HotStorage'26

GraceSSD: Per-File Reliability Hints for Error-Tolerant Machine Learning Storage

Seungkwan Kang, Hyungseok Ko, Hyun Dong Lee, Miryeong Kwon,
Myeonghoon Park, Jinseok Kim, Myoungsoo Jung

KAIST CAMEL panmnesia

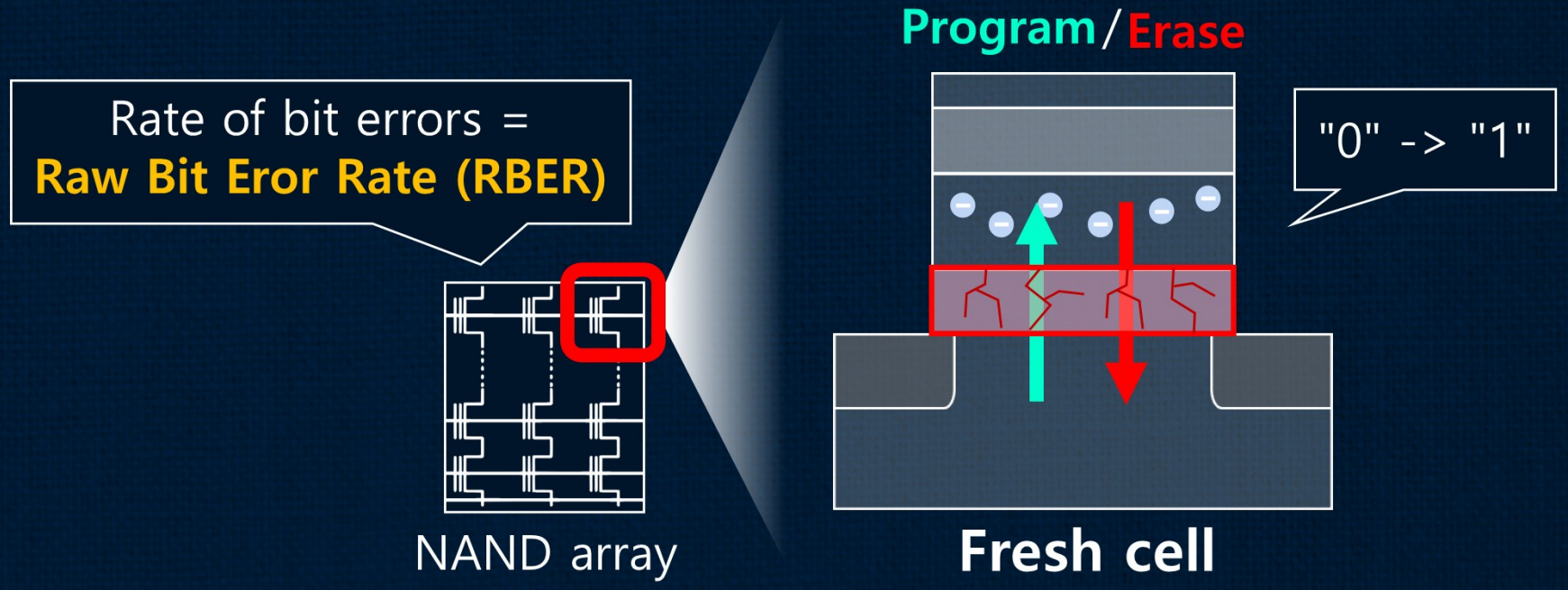
1. Modern SSD Error Correction

2. Observation - Bit Error Resilience

3. Design and Implementation

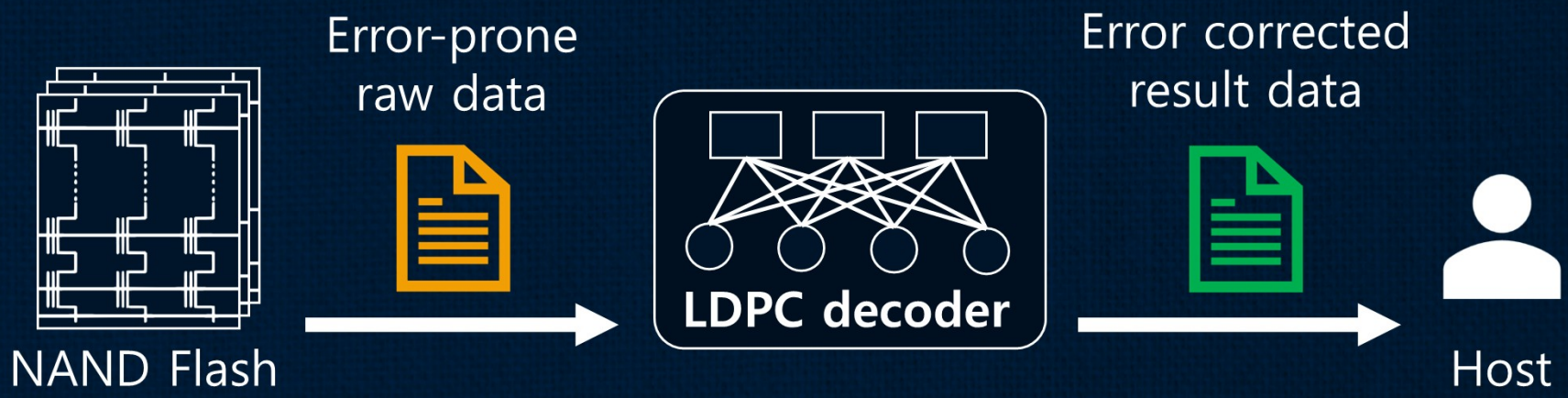
4. Evaluation Results

NAND Flash Wears Out



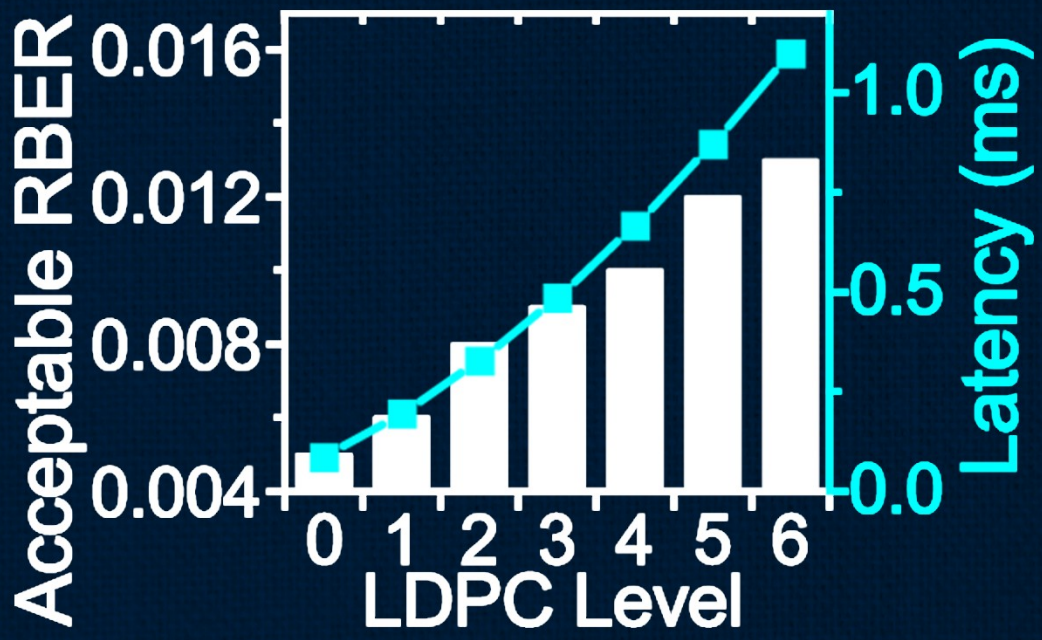
Raw Bit Error Rate (RBER) rises with the P/E cycles

