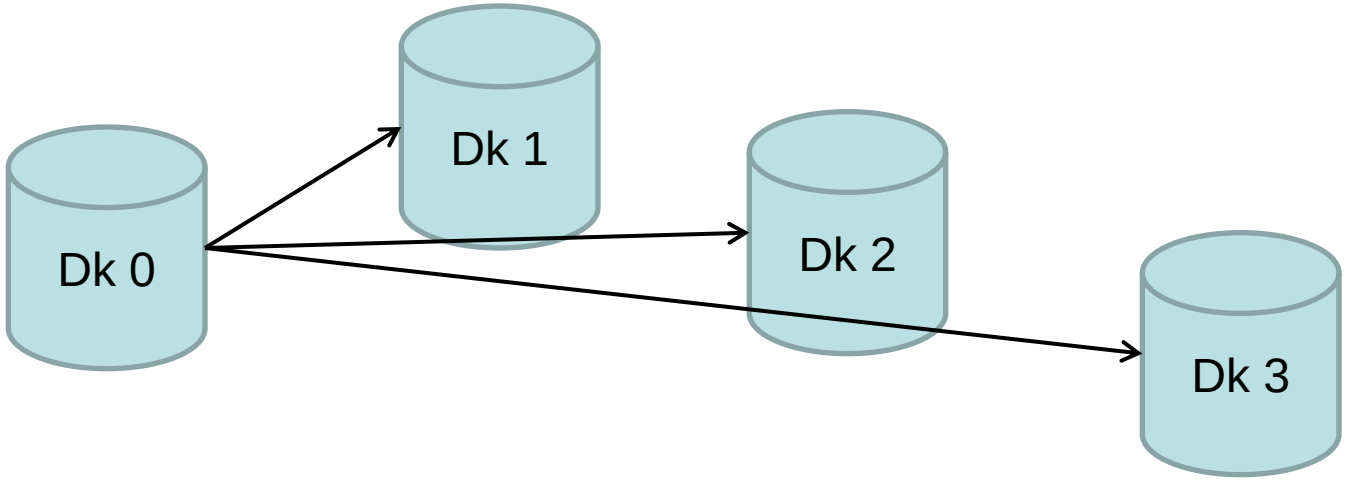
Lecture 16

RAID management

Problem

- Avoid loosing files/data due to disk failures.
- Solution: replication of file and store it on different disks.
- Example: SME with a local network. Transparently, when a file is stored, 2/3 copies are created and stored on randomly selected disks. Only the main copy is visible. This is a software solution.
- An addition (overhead) is the consistency preservation protocol.



New context

- Disks of extremely large capacity

- New requirements:
  - Reliability
  - Fast access to data
  - Fault tolerance
  - Scalability
  - Easy to manage
  - Transparent administration
  - Secure access to data

RAID concept

- The concept of Redundant Arrays of Inexpensive Disks (RAID) was introduced about three decades ago. A RAID taxonomy was first established by Patterson in 1988.
- RAID – multiple disk drives provides reliability via *redundancy*.
- Increases the *mean time to failure*.
- Frequently combined with *NVRAM* to improve write performance.
- RAID can be one of six different levels.
- Frequently, a small number of *hot-spare* disks are left unallocated, automatically replacing a failed disk and having data rebuilt onto them.

Faster access time by parallelism

- *Data striping*: splitting the bits of each byte across multiple disks – *bit-level striping*. With 8 disks, bit *i* will be stored on disk *i*. As a result, the "sector capacity" increases 8 times, the same for the access rate.

- This can be generalised to include a number of disks that is either a multiple of or divides 8. For example, for two disks, every second bit goes to the second disk.

- In *block-level striping*, blocks of a file are striped across multiple disks: with *n* disks, block *i* goes to disk *(i mod n) + 1*.

RAID levels

- Mirroring provides reliability but it is expensive. On the other hand, striping provides high data-transfer rates but not reliability.
- Different models that combine reliability and low cost were proposed:
- *RAID level 0*: striping at the level of blocks but without any redundancy.
- Benefits:
  - Best performance is achieved when data is striped across multiple controllers with only one drive per controller.
  - Very simple design, easy to implement.
- Major disadvantage:
  - Not a “true" RAID because it is NOT fault-tolerant; the failure of one drive will result in all data in an array being lost.

RAID level 1: disk mirroring

- Benefits:
  - One Write or two Reads possible per mirrored pair.
  - Twice the Read transaction rate of single disks, same Write transaction rate as single disks.
  - 100% redundancy of data means no rebuild is necessary in case of a disk failure, just a copy to the replacement disk. Under certain circumstances, RAID 1 can sustain multiple simultaneous drive failures.
  - Transfer rate per block is equal to that of a single disk.
  - Simplest RAID storage subsystem design.
- Disadvantages:
  - Highest disk overhead of all RAID types (100%) - inefficient
  - Typically the RAID function is done by system software, loading the CPU/Server and possibly degrading throughput at high activity levels. Hardware implementation is strongly recommended.

