Data Storage and Query Answering

Data Storage and Disk Structure (3)

Disk Failures

- In an *intermittent failure*, a read or write operation is unsuccessful, but succeeds with repeated tries.
  → parity checks to detect intermittent failures
- *Media decay* is a permanent corruption of one or more bits which make the corresponding sector impossible to read / write.
  → stable storage to recover from media decay
- A *disk crash* makes the entire disk permanently unreadable.
  → RAID to recover from disk crashes

Disk Failures

Checksums

- Add *n* parity bits every *m* data bits.
- The number of 1's among a collection of bits and their *parity bit* is always even.
- The parity bit is the *modulo-2 sum* of its data bits.
  - m=8, n=1
    - Block A: 01101000:1 (odd # of 1's)
    - Block B: 11101110:0 (even # of 1's)
- If Block A instead contains
    - Block A': 01100000:1 (has odd # of 1's)

  → error detected

Disk Failures

Checksums

- But what if multiple bits are corrupted?
- E.g., if Block A instead contains
  - Block A'': 01000000:1 (has even # of 1's)

  → error cannot be detected
- Probability that a single parity bit cannot detect a corrupt block is ½.
- This is assuming that the probability of disk failures involving an odd / even number of bits is identical.

Disk Failures

Checksums

- More parity bits decrease the probability of an undetected failure. With $n \leqslant m$ *independent* parity bits, this probability is only $1/2^n$.
- E.g., we can have eight parity bits, one for the first bit of every byte, the second one for the second bit of every byte . . .
- The chance for not detecting a disk failure is then only 1/256.

Disk Failures

Stable storage

- Sectors are paired, and information X is written both on sectors X_l and X_r.
- Assume that both copies are written with a sufficient number of parity bits so that bad sectors can be detected.
- If sector is bad (according to checksum), write to alternative sector.
- Alternate reading X_l and X_r until a good value is returned.
- Probability of X_l and X_r both failing is very low.

Disk Failures

Disk arrays

- So far, we cannot recover from disk crashes.
- To address this problem, use *Redundant Arrays of Independent Disks* (*RAID*), arrangements of several disks that gives abstraction of a single, large disk.
- Goals: Increase reliability (and performance).
- Redundant information allows reconstruction of data if a disk fails.
- Data striping improves the disk performance.

Disk Failures

Failure Models for Disks

- What is the expected time until disk crash?
- We assume uniform distribution of failures over time.
- *Mean time to failure*: time period by which 50% of a population of disks have failed (crashed).
- Typical mean time to failure is 10 years.
- In this case, 5% of disks crash in the first year, 5% crash in the second year, . . ., 5% crash in the tenth year, . . ., 5% crash in the twentieth year.

Disk Failures

Failure Models for Disks

- Given the mean time to failure (*mtf*) in years, we can derive the probability *p* of a particular disk failing in a given year.
- $p = 1 / (2 * mtf)$
- Ex.: $mtf = 10$, $p = 1/20 = 5\%$
- *Mean time to data loss*: time period by which 50% of a population of disks have had a crash that resulted in data loss.
- The mean time to disk failure is not necessarily the same as the mean time to data loss.