Error Correction with LDPC



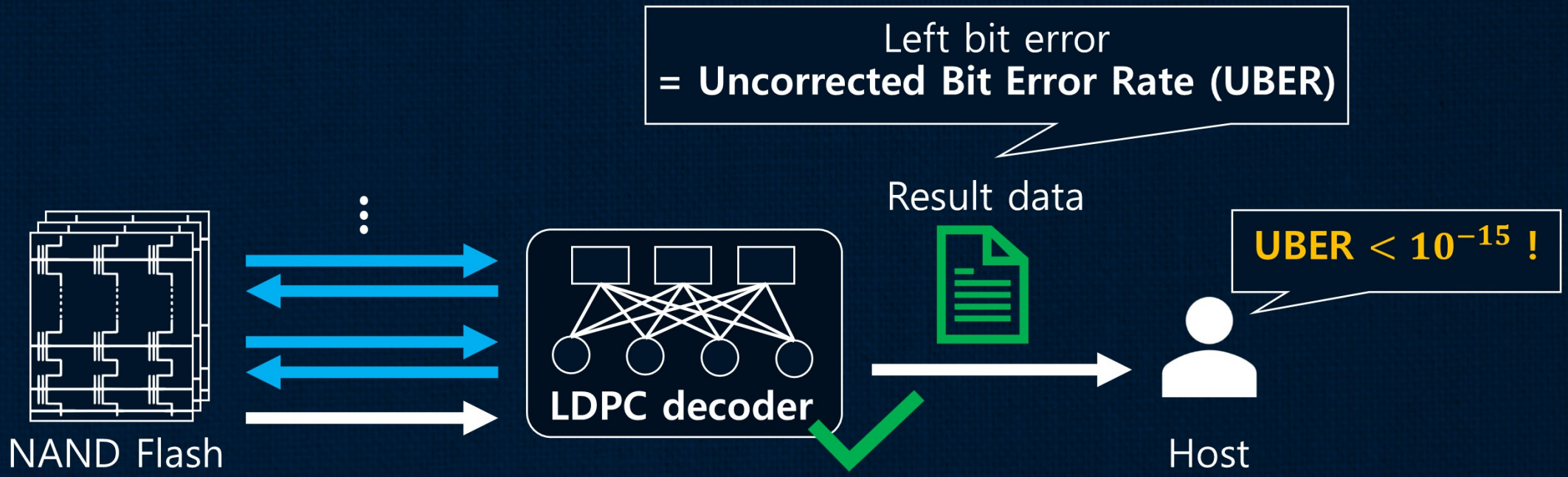
LDPC decoder tries to fix the errors with parity equations

Error Correction with LDPC



High LDPC level can fix more error at the cost of latency

Error Correction with LDPC



Today's SSDs target an UBER of 10^{-15} (JEDEC standard)

