COMP4611: Design and Analysis of Computer Architectures

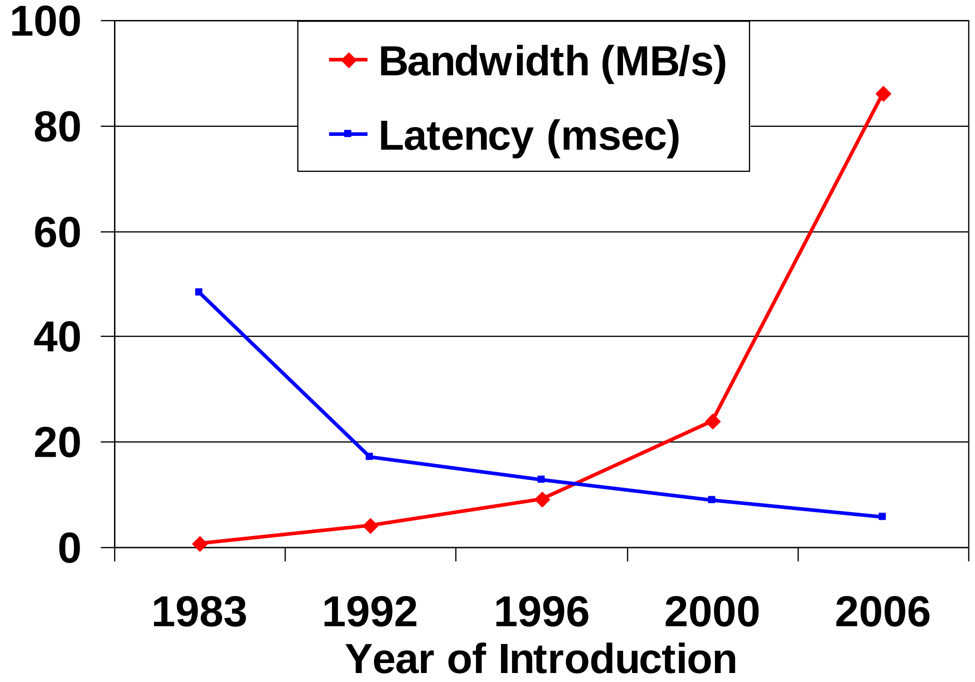
RAID (Redundant Array of Inexpensive Disks)

Lin Gu

CSE, HKUST

Disk Latency & Bandwidth Improvements

- Disk latency is one average seek time plus the rotational latency
- Disk bandwidth is the peak transfer rate of formatted data
- In the time that the disk bandwidth doubles the latency improves by a factor of only 1.2 to 1.4



