LaLDPC: Latency-aware LDPC for Read Performance Improvement of Solid State Drives

Yajuan Du[1,2], Deqing Zou[1], Qiao Li[3], Liang Shi[3], Hai Jin[1], and Chun Jason Xue[2]

[1]Huazhong University of Science and Technology

[2]City University of Hong Kong

[3]Chongqing University

Outline

- Background and Motivation
- Design of LaLDPC
- Evaluations
- Summary

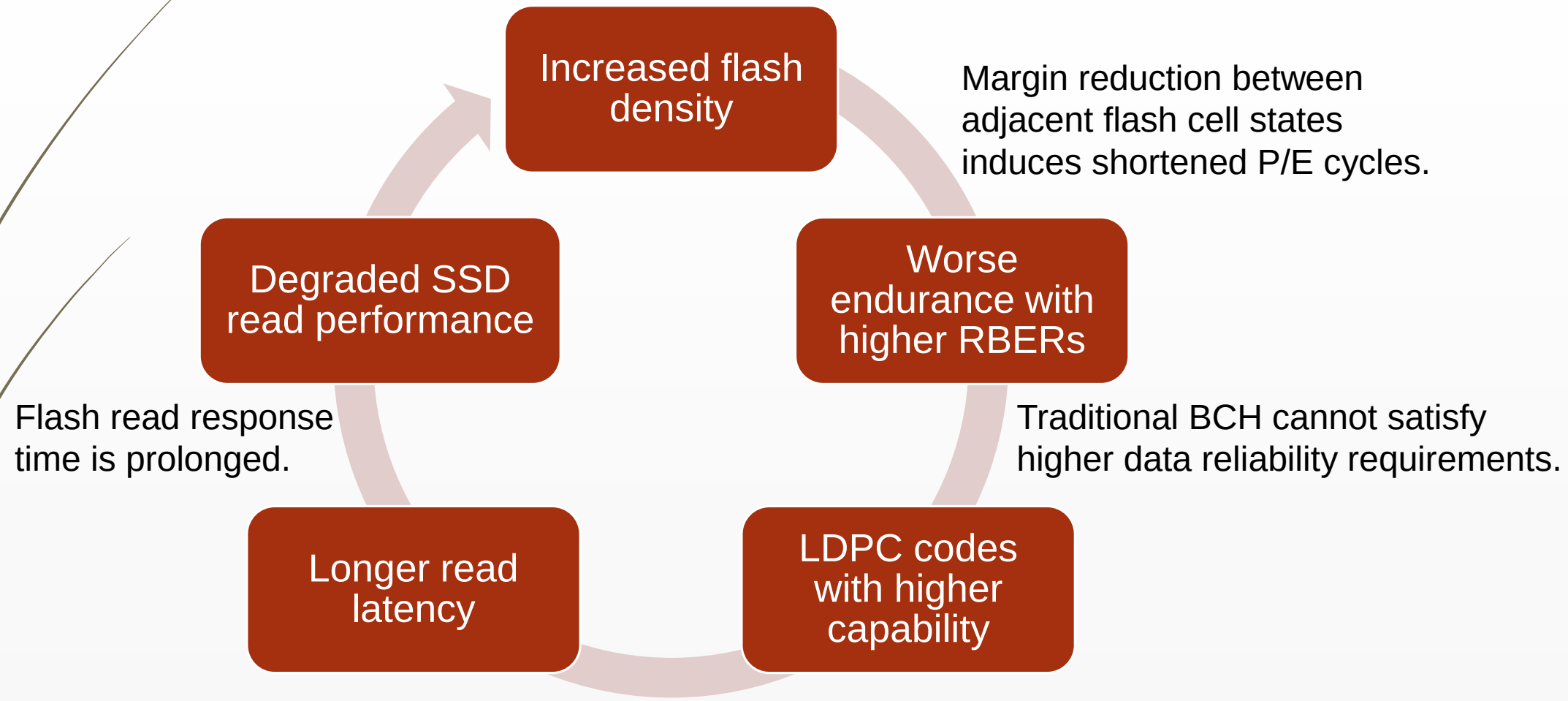
Outline

- ***Background and Motivation***
