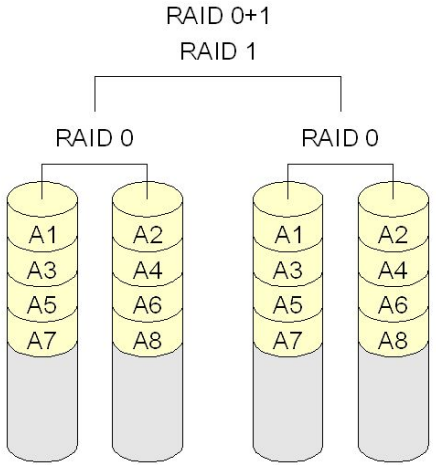
- Can handle failure of 2 drives

Disadvantages

- Expensive write

RAID 0 + 1

- RAID 1 of RAID 0
- 6 disks (2 sets of 3 disks)



Advantages

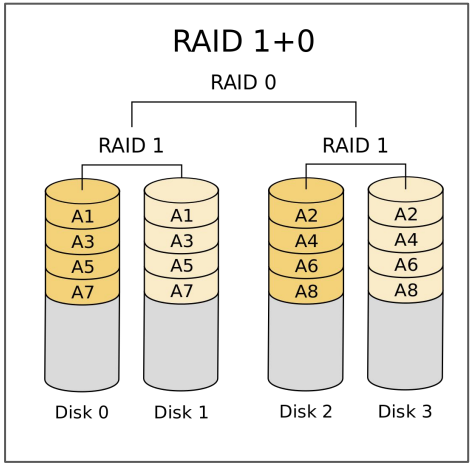
- Additional fault tolerance (as RAID 5)
- High data transfer rate

Disadvantages

- High overhead
- Limited scalability
- Requires 4 disks

RAID 1 + 0

- RAID 0 of with RAID 1
- 6 disks (3 sets of 2 disks)
- Database server requiring high performance and fault tolerance



Advantages

- High reliability
  (in each RAID 1 array, 1 disk can fail)

Disadvantages

- High overhead
- Limited scalability
- Requires 4 disks