RAID level 2

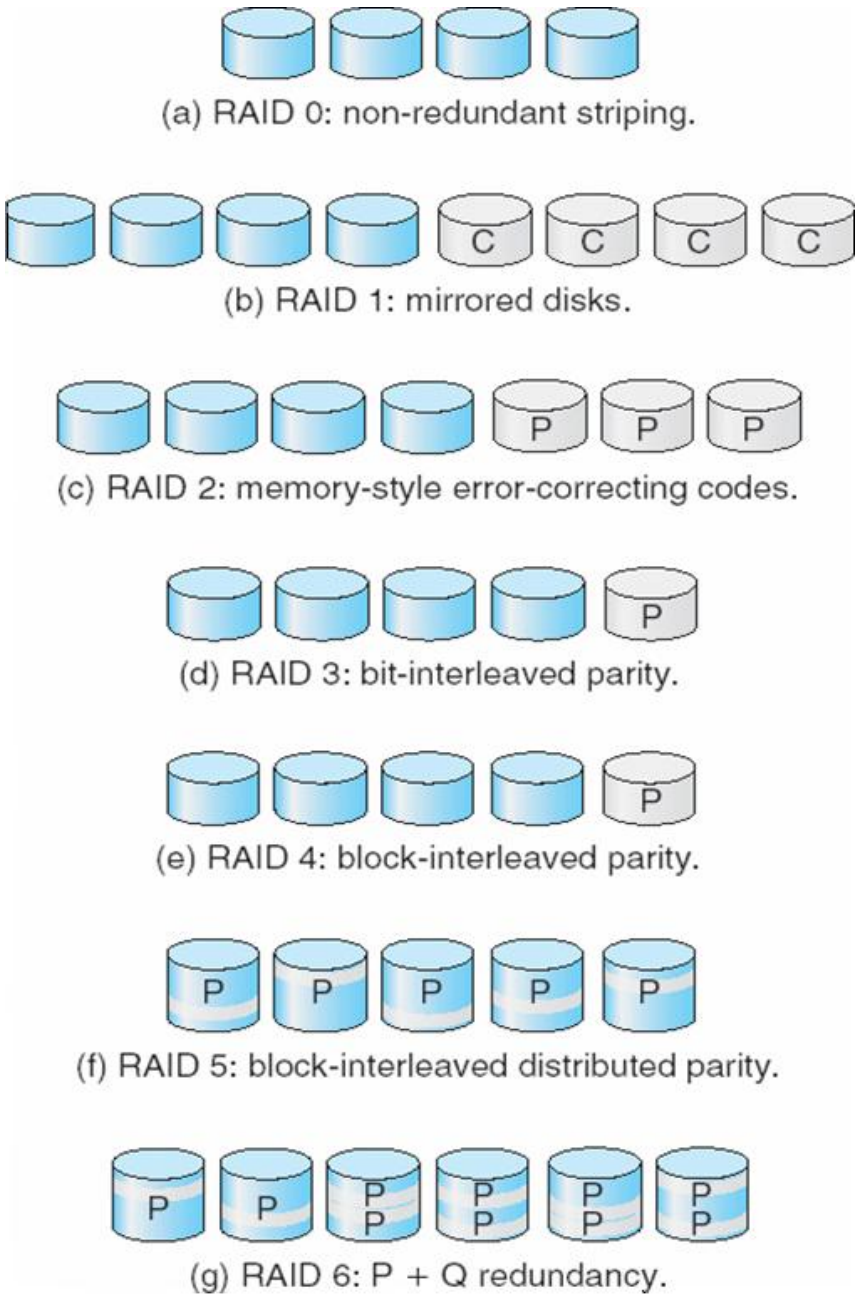
- It uses error-correcting code (ECC). One disk stores the 1st bit of all bytes, another disk the 2nd bit,...
- Three disks store the error-correcting bits. On Read, the ECC code verifies correct data or corrects single disk errors.
- Benefits:
  - "On the fly" data error correction.
  - Extremely high data transfer rates possible.
  - The higher the data transfer rate required, the better the ratio data disks/ECC disks
  - Relatively simple controller design compared to RAID levels 3,4 & 5.
- Disadvantages:
  - Very high ratio of ECC disks to data disks with smaller word sizes - inefficient
  - Entry level cost vey high - requires very high transfer rate requirement to justify
  - Transaction rate is equal to that of a single disk at best.
  - No commercial implementations exist.

- *RAID level 3*: bit-interleaved parity organisation, improves on level 2 by using only one disk for parity.

- *RAID level 4*: block-interleaved parity organisation, keeps a parity block on a separate disk. If one of the disks fails, the parity block can be used with the corresponding blocks from the other disks to restore the blocks on the failed disk.

- *RAID level 5*: block-interleaved distributed parity, spreads data and parity among all disks.