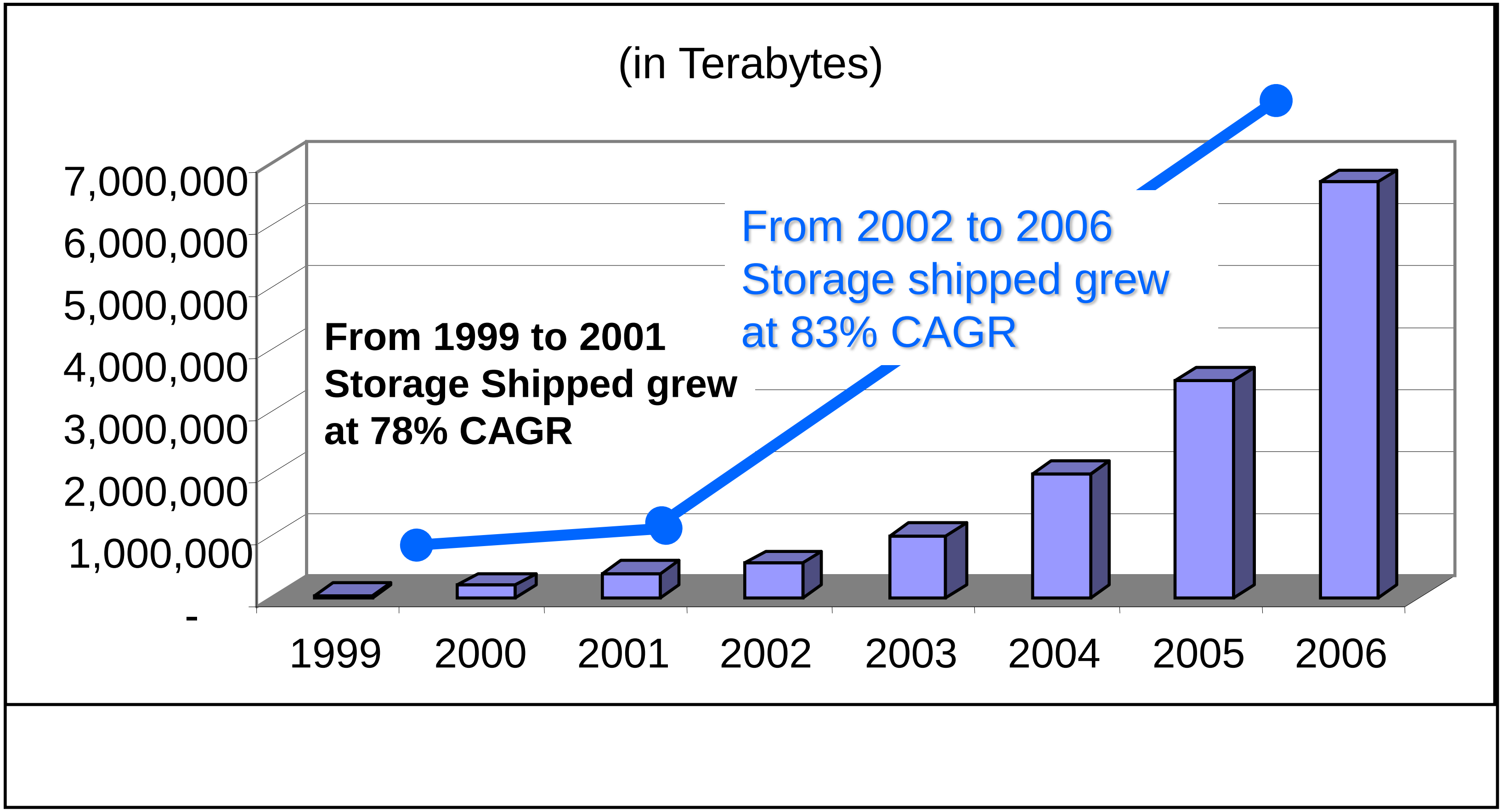
Media Bandwidth/Latency Demands

- Bandwidth requirements
  - High quality video
    - Digital data = (30 frames/s) $\times$ (640 x 480 pixels) $\times$ (24-b color/pixel) = 221 Mb/s (27.625 MB/s)
  - High quality audio
    - Digital data = (44,100 audio samples/s) $\times$ (16-b audio samples) $\times$ (2 audio channels for stereo) = 1.4 Mb/s (0.175 MB/s)
- Latency issues
  - How sensitive is your eye (ear) to variations in video (audio) rates?
  - How can you ensure a constant rate of delivery?
  - How important is synchronizing the audio and video streams?
    - 15 to 20 ms early to 30 to 40 ms late is tolerable