- Design of LaLDPC
- Evaluations
- Summary

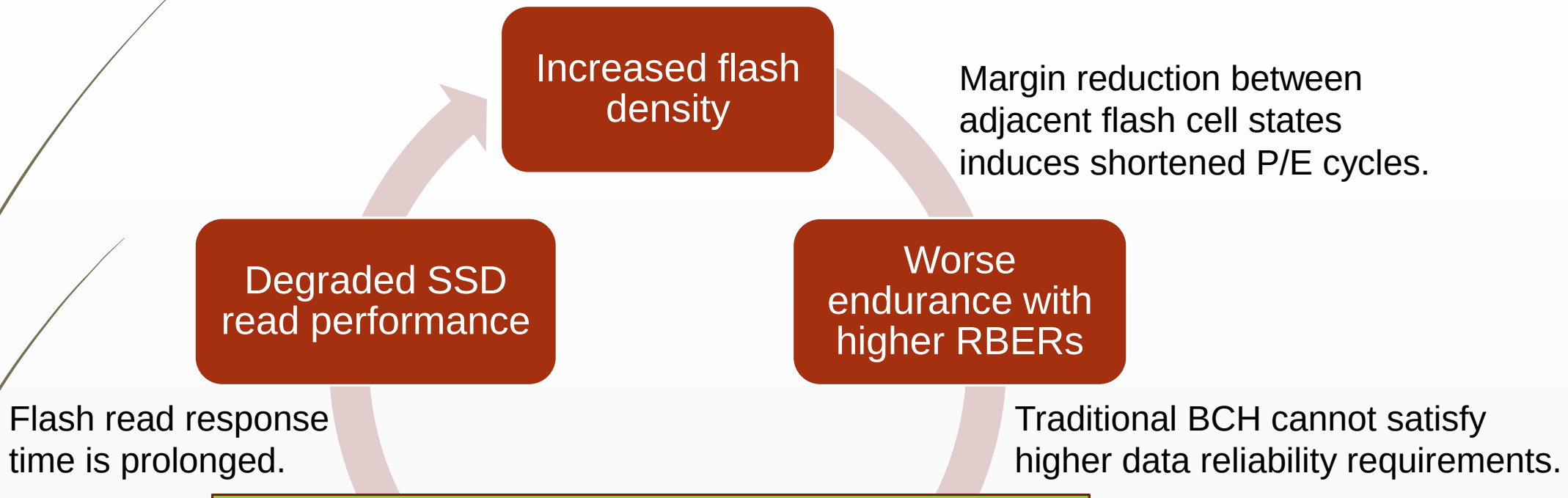
Popular Productions and Applications of Flash-based SSDs

- SSDs are widely deployed into mobile phones and personal computers;
- Advantages of flash-based SSDs: non-volatility, shock resistance, high speed and low energy consumption;
- High-density flash memories, such as TLC and 3D flash, are developed to decrease the price of SSDs.

