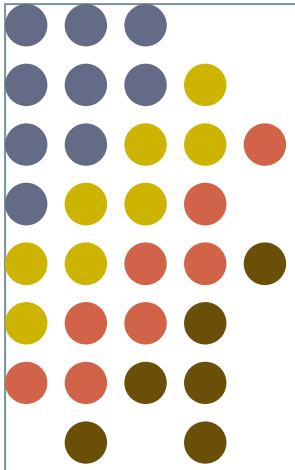


Input/Output

Disks

- Plates of magnetic material, organized in cylinders, divided in tracks, divided in sectors. Sectors go up to several hundreds. Heads vary from 1 to 16
- Some have little electronics, IDE drives have a microcontroller in the disk itself
- Controller: some can do *overlapped seeks* that is, while waiting for one seek to complete on one disk it can start another one on another disk

Example of the numbers

Parameter	IBM 360Kb floppy	WD 18300 HD
Number of cylinders	40	10601
Tracks per cylinder	2	12
Sectors per track	9	281 (average)
Sectors per disk	720	35742000
Bytes per sector	512	512
Disk capacity	360K	18.3G
Seek time	6ms	0.8ms
Seek time (average)	77ms	6.9ms
Rotation time	200ms	8.33ms
Motor start/stop	250ms	20s
Time to transfer 1 sector	22ms	17μs

Organization of the disk

- To simplify searching for sectors on the disk most disk presents:
  - A virtual organization in (c, t, s) that are mapped to the physical one
  - The number of sector per track changes as we move along the disk (minimum size of the magnetic site)
- On modern disks there's something called LBA (logical block addressing) where sectors are numbered sequentially without considering cylinder, track, or sector

Disk formatting

- Cylinder skew
  - Start cylinders at different points to give the head time to jump from one to the next
- Interleaving
  - To give the controller time to transfer to main memory
- Scheduling of head movements
  - Controllers can schedule the movements of the head

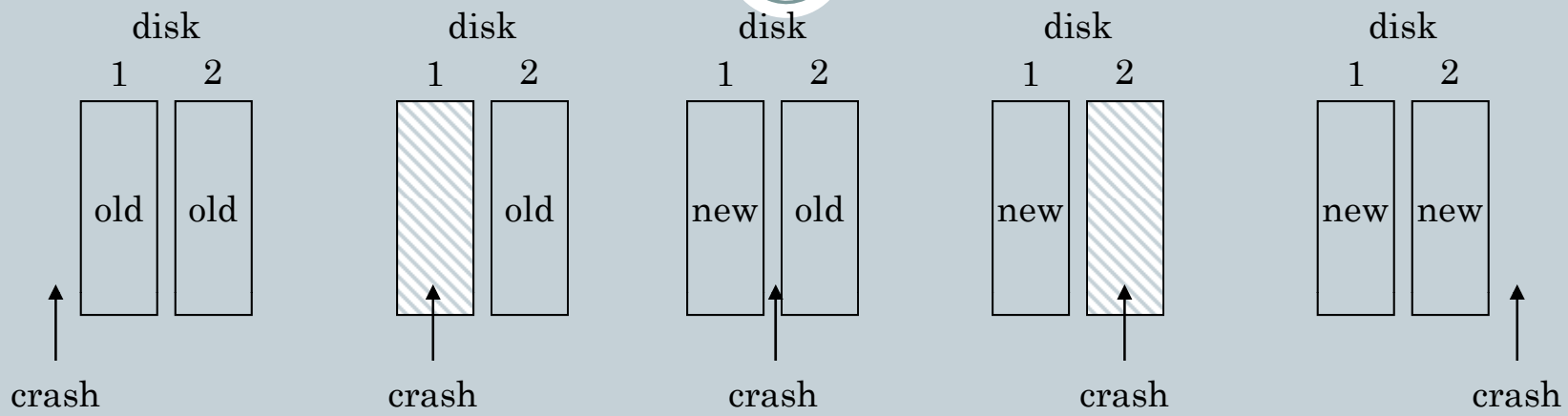
Error handling

- The controller can transparently take care of replacing a *bad* sector with a *spare* sector
- Bad sectors are due to wear and tear of the magnetic medium or construction defects (technology is always pushed to the extreme)