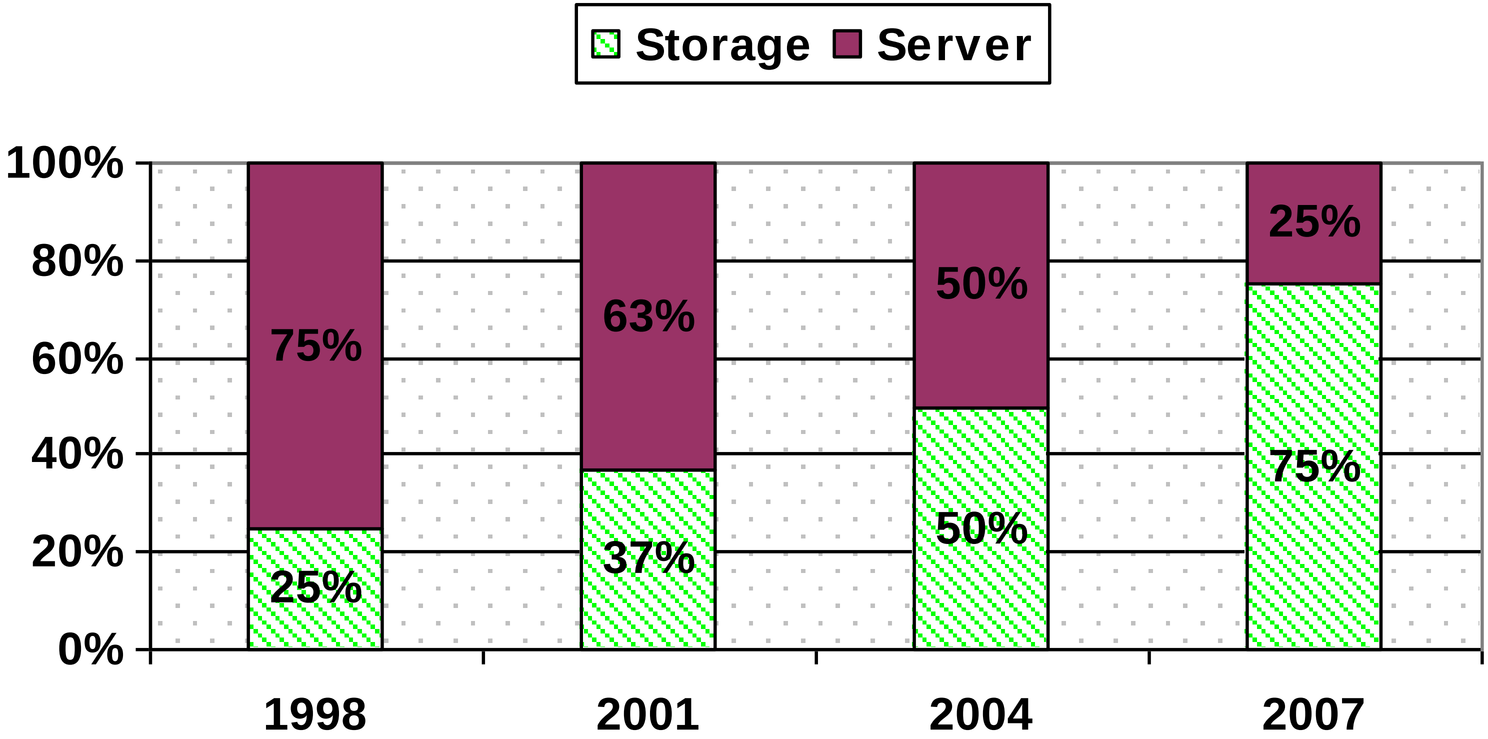
Storage Pressures

- Storage capacity growth estimates: 60-100% per year
  - Growth of e-business, e-commerce, and e-mail ⇨ now common for organizations to manage hundreds of TBs of data
  - Mission critical data must be continuously available
  - Regulations require long-term archiving
  - More storage-intensive applications on market
- Storage and Security are leading **pain points** for the IT community
- Managing storage growth effectively is a challenge

