Application of Erasure Codes

Extra Lecture Notes

Mitigate Hardware Failure

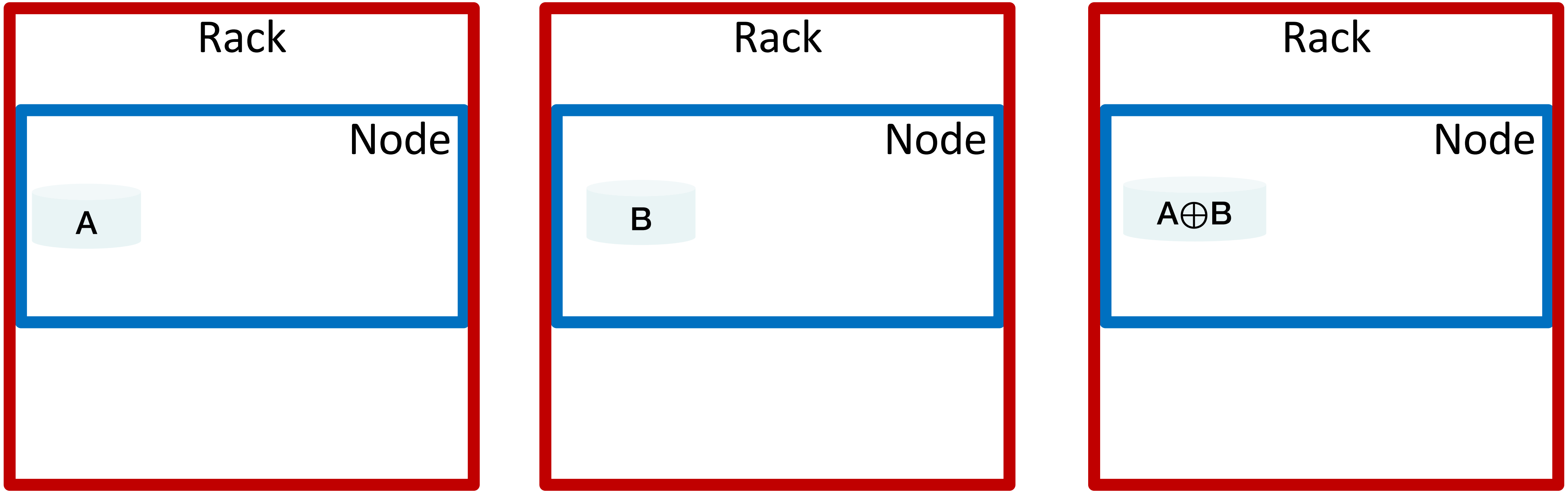
- Replication
  - Store multiple copies across multiple machines and data centers
  - Simple to implement (unless consistency is an issue)
  - Significant large storage overhead (2 times and more the size of the entire data)

- Erasure coding
  - Store ”codes data” to recover lost blocks in case of a failure.
    - For, example parity codes (🤔 we heard this before)
  - Less overhead
  - Combined with more reliable hardware, it can be a prime choice to balance cost, fault tolerance, and availabilty