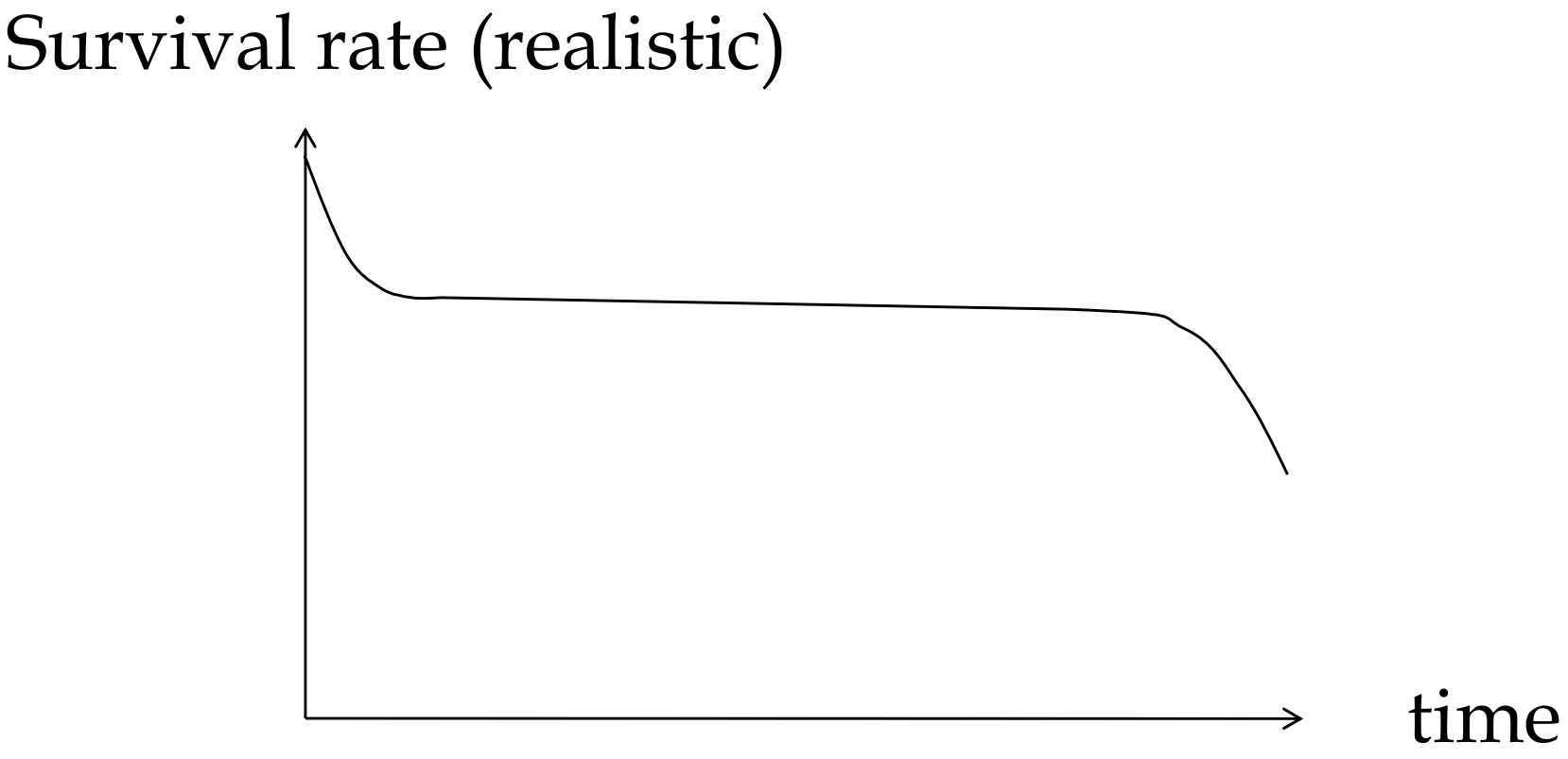
Disk Failures

Failure Models for Disks

- *Failure rate*: percentage of disks of a population that have failed until a certain point of time.
- *Survival rate*: percentage of disks of a population that have *not* failed until a certain point of time.
- While it simplifies the analysis, the assumption of uniform distribution of failures is unrealistic.
- Disks tend to fail early (manufacturing defects that have not been detected) or late (wear-and-tear).

Disk Failures

Failure Models for Disks



Disk Failures

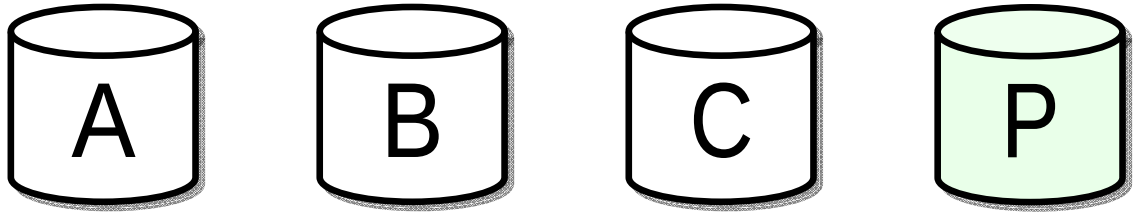
Mirroring

- The *data disk* is copied unto a second disk, the *mirror* disk.
- When one of the disk crashes, we replace it by a new disk and copy the other disk to the new one.
- Data loss can only occur if the second disk crashes while the first one is being replaced.
- This probability is negligible.
- Mirroring is referred to as *RAID level 1*.

Disk Failures

Parity blocks

- Mirroring doubles the number of disks needed.
- The parity block approach needs *only one redundant disk* for n (arbitray) data disks.
- In the redundant disk, the ith block stores parity checks for the ith blocks of all the n data disks.



- Parity block approach is called *RAID level 4*.

Disk Failures

Parity blocks

- Reading blocks is the same as without parity blocks.
- When writing a block on a data disk, we also need to update the corresponding block of the redundant disk.
- This can be done using four (three additional) disk I/O: read old value of data disk block, read corresponding block of redundant (parity) disk, write new data block, recompute and write new redundant block.

Disk Failures

Parity blocks

- If one of the disks crashes, we bring in a new disk.
- The content of this disk can be computed, bit by bit, using the remaining n disks.
- No difference between data disks and parity disk.
- Computation based on the definition of parity, i.e. total number of ones is even.

Disk Failures

Example

- n = 3 data disks
  - Disk 1, block 1: 11110000
  - Disk 2, block 1: 10101010
  - Disk 3, block 1: 00111000
- … and one parity disk
  - Disk 4, block 1: 01100010

→ Sum over each column is always an even number of 1's

→ Mod-2 sum can recover any missing *single* row

Disk Failures

Example

- Suppose we have:
  - Disk 1, block 1: 11110000
  - Disk 2, block 1: ????????
  - Disk 3, block 1: 00111000
  - Disk 4, block 1: 01100010 (parity)
- Use mod-2 sums for block 1 over disks 1,3,4 to recover block 1 of failed disk 2:

  → Disk 2, block 1: 10101010

Disk Failures

RAID level 5

- In the RAID 4 scheme, the parity disk is the bottleneck. On average, n-times as many writes on the parity disk as on the data disks.
- However, the failure recovery method does not distinguish the types of the $n + 1$ disks.
- *RAID level 5* does not use a fixed parity disk, but use block i of disk j as redundant if i MOD $n+1 = j$.