Degraded Read Performance of High-density Flash Memories



Degraded Read Performance of High-density Flash Memories

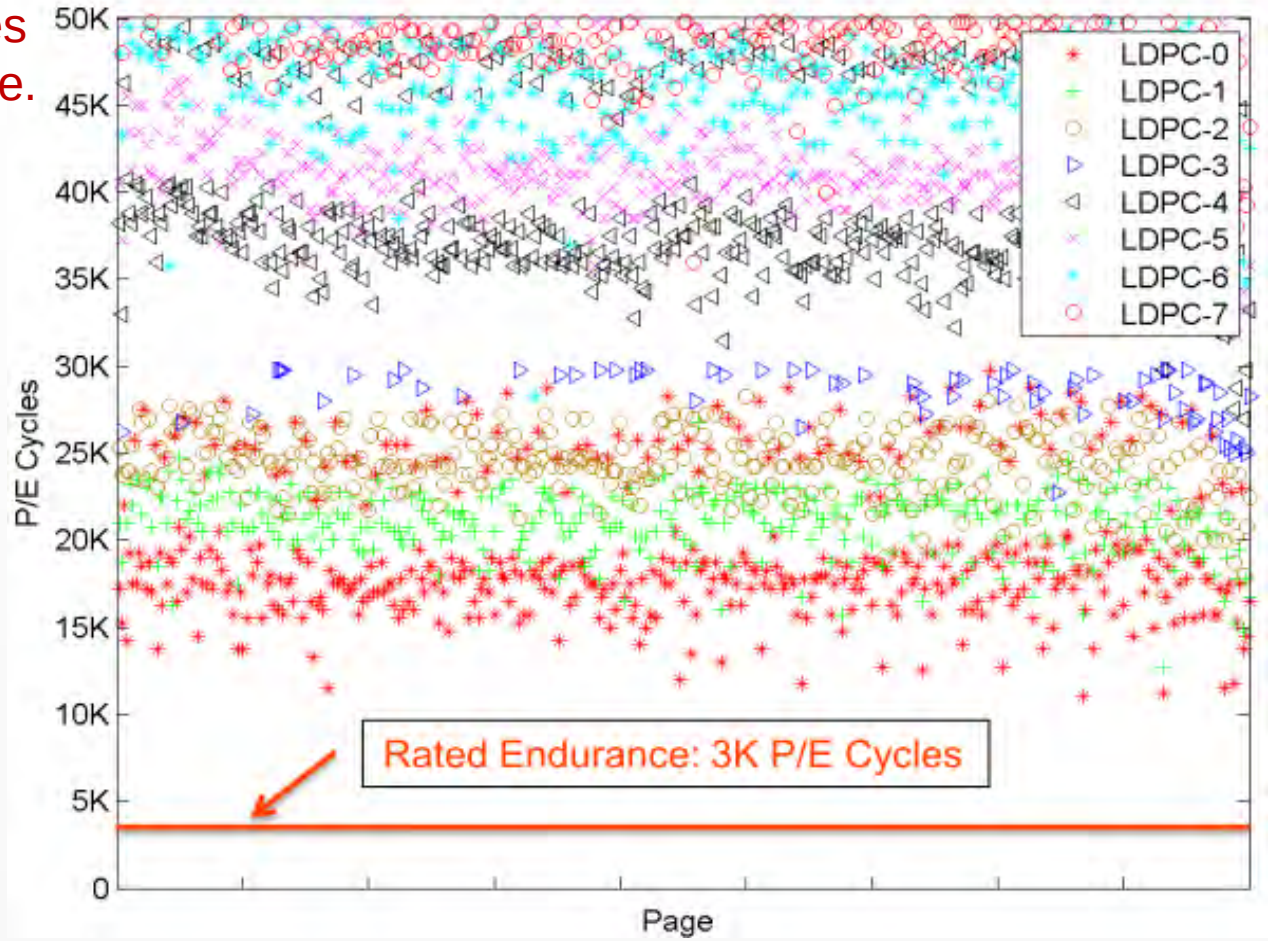


Our work focuses to Improve this relationship.

More sensing times are needed for successful decoding.

Error Correction Capability of LDPC Codes (1/2)

Higher-capability LDPC codes ensure better flash endurance.