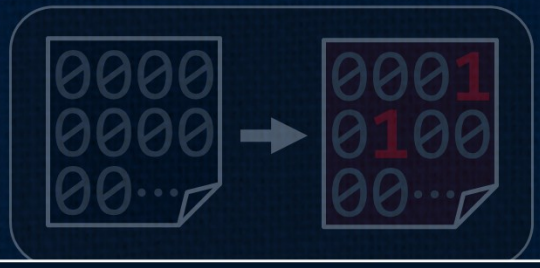
1. Modern SSD Error Correction

2. Observation - Bit Error Resilience

3. Design and Implementation

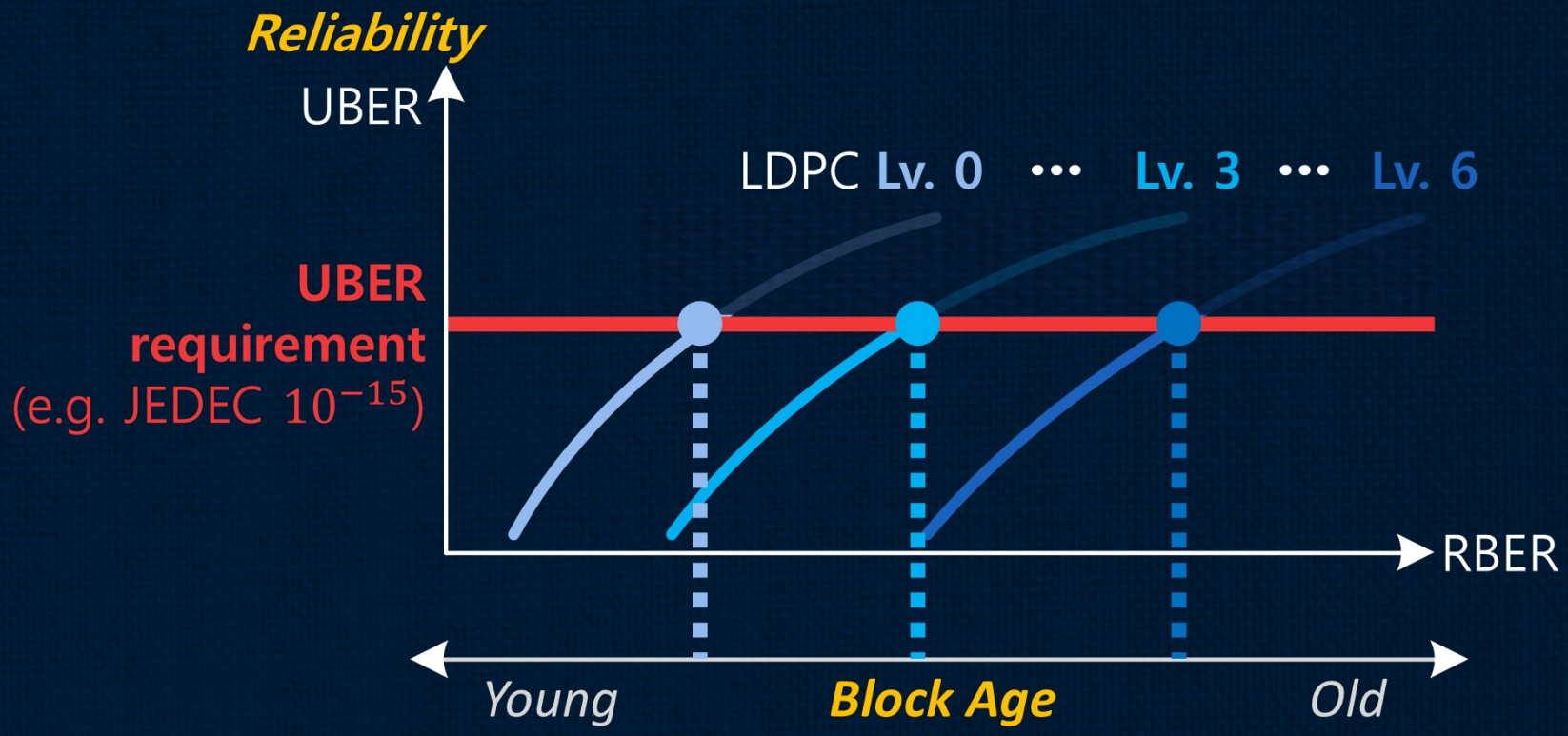
4. Evaluation Results

Bit Error Resilience



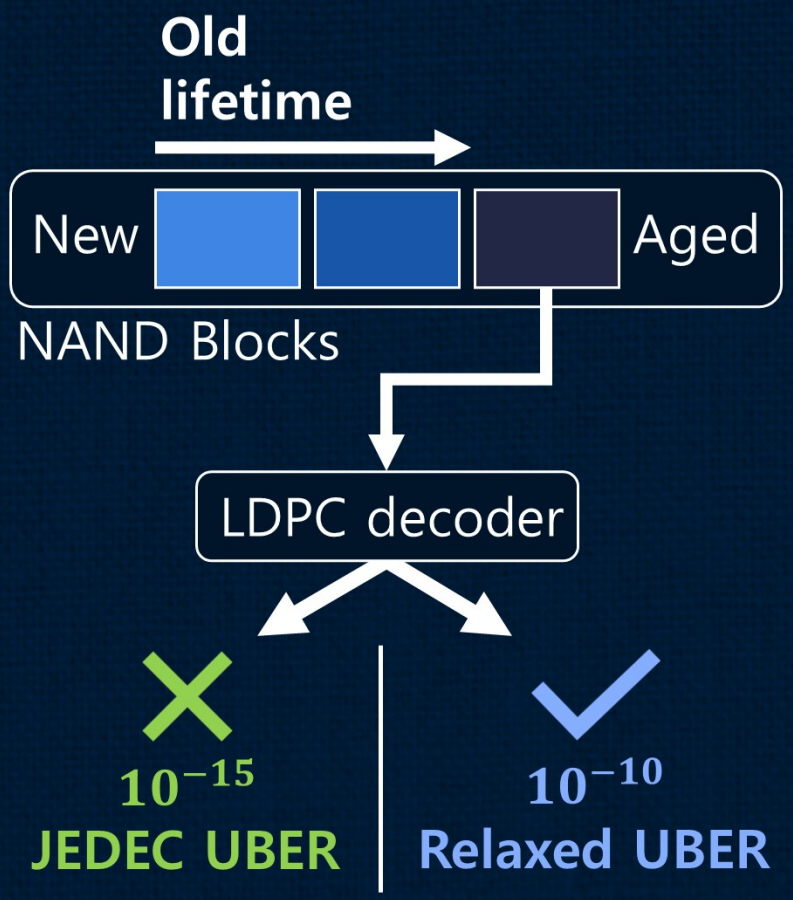
We will utilize bit error resilience to make SSD efficient by reliability relaxation

