

# Far-and-Near: Co-Designed Storage Reliability Between Database and SSDs

Jinwoo Jeong, Kibin Park, Sangjin Lee, Philippe Bonnet,  
**Alberto Lerner**, and Philippe Cudré-Mauroux

CIDR'23 - Amsterdam



HANYANG UNIVERSITY

Embedded and Network Computing Lab.

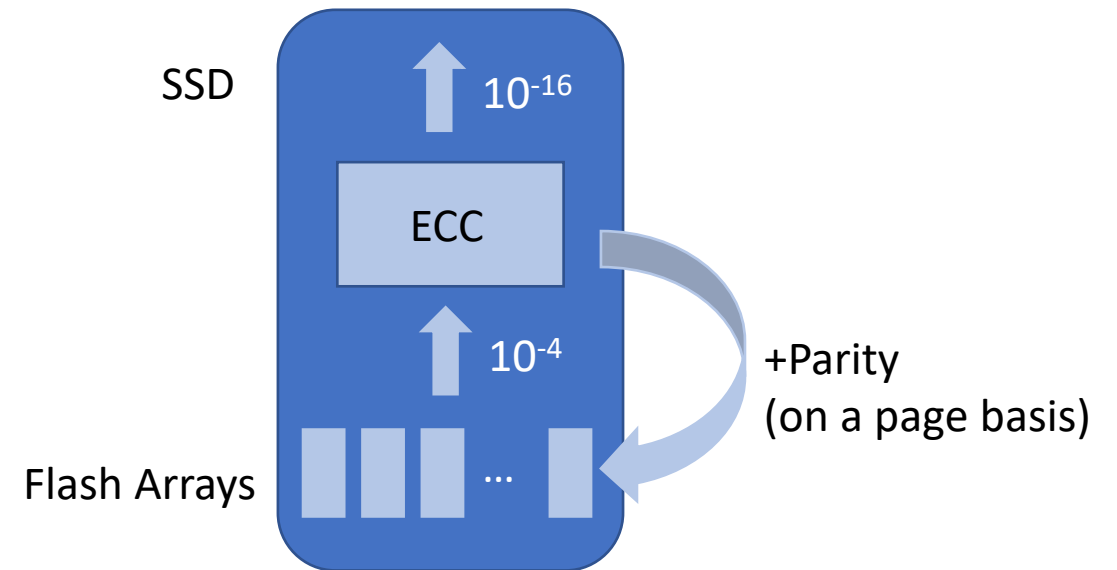
IT UNIVERSITY OF COPENHAGEN



eXascale Infolab

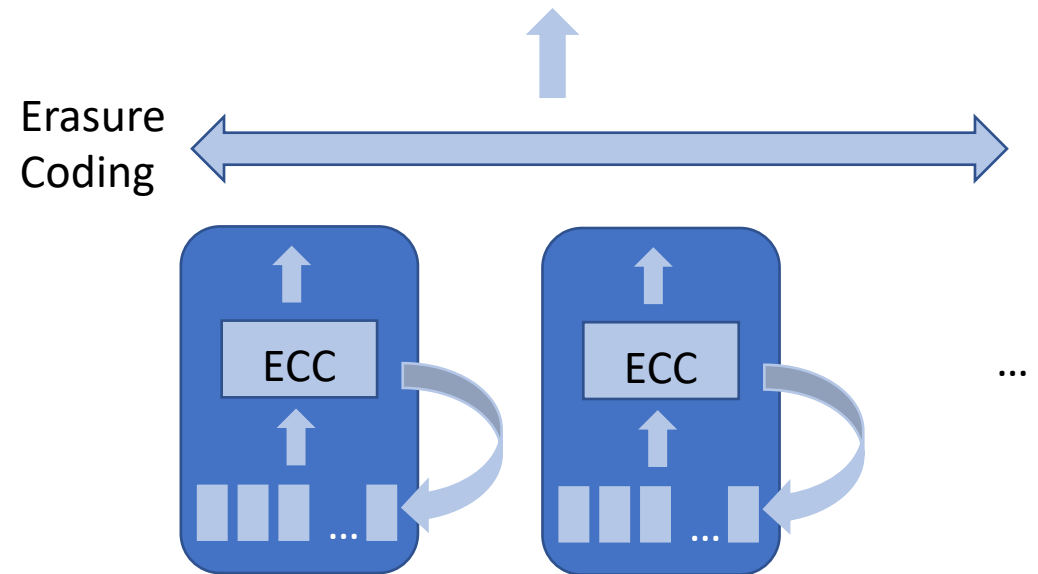
# ECC in SSDs

- NAND Flash is prone to errors
  - 1 bit flipped at every 10k read ( $10^{-4}$  bit error rate)
- Devices carry heavy machinery to deliver bit error rates of  $10^{-15}$  (consumer) or  $10^{-16}$  (enterprise)
  - Still, databases implement data correction measures atop of it
- Side effects of ECC: page size, latency, energy consumption,...
- Main issue: one-size-fits-all ECC
  - Different use cases have different requirements



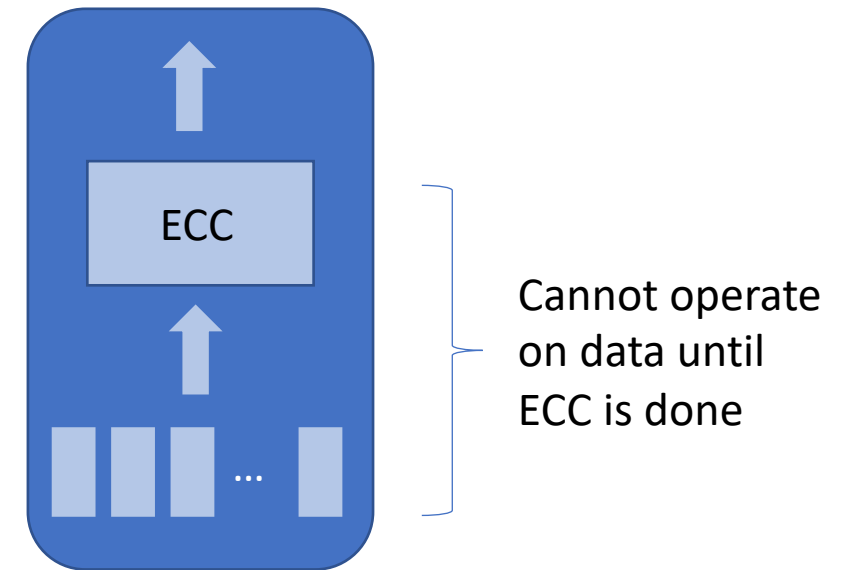
# “Far” Use Case

- "Far" as in the **data is manipulated very far away from the Flash array**
- Examples: fault-tolerant DB, cold storage
  - Implements erasure coding (e.g., Reed-Solomon) atop of SSD's ECC
- Issues
  - **Page size mismatch** (much, much larger in RS)
  - **Parity mismatch** (SSD doesn't benefit from RS nor vice-versa)



# “Near” Use Case

- "Near" as in the **data is processed before leaving the device**
- Example: near or in-storage processing (running **predicate/aggregation/etc closer to Flash**)
- Issues
  - Once again, **page size mismatch**: 16K needs to be decoded before it can be operated on
  - **Tremendous impact on latency, channel utilization, energy consumption, etc**



# Vision: Co-Designed ECC

- **DB (application) and device negotiate the right ECC scheme for each case**
  - Page size in terms of ECC, strength of ECC, etc
  - Negotiation can be as fine grained as on a stream basis
- **Transparency between hardware and software**
  - Device is informed if some page contains "application level" parity
  - Application is informed of size/contents of ECC
- **Benefits expected**
  - **Lower latency, higher throughput, better energy utilization**

Thanks!