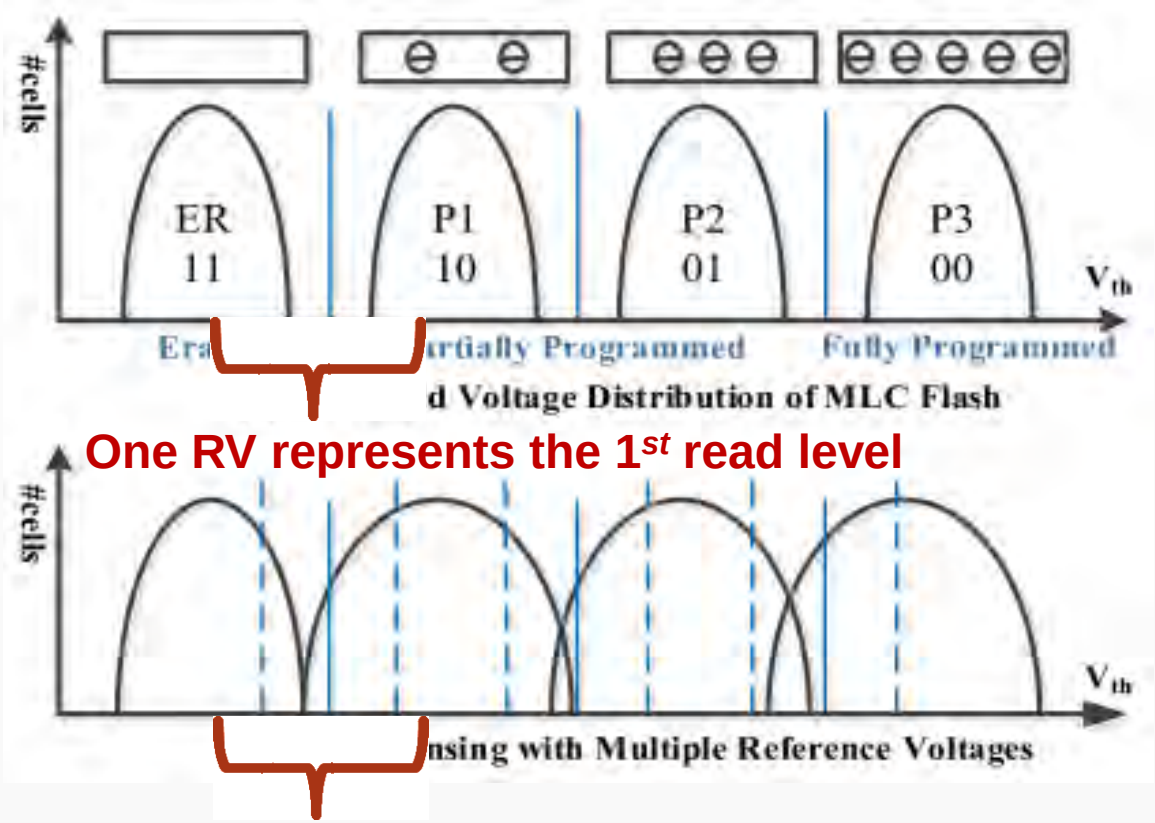


Source: Flash memory summit, 2014, Erich F. Haratsch,
LDPC Code Concepts and Performance on High-Density Flash Memory

Error Correction Capabilities of LDPC Codes (2/2)

- Error correction capabilities of LDPC codes closely relate to ***read levels*** in flash sensing;
- Read level equals to one third of number of ***reference voltages (RVs),*** which is exactly RV number between adjacent states.

$$\mathrm{RL} = Num.of\ RVs/3$$



LDPC Read Level vs. Read Latency

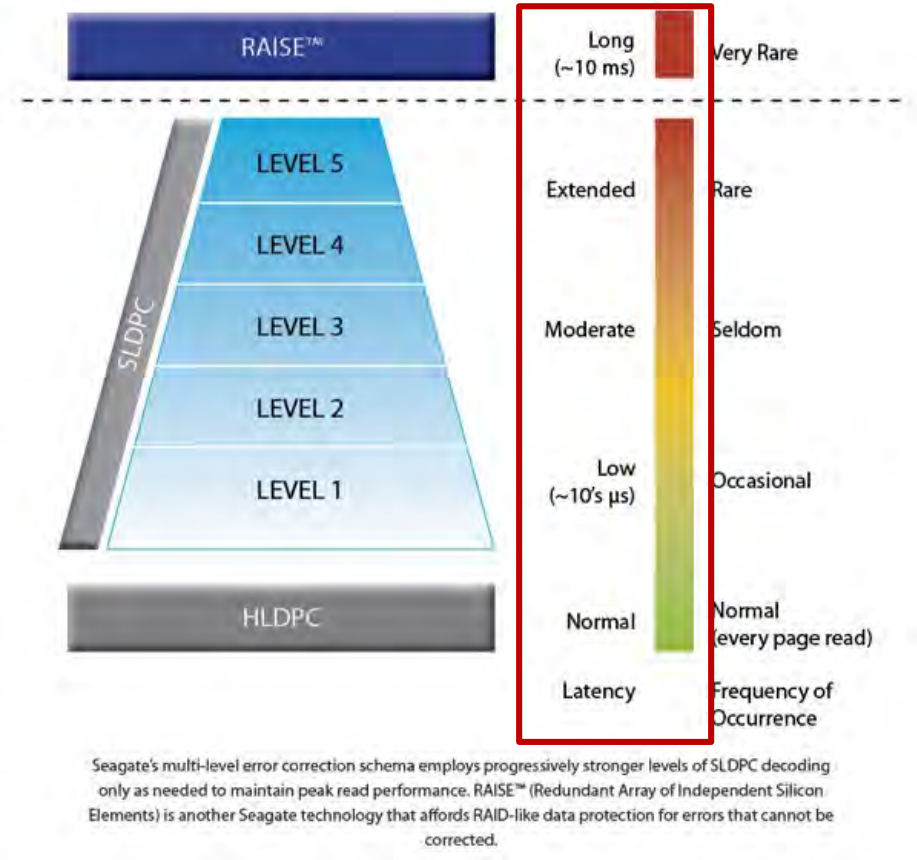
Read level ↑ 🙂

LDPC capability ↑ 🙂

Read latency ↑ ☹

High read level provides higher error correction capability but induces read performance degradation!