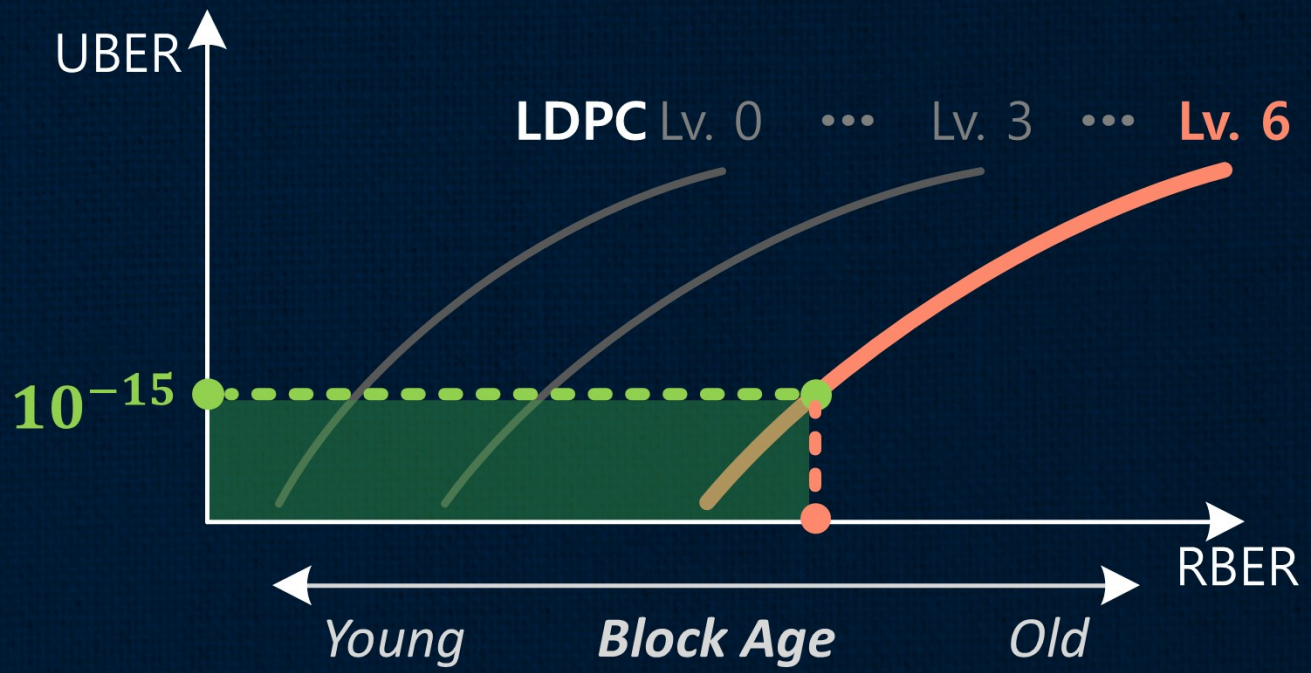
Reliability Relaxation



To visualize the effect of reliability relaxation, we use UBER-RBER plot

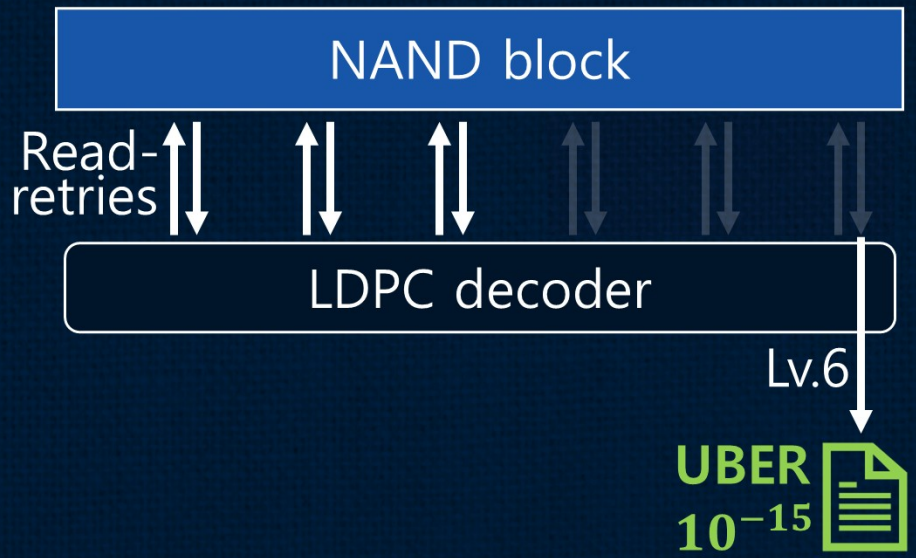
Reliability Relaxation

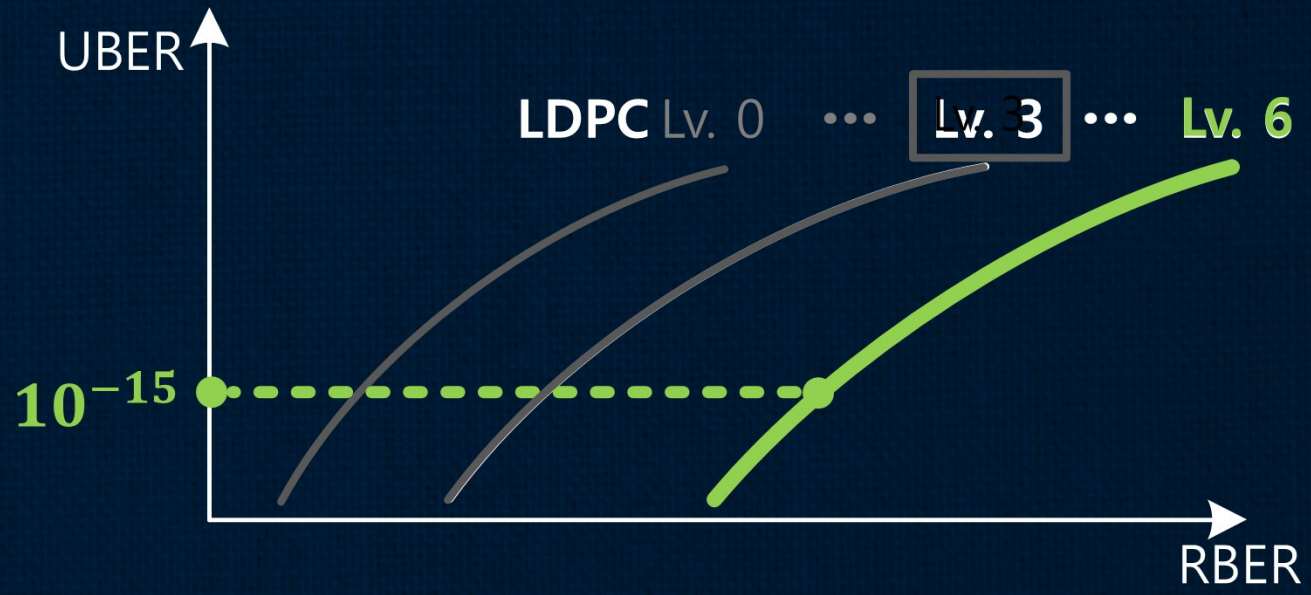




SSD can utilize aged NAND block longer - "Lifetime extension"

Reliability Relaxation





SSD can use lower LDPC level - "Performance enhancement"

