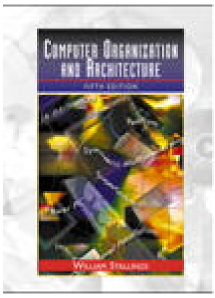
Campus de Gualtar
4710-057 Braga

UNIVERSIDADE DO MINHO
ESCOLA DE ENGENHARIA

Departamento de
Informática



Computer Organization and Architecture

5th Edition, 2000

by William Stallings

Table of Contents

II. THE COMPUTER SYSTEM.

3. ...
4. ...
5. External Memory. (28-Mar-00)

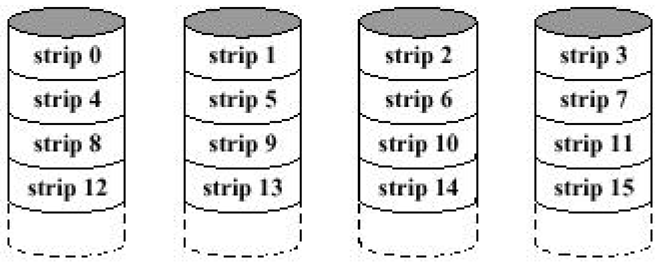
RAID (5.2)

Redundant Arrays of Independent Disks

Three Common (mostly) Characteristics

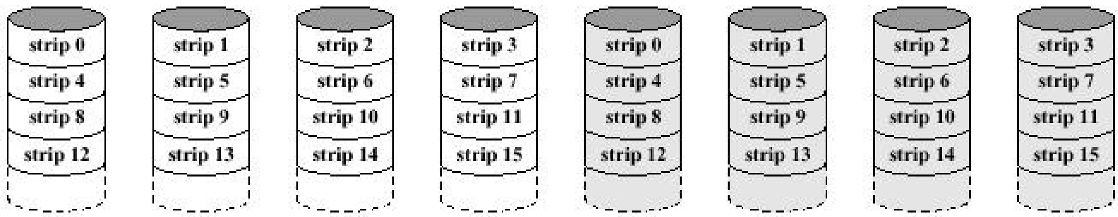
- RAID is a set of physical disk drives viewed by the operating system as a single logical drive.
- Data are distributed across the physical drives of an array.
- Redundant disk capacity is used to store parity information, which guarantees data recoverability in case of a disk failure.* * Except for RAID level 0.

Level 0 (Non-redundant)



- Not a true member of RAID – no redundancy!
- Data is striped across all the disks in the array
    - o Each disk is divided into strips which may be blocks, sectors, or some other convenient unit.
    - o Strips from a file are mapped round-robin to each array member
    - o A set of logically consecutive strips that maps exactly one strip to each array member is a stripe
- If a single I/O request consists of multiple contiguous strips, up to n strips can be handled in parallel, greatly reducing I/O transfer time.

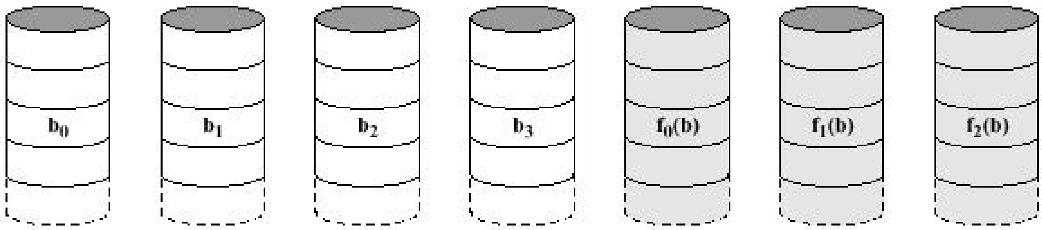
Level 1 (Mirrored)



- Only level where redundancy is achieved by simply duplicating all the data
- Data striping is used as in RAID 0, but each logical strip is mapped to two separate physical disks

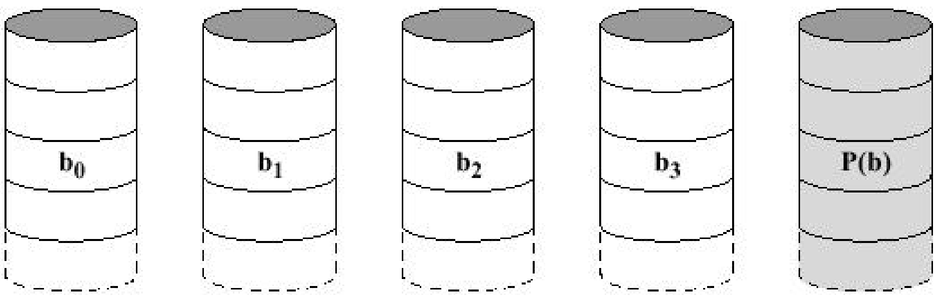
- A read request can be serviced by disk with minimal seek and latency time
- Write requests require updating 2 disks, but both can be updated in parallel, so no penalty
- When a drive fails, data may be accessed from other drive
- High cost for high performance
    - Usually used only for highly critical data.
    - Best performance when requests are mostly reads