- *RAID level 6*: block-level striping with double distributed parity. Double parity provides fault tolerance up to two failed drives.



(a) RAID 0: non-redundant striping.

(b) RAID 1: mirrored disks.

(c) RAID 2: memory-style error-correcting codes.

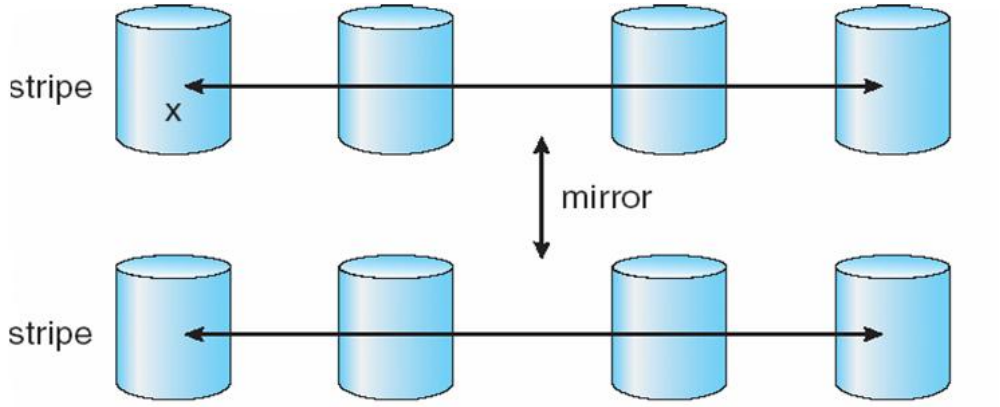
(d) RAID 3: bit-interleaved parity.

(e) RAID 4: block-interleaved parity.

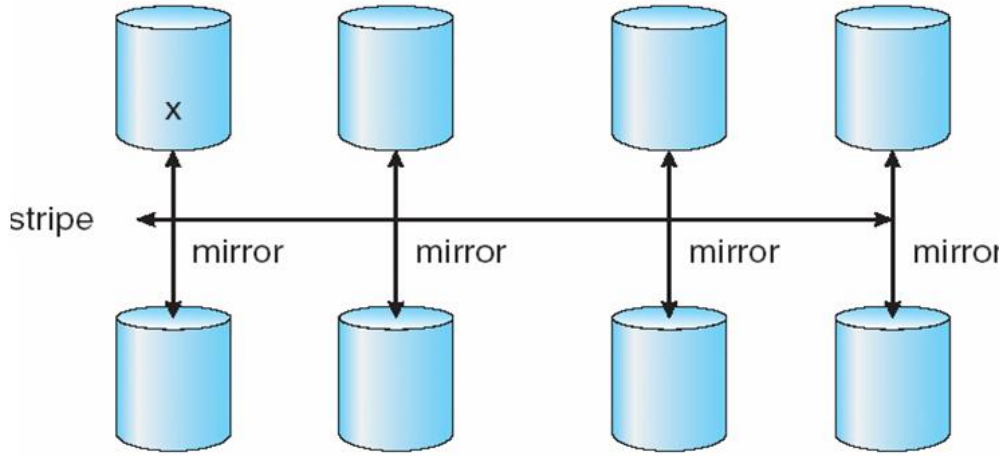
(f) RAID 5: block-interleaved distributed parity.

(g) RAID 6: P + Q redundancy.

RAID (0 + 1) and (1 + 0)



a) RAID 0 + 1 with a single disk failure.



b) RAID 1 + 0 with a single disk failure.

Selection of RAID level

- RAID 0 – High-Performance applications where data loss is not critical.
- RAID 1 – High Reliability with fast recovery.
- RAID 10/01 – Both performance and reliability are important, e.g. in small databases.
- RAID 5 – Preferred for storing large volumes of data.
- RAID 6 – Not Supported currently by many RAID implementations.

Implementation of RAID

- In the kernel, e.g. RAID 0, 1, or 0+1
- In the host-bus adapter hardware. It’s not flexible.
- In the hardware of the storage array. The OS needs to implement the file system on each of the volumes.
- In the storage area network layer by disk virtualisation.

Linux RAID support

- 2.6 kernel supports md (multiple devices): arrays can be built on top of entire disks or on partitions.
- mdadm is now the standard RAID management tool and should be found in any modern distribution. See http://linux.die.net/man/8/mdadm
- mdadm has 7 major modes of operation. Normal operation just uses the 'Create', 'Assemble' and 'Monitor' commands.
- /proc/mdstat: the array is running. you can create a file system, just like you would on any other device, mount it, include it in your /etc/fstab, and so on.

Questions

- Could RAID level 1 achieve better performance for read requests than RAID 0 (with non-redundant striping of data)? If so, how?

- Consider a RAID 5 architecture with five disks; the fifth stores the parity block. How many blocks are accessed in order to perform the following?
  - A write of one data block;
  - A write of seven continuous blocks of data.