RAID

- Redundant Array of Inexpensive Disks
  - "A Case for Redundant Array of Inexpensive Disks" Patterson, Gibson, and Katz *1988*
  - ….old but crucial....

Observations

- CPU speed
  - MIPS= $2^{Year-1984}$ (Joy's Law, 1985)
- Memory
  - Amdahl's constant: each CPU instruction per second requires one byte of Main Memory
    - 1GHz CPU speed….needs ~ 1GB memory
- Chip capacity
  - #Trans/Chip = $2^{Year-1964}$ (Moore's Law, 1975)

Observation

- Secondary storage is largely mechanical!!!!
  - Density: $10^{(Year-1971)/10}$ (Frank, 1987)
  - Speed??

Observation

- Secondary storage is largely mechanical!!!!
  - Density: $10^{(Year-1971)/10}$ (Frank, 1987)
  - Speed??
    - Seek & rotation
      - From '71 to '81
        - seek time improved 2 times
        - rotation 0 times (now rotation is faster…but not by much)

Observation

- Secondary storage is largely mechanical!!!!
  - Density: $10^{(Year-1971)/10}$ (Frank, 1987)
  - Speed??
    - Seek & rotation
      - From '71 to '81
        - seek time improved 2 times
        - rotation 0 times (now rotation is faster…but not by much)
      - …caching helps
        - there are "Main Memory DBMSs", e.g., TimesTen
        - …..not a practical solution for all applications

Solution…..

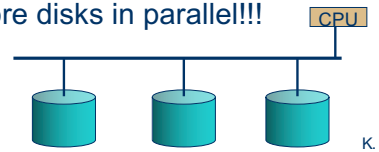
- Find a way to increase disk speed…
  - But how??

Solution…..

- Find a way to increase disk speed…
  - But how??
- Use parallel I/O!!!
  - But how??

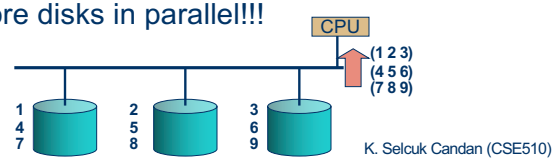
Solution…..

- Find a way to increase disk speed…
  - But how??
- Use parallel I/O!!!
  - But how??
- Price of disks keep decreasing…..so, use more disks in parallel!!!



Solution…..

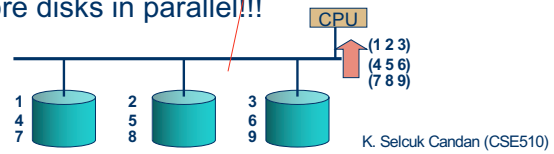
- Find a way to increase disk speed…
  - But how??
- Use parallel I/O!!!
  - But how??
- Price of disks keep decreasing…..so, use more disks in parallel!!!



Solution…..

- Find a way to increase disk speed…
  - But how??
- Use parallel I/O!!!
  - But how??
- Price of disks keep decreasing…..so, use more disks in parallel!!!

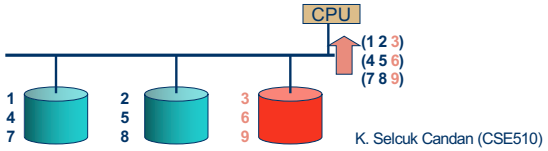
Good news: Disk don't need to be perfectly synchronized



Problem…

- Reliability
  - as you add more disks Mean Time to Failure (MTTF) of the system drops:

$$MTTF(n) = MTTF/n$$

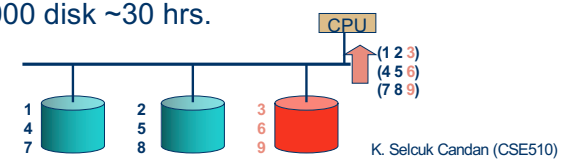


Problem…

- Reliability
  - as you add more disks Mean Time to Failure (MTTF) of the system drops:

$$MTTF(n) = MTTF/n$$

- 1 disk ~30K hrs.
- 1000 disk ~30 hrs.



Solution

- Use redundancy to increase reliability
- For every G disks in the array use C disks for error detection and correction
  - D: total # of disks with data (exc. check disks)
  - G: number of data disks in a group (exc. check disks)
    - In a single-group RAID G=D
  - C number of check disks in a group
  - ng: number of groups

K. Selcuk Candan (CSE510)

So what do we pay??

$$overhead = \frac{C}{G}$$

K. Selcuk Candan (CSE510)

Different applications

- Streaming media
  - High data rate needed
  - G disks are used in synchrony
- Databases (transactions)
  - Fast small reads/writes needed
  - G disks work independently

K. Selcuk Candan (CSE510)

RAID 1 (Mirror Disks)

- G=1, C = 1
- Every write to a disk also does to the check disk
  - Overhead: 100%
  - Large reads: 2D/S
  - Large writes: D/S
  - Large R-M-W = 4D/3S
  - Small reads: 2D
  - Small writes: D
  - Small R-M-W = 4D/3

S: synchronization slowdown

K. Selcuk Candan (CSE510)

RAID 2 (Hamming code for ECC)

- Hamming code: corrects 1bit, detects 2bit errors

D7	D6	D5	D4	D3	D2	D1	D0

RAID 2 (Hamming code for ECC)

- Hamming code: corrects 1bit, detects 2bit errors

D7	D6	D5	D4	D3	D2	D1	D0

C3	C2	C1	C0

RAID 2 (Hamming code for ECC)

- Hamming code: corrects 1bit, detects 2bit errors

D7	D6	D5	D4	D3	D2	D1	D0

C3	C2	C1	C0

C0 = D7⊕ D5⊕ D3⊕ D1
C1 = D7⊕ D6⊕ D3⊕ D2
C2 = D7⊕ D6⊕ D5⊕ D4
C3 = D7⊕ D6⊕ D5⊕ D4 ⊕ D3⊕ D2⊕ D1⊕ D0

RAID 2

- G=25, C = 5
- Disks are bit interleaved!!!!
  - Overhead: 20%
  - Large reads: D/S
  - Large writes: D/S
  - Large R-M-W = D/S
  - Small reads: D/SG
  - Small writes: D/2SG
  - Small R-M-W = D/SG

small write penalty (you need to read first to verify)!!!