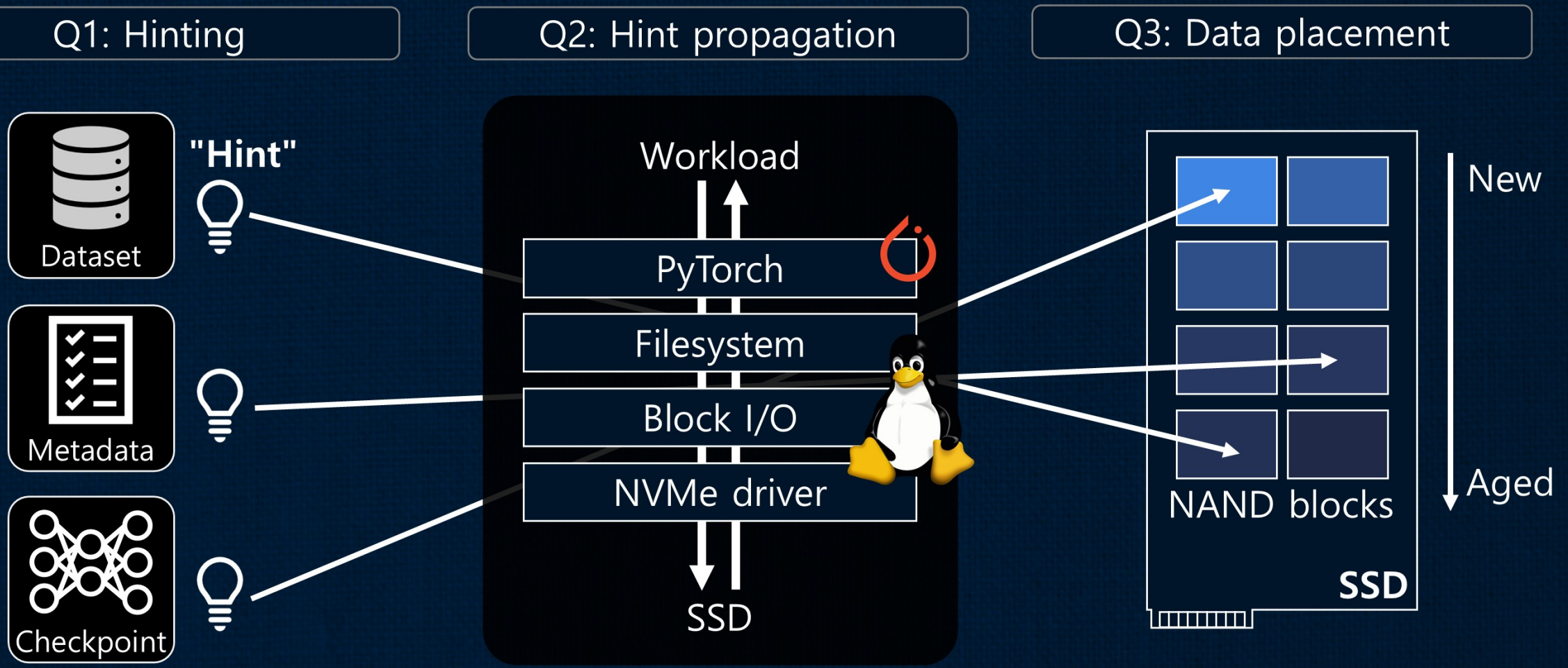
1. Modern SSD Error Correction

2. Observation - Bit Error Resilience

3. Design and Implementation

4. Evaluation Results

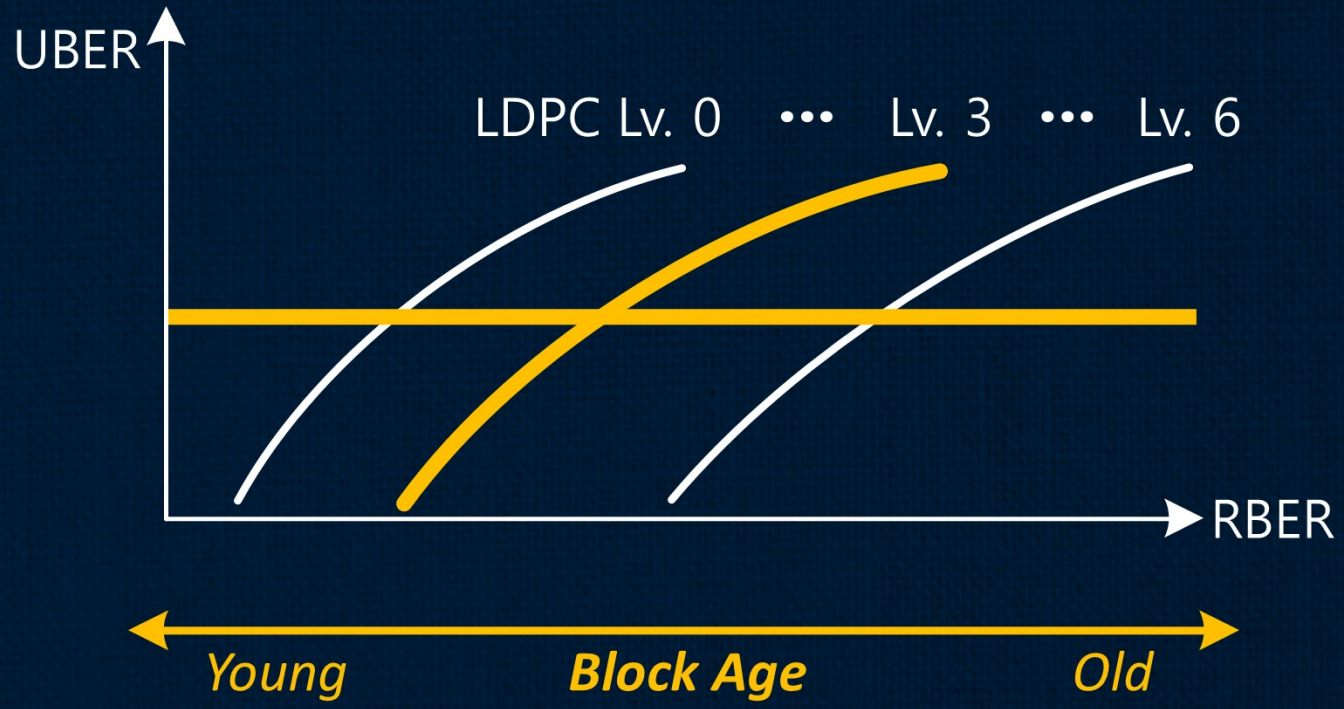
GraceSSD Overview



Serve various data with different requirements within a single SSD

Q1: Hints for Placement

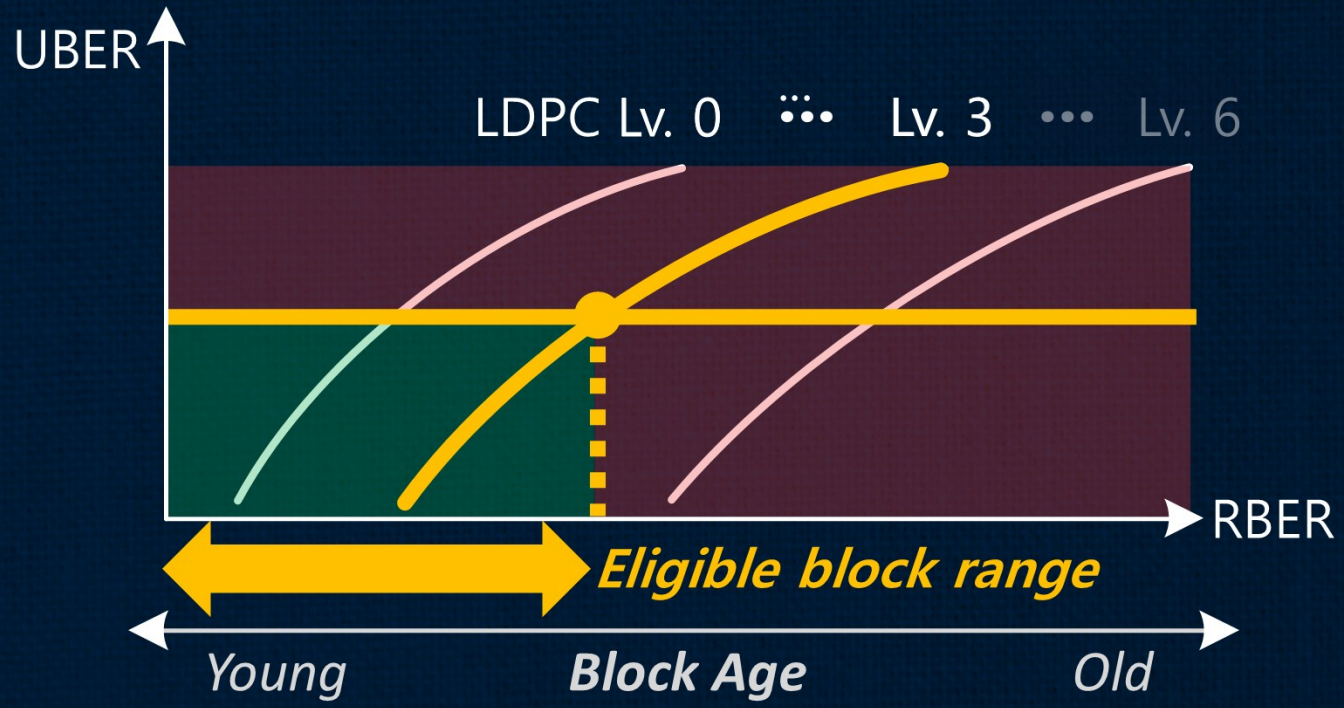
Q1: Hinting | Q2: Hint propagation | Q3: Data placement