Data Growth Trends



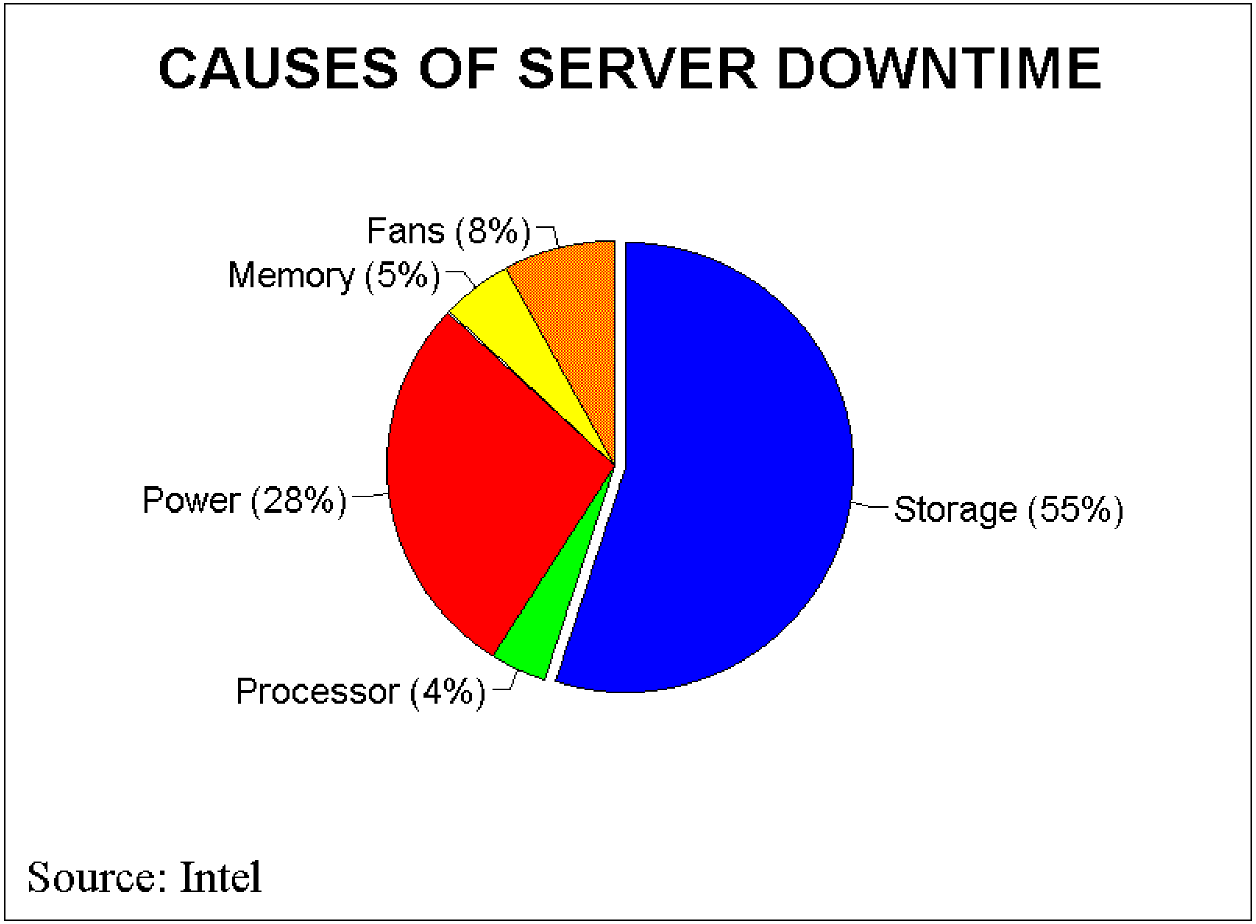
Storage Cost



Storage vs. server costs in IT spending

Availability/Reliability and Performance are EXTREMLY important

Importance of Storage Reliability



RAID

- To increase the availability and the performance (bandwidth) of a storage system, instead of a single disk, a set of disks (disk arrays) can be used.
- Similar to memory interleaving, data can be spread among multiple disks (*striping*), allowing simultaneous access to the data, improving the throughput and latency besides availability.
- However, the reliability of the system drops (n devices have $1/n$ the reliability of a single device).

Dependability Measures

- Reliability: mean time to failure (MTTF)
- Service interruption: mean time to repair (MTTR)
- Mean time between failures
  - MTBF = MTTF + MTTR
- Availability = MTTF / (MTTF + MTTR)
- Improving Availability
  - Increase MTTF: fault avoidance, fault tolerance, fault forecasting
  - Reduce MTTR: improved tools and processes for diagnosis and repair

Array Reliability

- Reliability of N disks = Reliability of 1 Disk ÷N

  50,000 Hours ÷ 70 disks = 700 hours

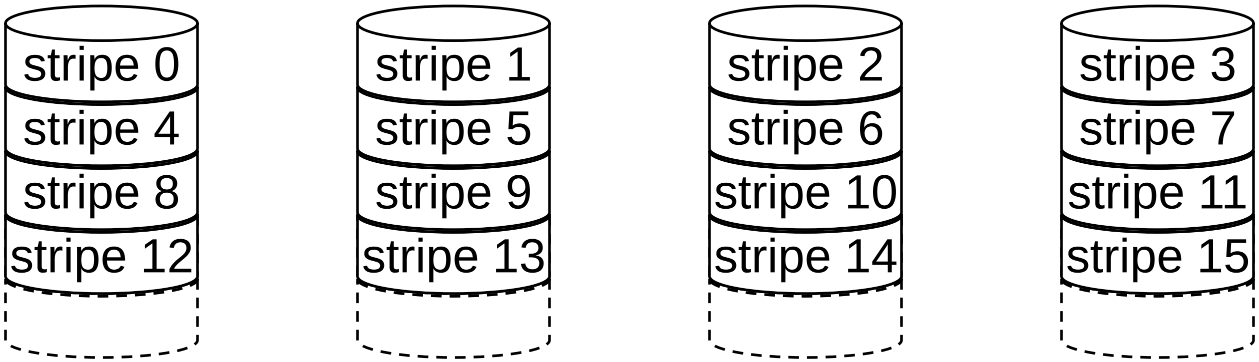
  Disk system Mean Time To Failure (MTTF): Drops from 6 years to 1 month!

Disks without redundancy are too unreliable to be useful!