Stable storaging

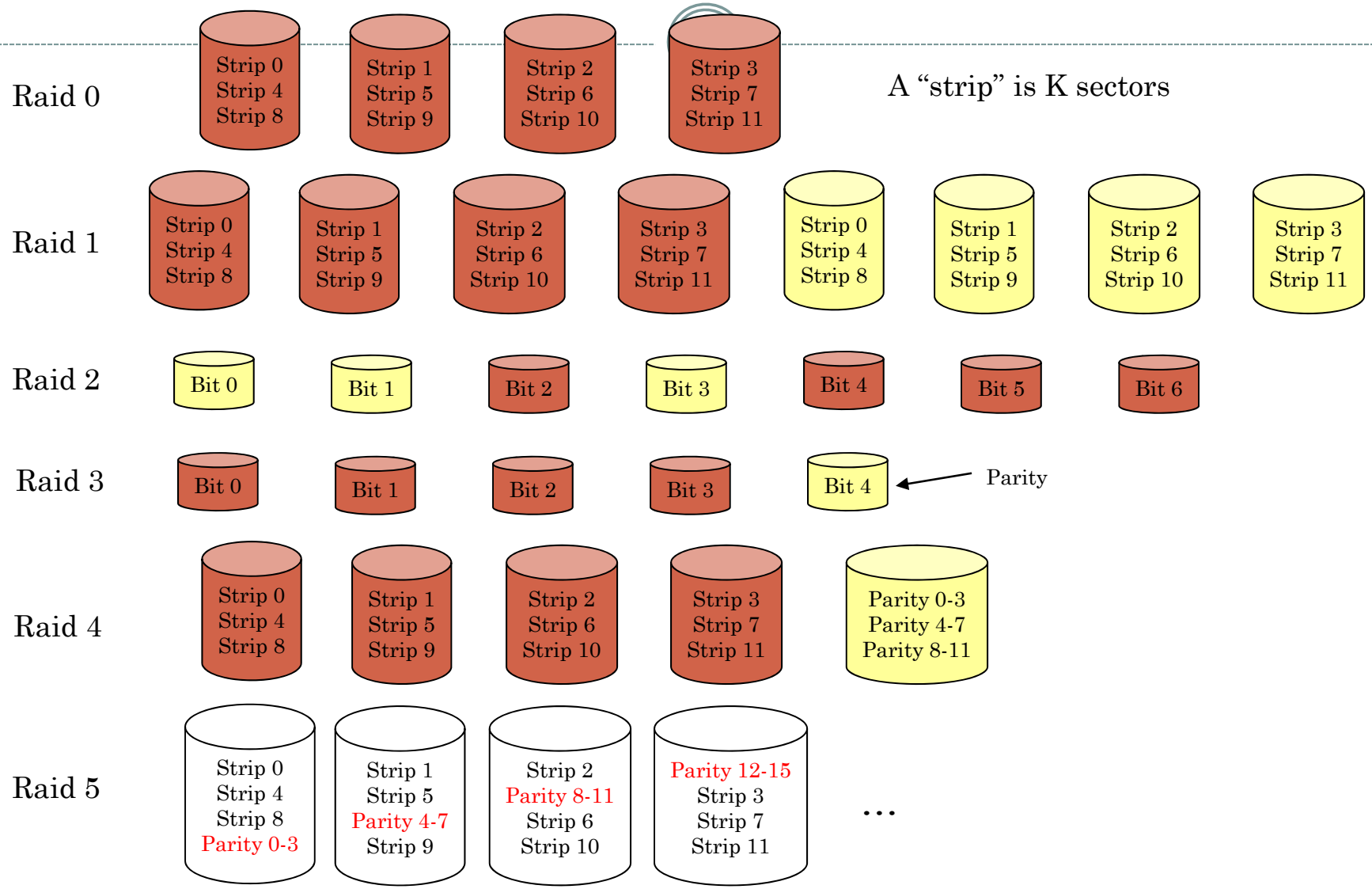
- Two identical disks
  - Probability of same block spontaneously go bad on two disk is negligible
- Stable write:
  - Write block on disk 1, read it back (re-read until it works up to N times). This would eventually work. Write on disk 2. Same procedure. In absence of CPU crashes the block is written twice
- Stable read:
  - Read from block 1. Read up to N times. If disk 1 went bad for any reason, read from disk 2. Since a stable write always succeeds also the read succeeds
- Crash recovery:
  - See next slide