Level 2 (Redundancy through Hamming Code)



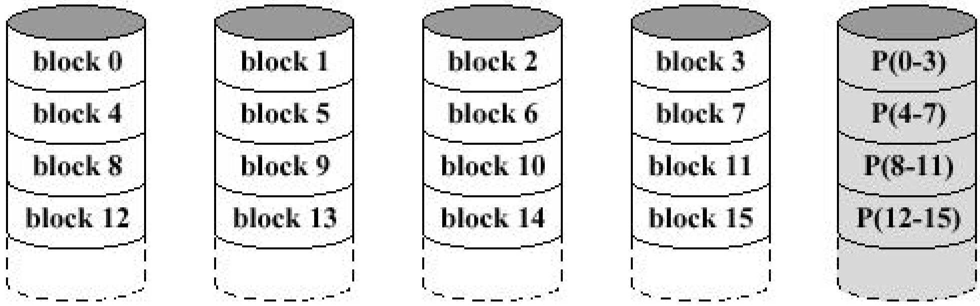
- Uses parallel access – all member disks participate in every I/O request
- Uses small strips, often as small as a single byte or word
- An error-correcting code (usually Hamming) is calculated across corresponding bits on each data disk, and the bits of the code are stored in the corresponding bit positions on multiple parity disks.
- Useful in an environment where a lot of disk errors are expected
    - Usually expensive overkill.
    - Disks are so reliable that this is never implemented

Level 3 (Bit-Interleaved Parity)



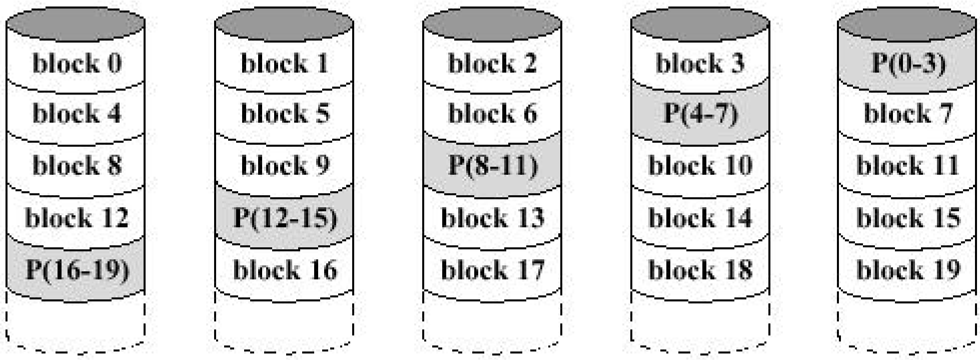
- Uses parallel access – all member disks participate in every I/O request
- Uses small strips, often as small as a single byte or word
- Uses only a single parity disk, no matter how large the disk array
    - A simple parity bit is calculated and stored
    - In the event of a failure in one disk, the data on that disk can be reconstructed from the data on the others
    - Until the bad disk is replaced, data can still be accessed (at a performance penalty) in reduced mode

Level 4 (Block-Level Parity)



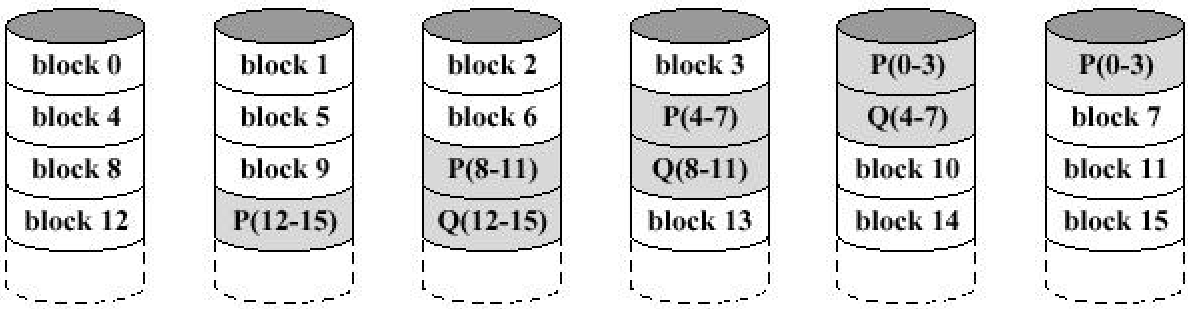
- Uses an independent access technique
  - o each member disk operates independently, so separate I/O requests can be satisfied in parallel.
  - o More suitable for apps that require high I/O request rates rather than high data transfer rates.
- Relatively large strips
- Has a write penalty for small writes, but not for larger ones (because parity can be calculated from values on other strips)
- In any case, every write involves the parity disk

Level 5 (Block-Level Distributed Parity)



- Like Level 4, but distributes parity strips across all disks, removing the parity bottleneck

Level 6 (Dual Redundancy)



- Like Level 6, but provides 2 parity strips for each stripe, allowing recovery from 2 simultaneous disk failures.