RAID

- A disk array's availability can be improved by adding redundant disks:
  - If a single disk in the array fails, the lost information can be reconstructed from redundant information.
- This leads to a technology known as RAID - Redundant Array of Inexpensive Disks.
  - Depending on the number of redundant disks and the redundancy scheme used, RAIDs are classified into levels.
  - At least 6 levels of RAID (0-5) are accepted by the industry.
  - Level 2 is not commercially available

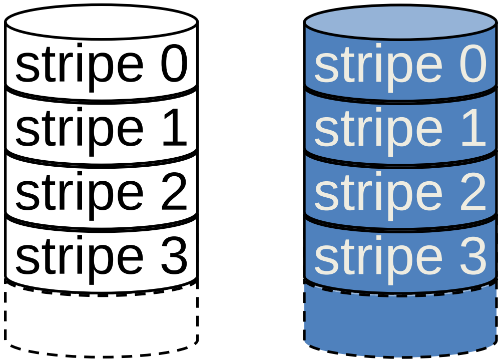
RAID-0



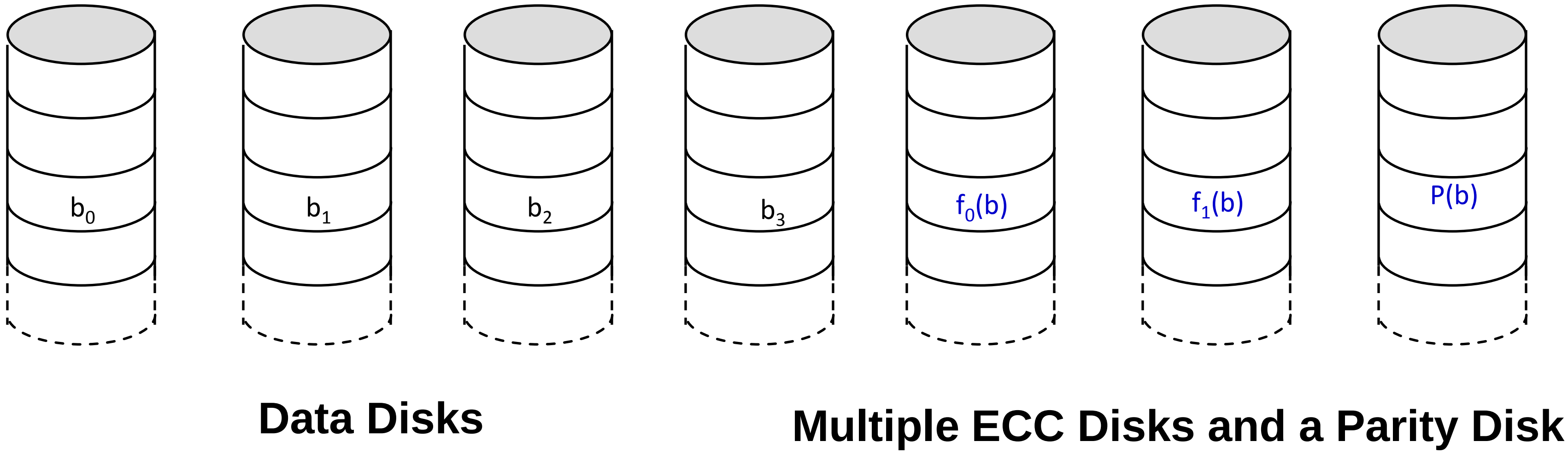
- Striped, non-redundant
  - Parallel access to multiple disks
  - ➔ Excellent data transfer rate
  - ➔ Excellent I/O request processing rate (for large stripes) if the controller supports independent Reads/Writes
  - ➔ Not fault tolerant (**AID**)
- Typically used for applications requiring high performance for non-critical data (e.g., video streaming and editing)

RAID-1 - Mirroring

- Called mirroring or shadowing, uses an extra disk (mirror) for each disk in the array
  - costly form of redundancy (but some FS, e.g., GFS, makes 3 copies)
- Whenever data is written to one disk, the data is also written to the mirror: good for reads (lower latency), fair for writes
- If a disk fails, the system goes to the mirror and gets the desired data.
- Fast, but very expensive.
- Typically used in system drives and critical files
  - Banking, insurance data
  - e-commerce servers



RAID-2: Memory-Style ECC

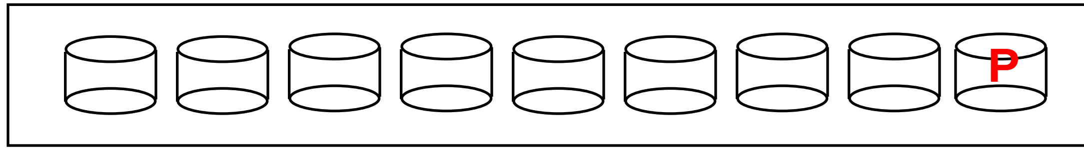


- Multiple disks record the (error correcting code) ECC information to determine which disk is in fault
- A parity disk is then used to reconstruct corrupted or lost data Needs $\log_2$(number of disks) redundancy disks
- Least used since ECC is irrelevant because most new Hard drives support built-in error correction

RAID-3 - Parity

10010011
11001101
10010011
. . .