Crash recovery



- The recovery program scans both disks. If a pair is good nothing is done, it one has an error (ECC) copy the good one into the bad one. If a pair has both block good but different, write 1 onto 2.

RAID


Raid 0
Strip 0
Strip 4
Strip 8
Strip 1
Strip 5
Strip 9
Strip 2
Strip 6
Strip 10
Strip 3
Strip 7
Strip 11
A “strip” is K sectors
Raid 1
Strip 0
Strip 4
Strip 8
Strip 1
Strip 5
Strip 9
Strip 2
Strip 6
Strip 10
Strip 3
Strip 7
Strip 11
Strip 0
Strip 4
Strip 8
Strip 1
Strip 5
Strip 9
Strip 2
Strip 6
Strip 10
Strip 3
Strip 7
Strip 11
Raid 2
Bit 0
Bit 1
Bit 2
Bit 3
Bit 4
Bit 5
Bit 6
Raid 3
Bit 0
Bit 1
Bit 2
Bit 3
Bit 4
Parity
Raid 4
Strip 0
Strip 4
Strip 8
Strip 1
Strip 5
Strip 9
Strip 2
Strip 6
Strip 10
Strip 3
Strip 7
Strip 11
Parity 0-3
Parity 4-7
Parity 8-11
Raid 5
Strip 0
Strip 4
Strip 8
Parity 0-3
Strip 1
Strip 5
Parity 4-7
Strip 9
Strip 2
Parity 8-11
Strip 6
Strip 10
Parity 12-15
Strip 3
Strip 7
Strip 11
...

CD technology

- Pits and lands:
  - Pit: depression in the plastic
  - Land: unburned area
- Different reflectance of the pits
  - It can be identified as a 1 or 0
- Recording follows a spiral
- Various materials for CD-R and CD-RW
- Improvement for DVD (also changed the laser wavelength)

Clocks

- Preventing one process to monopolize the CPU
- Maintaining the time of the day
- Accounting for CPU usage
- Handling ALARM signals
  - A device driver considers all the events (e.g. alarms) within the system (not a timer for every process/thread)
- Providing watchdog timers for part of the OS
  - e.g. timeouts – stop rotation of disk if not used
- Doing profiling, monitoring, statistics

Soft timers

- Handling an interrupt at each clock time would be too much
  - Interrupts involve delays
  - Can only be serviced with certain rates
- Idea! Why not calling the timer handling whenever in kernel mode
  - Occurrence of system calls, TLB misses, page faults, I/O interrupts, CPU idle, etc.
  - Experimentally
    - Going in kernel mode on average between 2 and 18μs
- Combine soft timer with a lower frequency one

More...

- Terminals
  - Character oriented
  - Over a serial line
- GUIs
  - MS Windows (GUI into the kernel)
  - X-Windows (GUI as a user process, native networking)
    - X-server: does the display
    - X-client: an application
- In practice the controller is seen by the OS as a buffer (the video buffer) mapped in memory

Last but not least

- Since batteries are big, expensive, and not particularly efficient
- Need to save power
- Power management (on laptops)…
- …but also on desktop computers
  - For fun. Each desktop has 200W power supply, 85% efficient. 100000 PCs consume 20MW equivalent to 20 average-size nuclear power plants