SSD only know P/E cycles – Need to derive eligible P/E cycle range

Q1: Hints for Placement

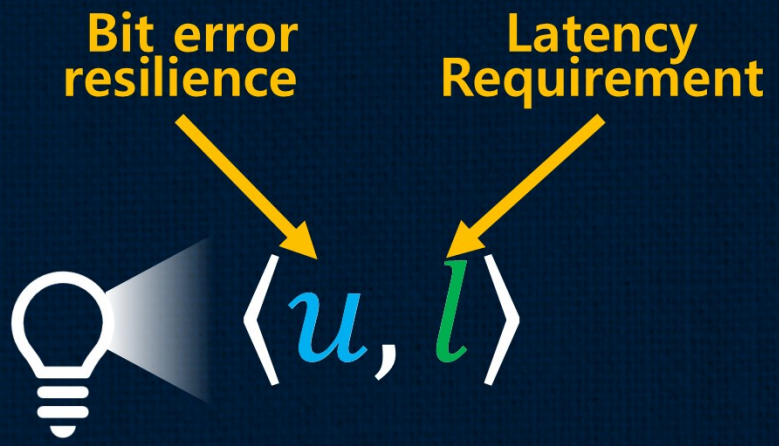
Q1: Hinting | Q2: Hint propagation | Q3: Data placement



SSD only know P/E cycles – Need to derive eligible P/E cycle range

Q1: Hints for Placement

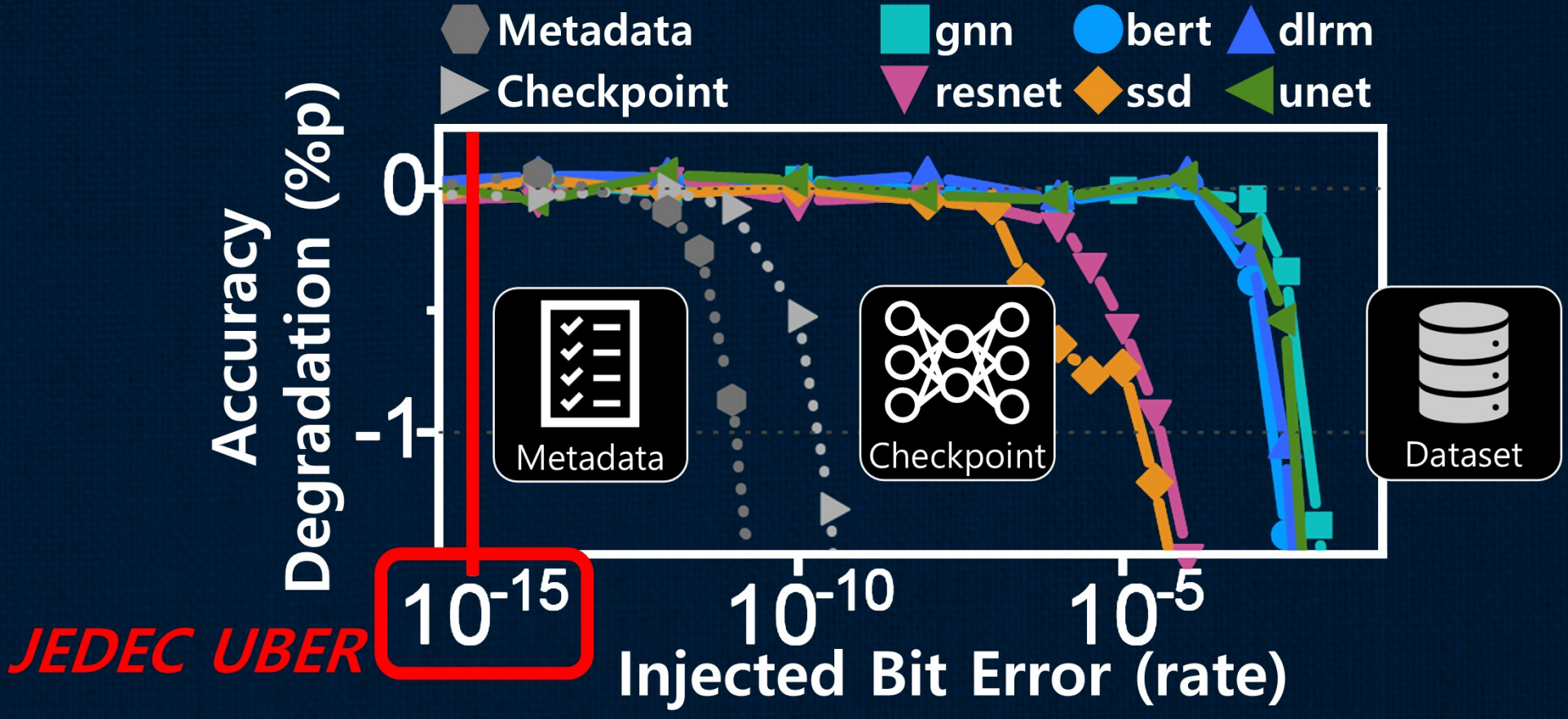
Q1: Hinting | Q2: Hint propagation | Q3: Data placement



Two hints - "Bit error resilience" & "Latency requirement"