RAID 2

On average better than RAID 1, because more disks can be used as data disks

- G=25, C = 5
- Disks are bit interleaved!!!!
  - Overhead: 20%
  - Large reads: D/S
  - Large writes: D/S
  - Large R-M-W = D/S
  - Small reads: D/SG
  - Small writes: D/2SG
  - Small R-M-W = D/SG

small write penalty (you need to read first to verify)!!!

RAID 3

- Single (!) check disk per group
  - Can't find which disk failed…but, usually the disk controller will signal it anyway
- ..i.e., only one parity disk per group!!

RAID 3

- G=25, C = 1
- Disks are bit interleaved!!!!
  - Overhead: 4%
  - Large reads: D/S
  - Large writes: D/S
  - Large R-M-W = D/S
  - Small reads: D/SG
  - Small writes: D/2SG
  - Small R-M-W = D/SG

RAID 3

- G=25, C = 1
- Disks are bit interleaved!!!!
  - Overhead: 4%
  - Large reads: D/S
  - Large writes: D/S
  - Large R-M-W = D/S
  - Small reads: D/SG
  - Small writes: D/2SG
  - Small R-M-W = D/SG

SO…WHERE IS THE GAIN??

RAID 3

- G=25, C = 1
- Disks are bit interleaved!!!!
  - Overhead: 4%
  - Large reads: D/S
  - Large writes: D/S
  - Large R-M-W = D/S
  - Small reads: D/SG
  - Small writes: D/2SG
  - Small R-M-W = D/SG

SO…WHERE IS THE GAIN??

More disks can be utilized for storing data…
….so, on the average per disk performance increases!!!

RAID 4

- How to fix small accesses??

RAID 4

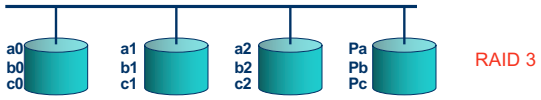
- How to fix small accesses??
  - Can we achieve independent reads writes?

RAID 4

- How to fix small accesses??
  - Can we achieve independent reads writes?
- We need to do more than one I/O per group

RAID 4

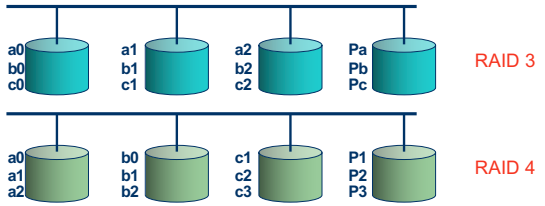
- How to fix small accesses??
  - Can we achieve independent reads writes?
- We need to do more than one I/O per group



K. Selcuk Candan (CSE510)

RAID 4 (sector interleave)

- How to fix small accesses??
  - Can we achieve independent reads writes?
- We need to do more than one I/O per group



K. Selcuk Candan (CSE510)

RAID 4

- G=25, C = 1
- Disks are sector interleaved!!!!
  - Overhead: 4%
  - Large reads: D/S
  - Large writes: D/S
  - Large R-M-W = D/S
  - Small reads: D
  - Small writes: D/2G
  - Small R-M-W = D/G

K. Selcuk Candan (CSE510)

RAID 5

- Check disk is bottleneck for small writes
  - How can we remove it?

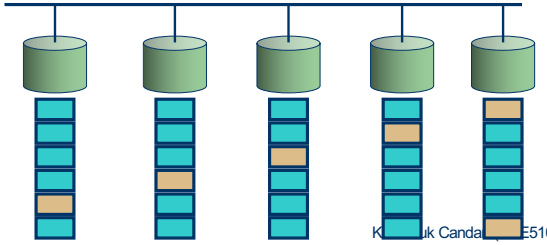
K. Selcuk Candan (CSE510)

RAID 5 (no single check disk)

- Check disk is bottleneck for small writes
  - How can we remove it?
- Distribute data onto check disks, vice versa!

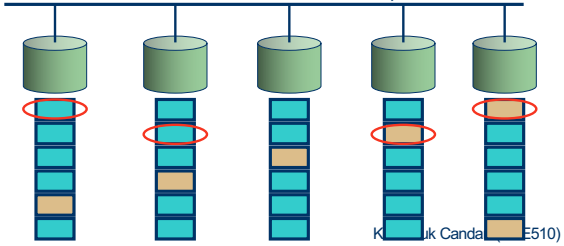
RAID 5 (no single check disk)

- Check disk is bottleneck for small writes
  - How can we remove it?
- Distribute data onto check disks, vice versa!



RAID 5 (no single check disk)

- Check disk is bottleneck for small writes
  - How can we remove it?
- Distribute data onto check disks, vice versa!



RAID 5

- G=25, C = 1
- Disks are sector interleaved!!!!
  - Overhead: 4%
  - Large reads: D/S
  - Large writes: D/S
  - Large R-M-W = D/S
  - Small reads: (1+C/G)D
  - Small writes: (1+C/G)D/4
  - Small R-M-W = (1+C/G)D/2