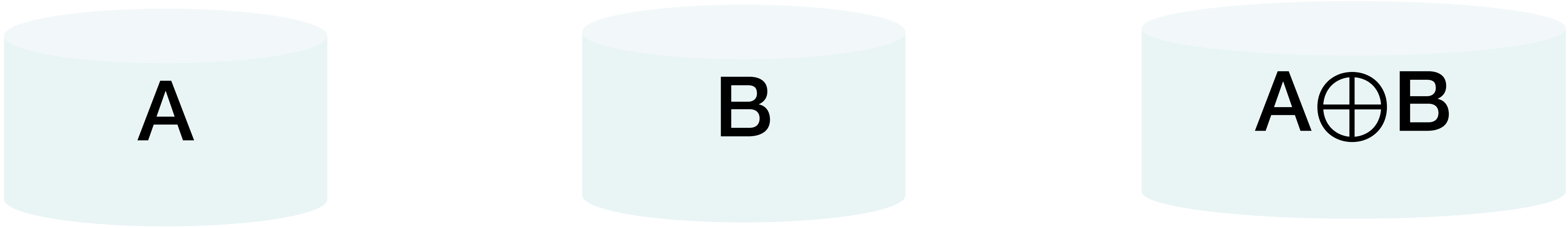
Erasure Codes using Bitwise XOR

/foo.csv

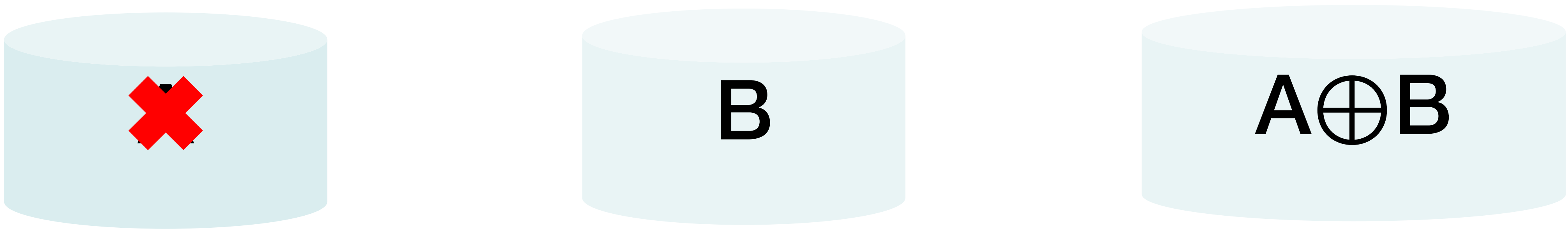
Block A
Block B



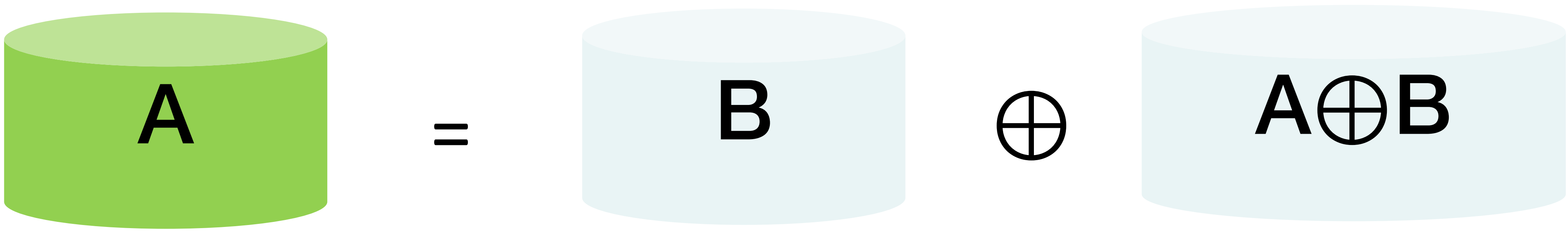
Erasure Codes using Bitwise XOR



Erasure Codes using Bitwise XOR



Erasure Codes using Bitwise XOR



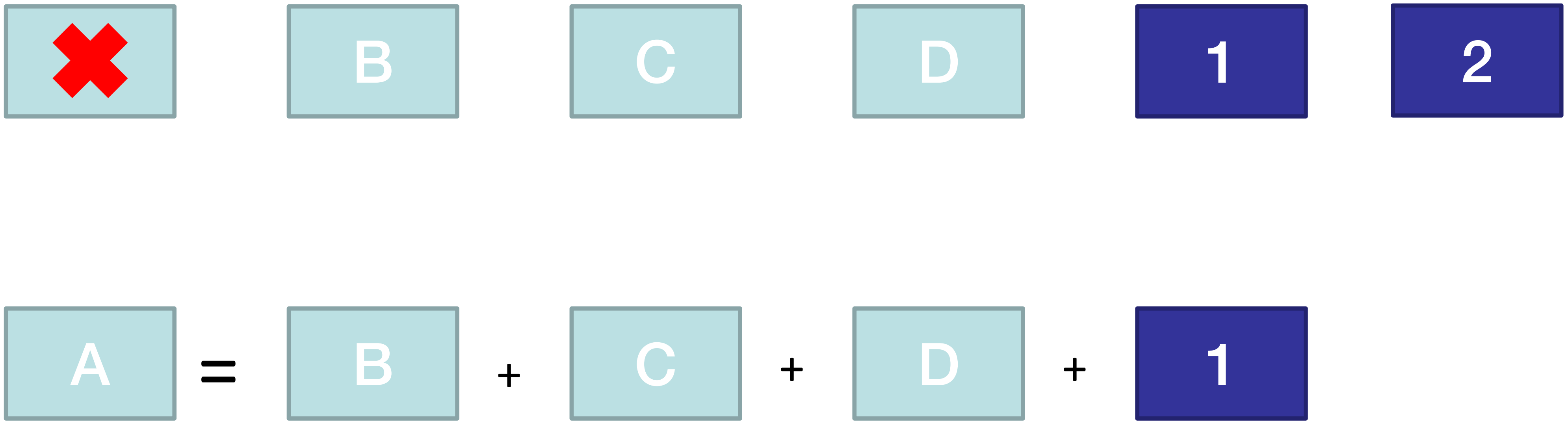
Reed-Solomon

- Given a specific setup:
  - N : the number of data blocks for a given file
  - K : the number coding blocks
  - M = N+K total blocks
- Characteristics:
  - Recover any 1 block from any N other blocks
  - Can withstand up to K concurrent failures
  - Applicable for any values of N and K
- Notation: RS(N,K)
  - (6,3): 6 data blocks, 3 coding blocks
- Technique:
  - Galois Field arithmetic. $GF(2^W)$

Reed-Solomon (4,2)

Reed-Solomon (4,2)

Reed-Solomon (4,2)



Reed-Solomon (4,2)