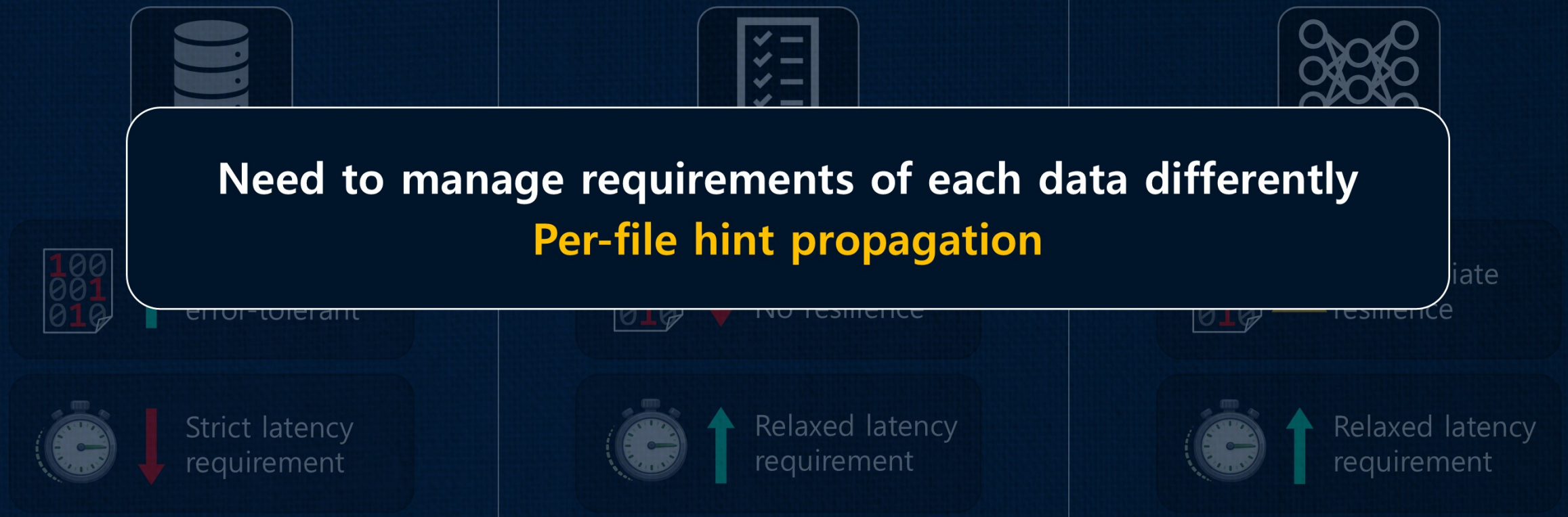
Q1: Hints for Placement - ML Training

Q1: Hinting | Q2: Hint propagation | Q3: Data placement



Variance in tolerance necessitates managing each data differently

Q1: Hints for Placement - ML Training

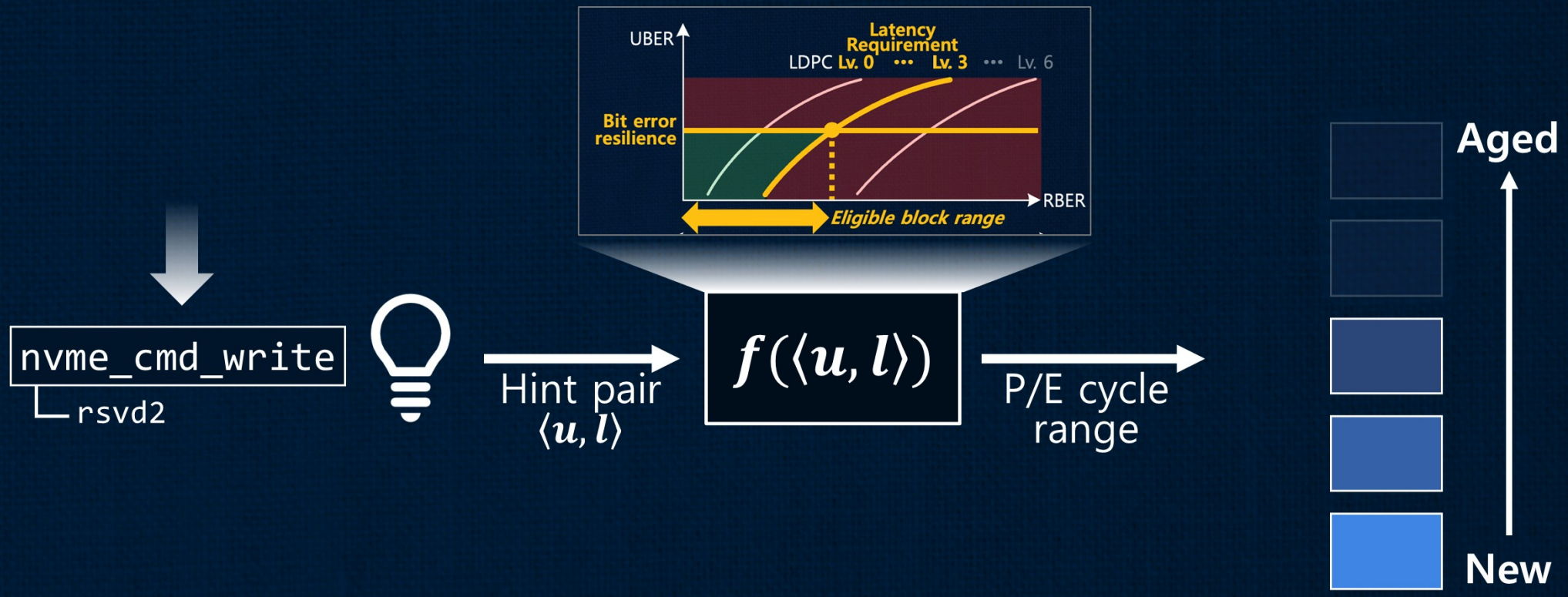


Q2: Hint Propagation



Hint travels from xattr to SSD through kernel I/O stack

Q3: Hint-based Data Placement



Find eligible blocks within P/E cycle range estimated from the hint pair

1. Modern SSD Error Correction

2. Observation - Bit Error Resilience

3. Design and Implementation

4. Evaluation Results

Methodology

Workloads - MLPerf Training



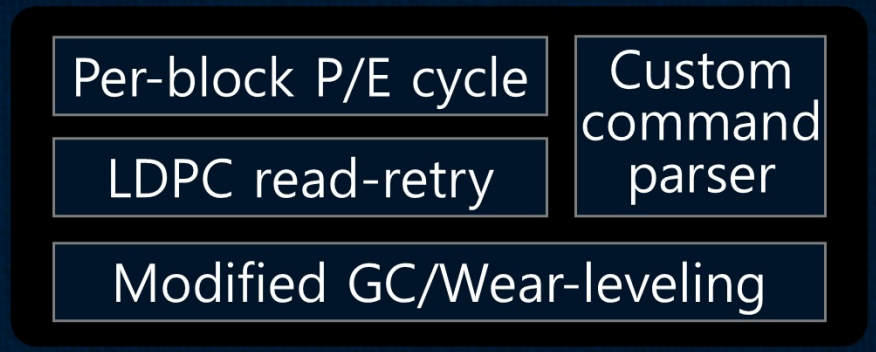
Configurations



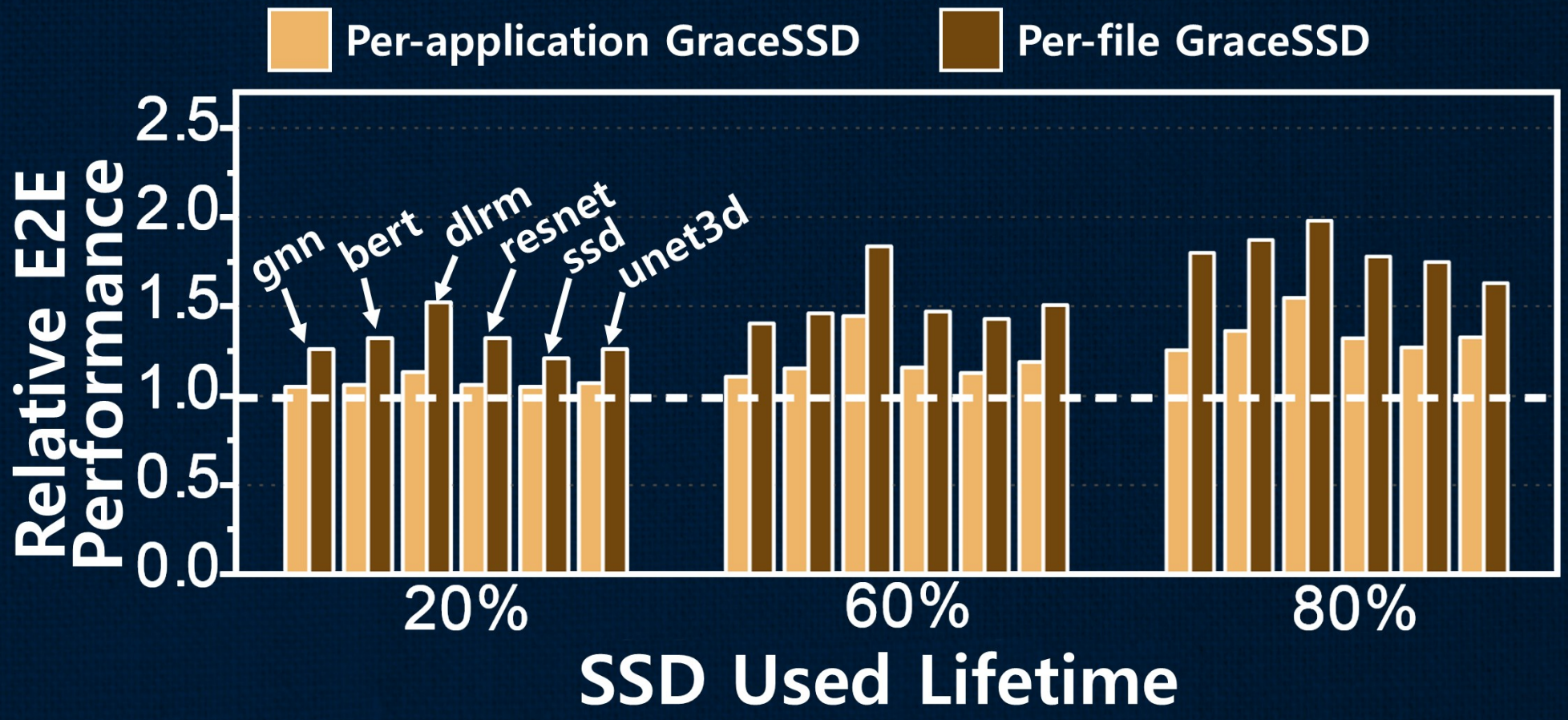
Testbed configuration

Software	Kernel	v5.15.0
	PyTorch	v23.09-nv
Hardware	CPU	EPYC 7R13
	GPU	H100 x8
	Memory	2TB

Modified NVMeVirt