Striped physical records



1	0	0	1	0	0	1	1	0
1	1	0	0	1	1	0	1	1
1	0	0	1	0	0	1	1	0

Logical record

Physical record

- Use 1 extra disk for each array of *n* disks. Bytes in a data block are stored alternately on all disks except for the parity disk.
- Reads or writes go to all disks in the array, with the extra disk to hold the parity information in case there is a failure.
- The parity is carried out at bit level:
  - A parity bit is kept for each bit position across the disk array and stored in the redundant disk.
  - Parity: sum modulo 2.
    - parity of 1010 is 0
    - parity of 1110 is 1

Or use XOR of bits

RAID-3 - Parity

If one of the disks fails, the data for the failed disk must be recovered from the parity information:

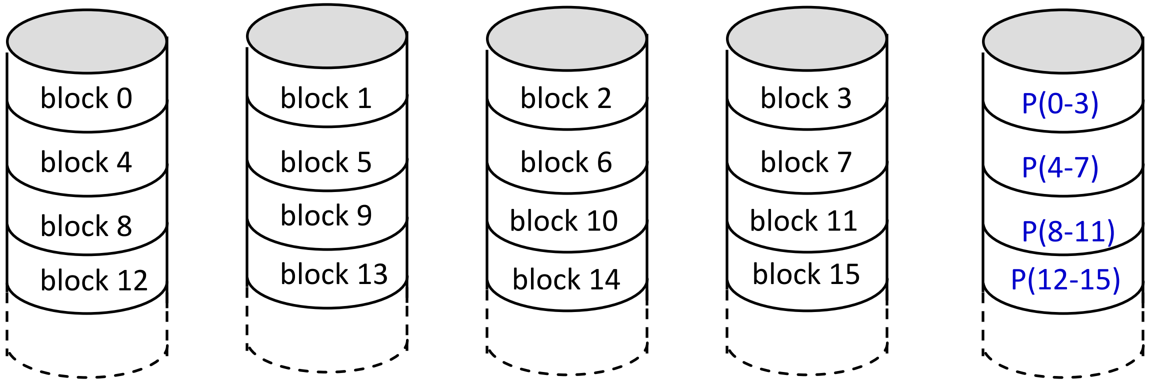
- This is achieved by subtracting the parity of good data from the original parity information:
- Recovering from failures takes longer than in mirroring
- Examples:

Original data	Original Parity	Failed Bit	Recovered data
1010	0	101X	|0-0| = 0
1010	0	10X0	|0-1| = 1
1110	1	111X	|1-1| = 0
1110	1	11X0	|1-0| = 1

RAID-4 - Block-interleaved Parity

- In RAID 3, every read or write needs to go to all disks since bits are interleaved among the disks.
- Performance of RAID 3:
  - Only one request can be serviced at a time
  - Poor I/O request rate
  - Excellent data transfer rate
  - Typically used in large I/O request size applications, such as imaging or CAD
- RAID 4: If we distribute the information block-interleaved, where a disk sector is a block, then for normal reads different reads can access different segments in parallel. Only if a disk fails will we need to access all the disks to recover the data.

RAID-4: Block Interleaved Parity



- Allow for parallel access by multiple I/O requests
- Doing multiple small reads is now faster than before.
- A write, however, is a different story since we need to update the parity information for the block.
- Large writes (full stripe), update the parity:

$$P' = d0' \oplus d1' \oplus d2' \oplus d3';$$

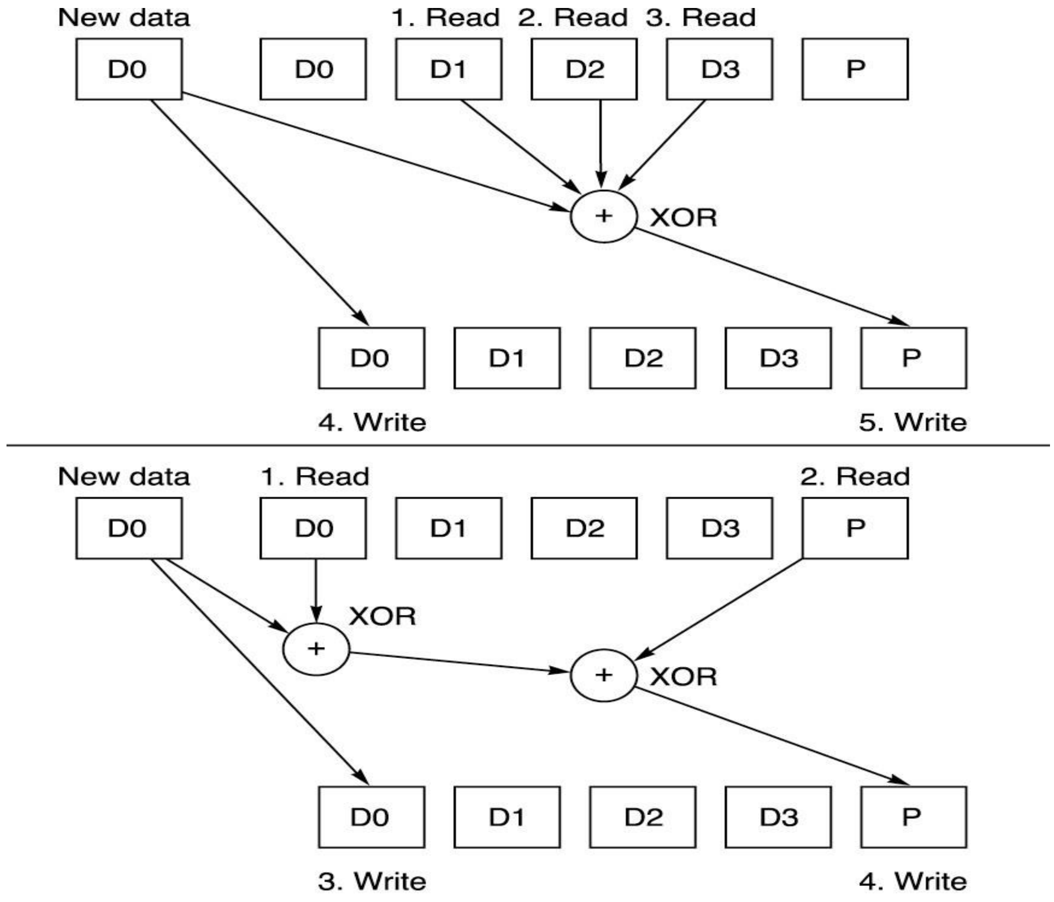
- Small writes (eg. write on d0), update the parity:

$$P = d0 \oplus d1 \oplus d2 \oplus d3$$

$$P' = d0' \oplus d1 \oplus d2 \oplus d3 = d0' \oplus d0 \oplus P;$$

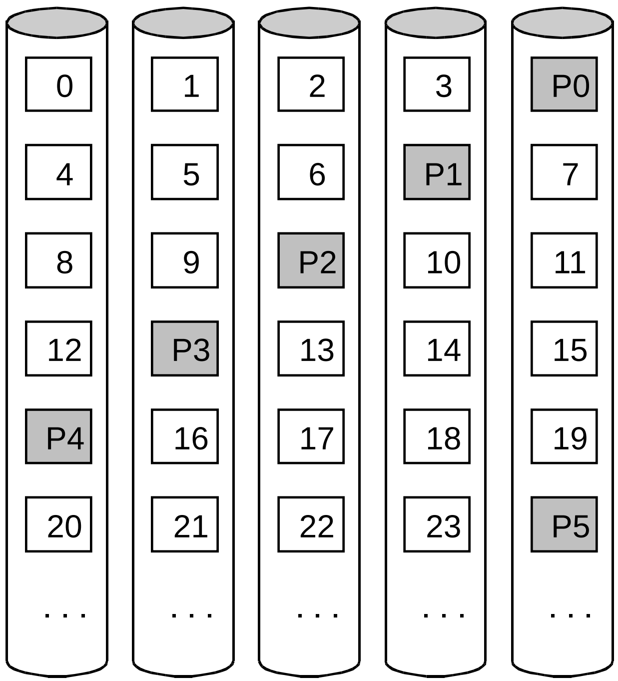
- However, writes are still very slow since parity disk is the bottleneck.

RAID-4: Small Writes



RAID-5 - Block-interleaved Distributed Parity

RAID 5 distributes the parity blocks among all the disks.



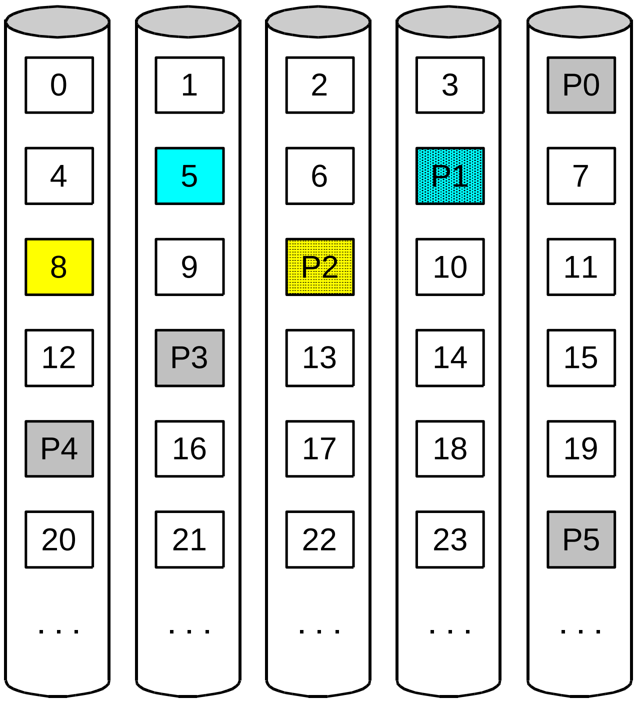
RAID 5

Why is this helpful?

RAID-5 - Block-interleaved Distributed Parity

This allows *some* writes to proceed in parallel

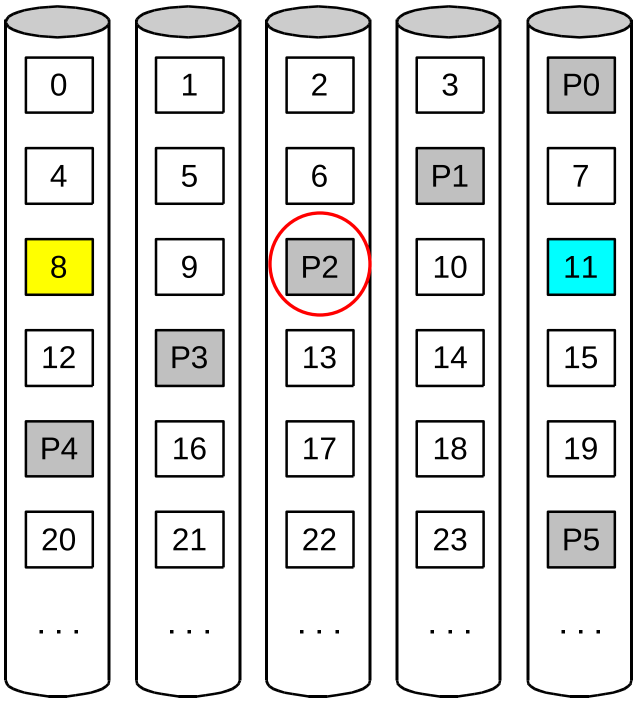
- For example, writes to blocks 8 and 5 can occur simultaneously.



RAID 5

RAID-5 - Block-interleaved Distributed Parity

- However, writes to blocks 8 and 11 cannot proceed in parallel.



RAID 5

Performance of RAID-5 - Block-interleaved Distributed Parity

- **Performance of RAID-5**
  - I/O request rate: excellent for reads, good for writes
  - Data transfer rate: good for reads, good for writes
  - Typically used for high request rate, read-intensive data lookup
  - **File and Application servers, Database servers, WWW, E-mail, and News servers, Intranet servers**
- Widely used.

RAID-6 – Row-Diagonal Parity

- To handle 2 disk errors
  - In practice, another disk error can occur before the first problem disk is repaired
- Use p-1 data disks, 1 row-parity disk, 1 diagonal-parity disk
- If any two of the p+1 disks fail, data can still be recovered

Data Disk 0	Data Disk 1	Data Disk 2	Data Disk 3	Row Parity Disk	Diagonal Parity Disk
0	1	2	3	4	0
1	2	3	4	0	1
2	3	4	0	1	2
3	4	0	1	2	3