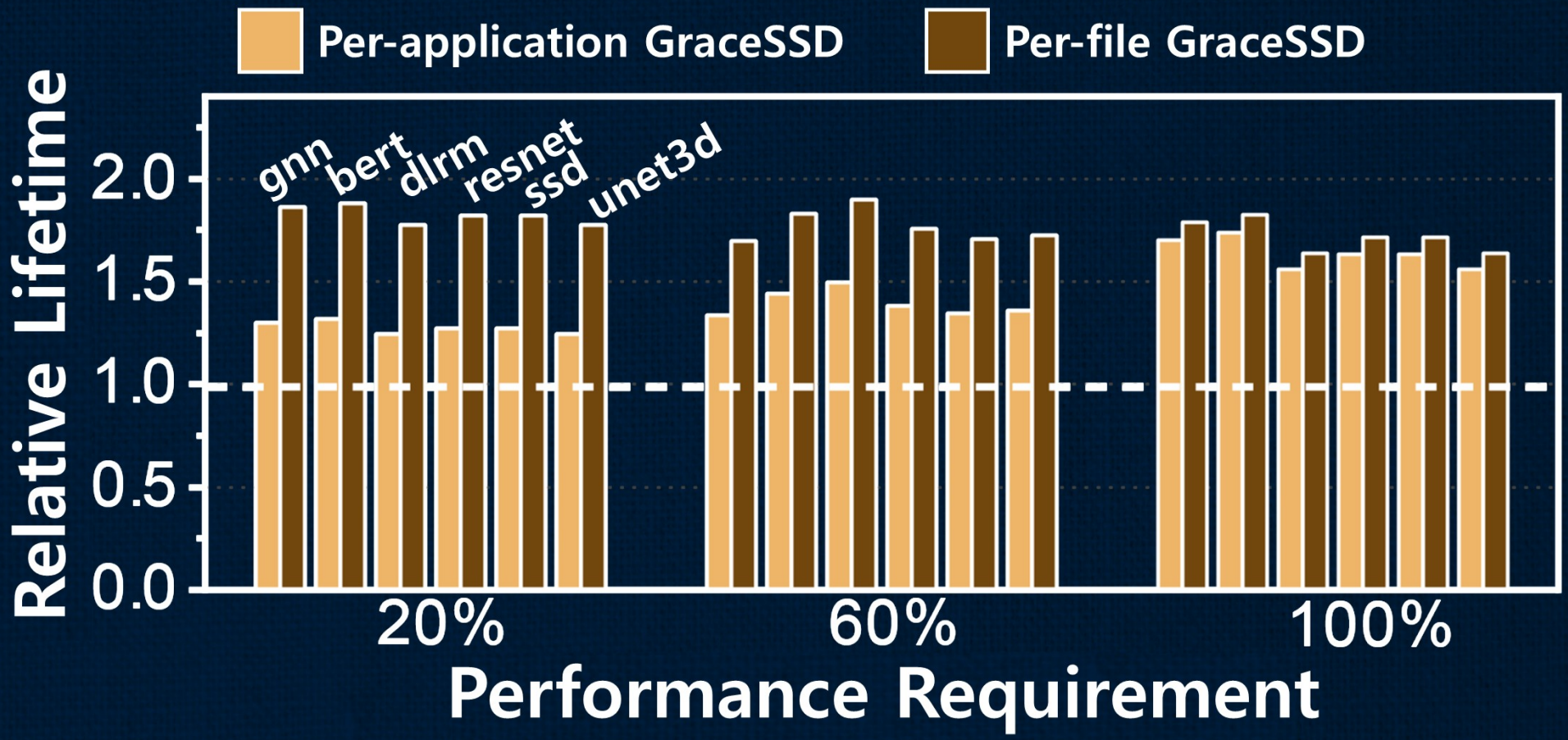


Performance Enhancement



GraceSSD delivers **54.3%** higher training throughput on average

Lifetime Enhancement



GraceSSD extends lifetime 76.7% on average at full performance

Summary

- We identified that SSDs enforce **one strict reliability target** on every page.
- This costs retries and early block retirement that ML data does not need.

- We proposed **GraceSSD**, a per-file hint of the error and latency a file tolerates.

- GraceSSD carries the hint from **file's xattr down to the NVMe command**.
- GraceSSD **relaxes the bit error integrity per-file**, so aged blocks keep serving tolerant data.
- GraceSSD lets data with different requirements **share one SSD** without interference.

- We achieved **76.7% longer lifetime** and **54.3% higher throughput** on average, without accuracy loss.

Thank You



KAIST CAMEL panmnesia