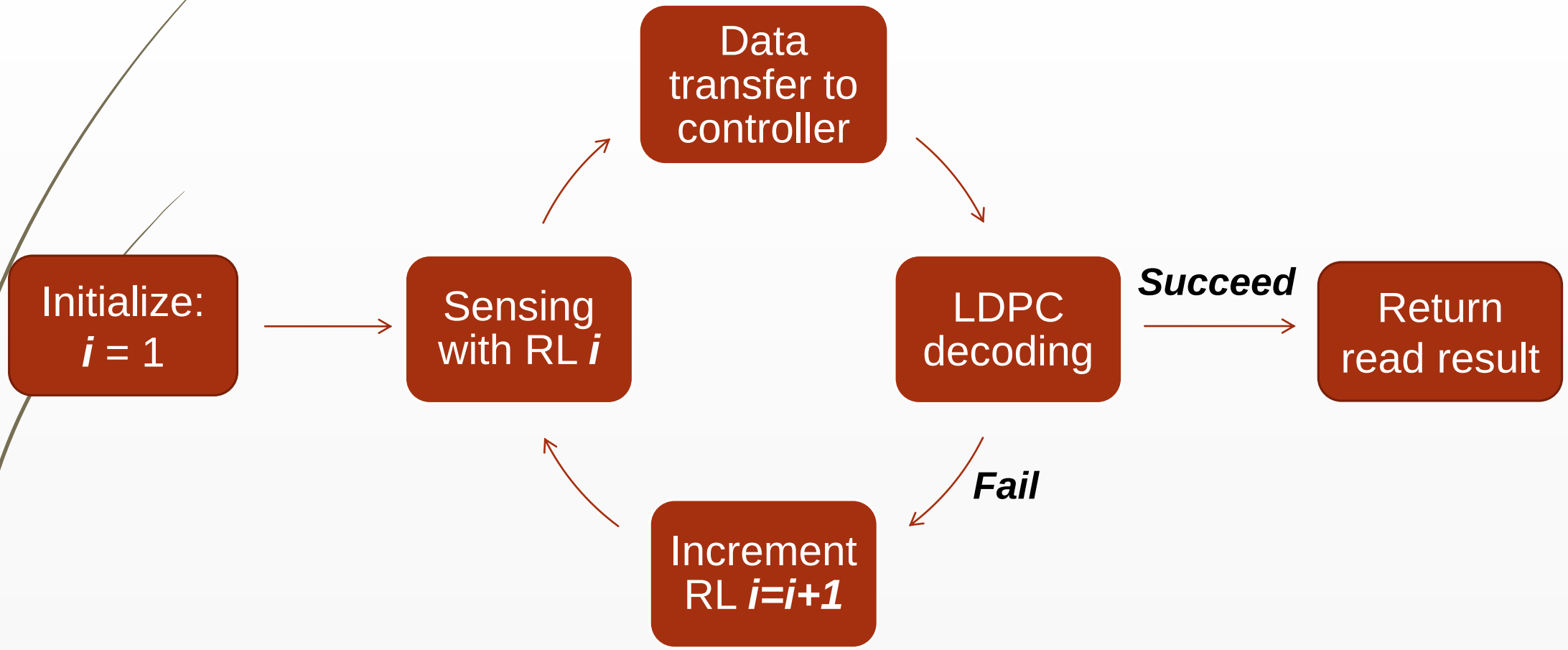
Read latency increases along with read levels



Seagate's multi-level error correction schema employs progressively stronger levels of SLDPC decoding only as needed to maintain peak read performance. RAISE™ (Redundant Array of Independent Silicon Elements) is another Seagate technology that affords RAID-like data protection for errors that cannot be corrected.

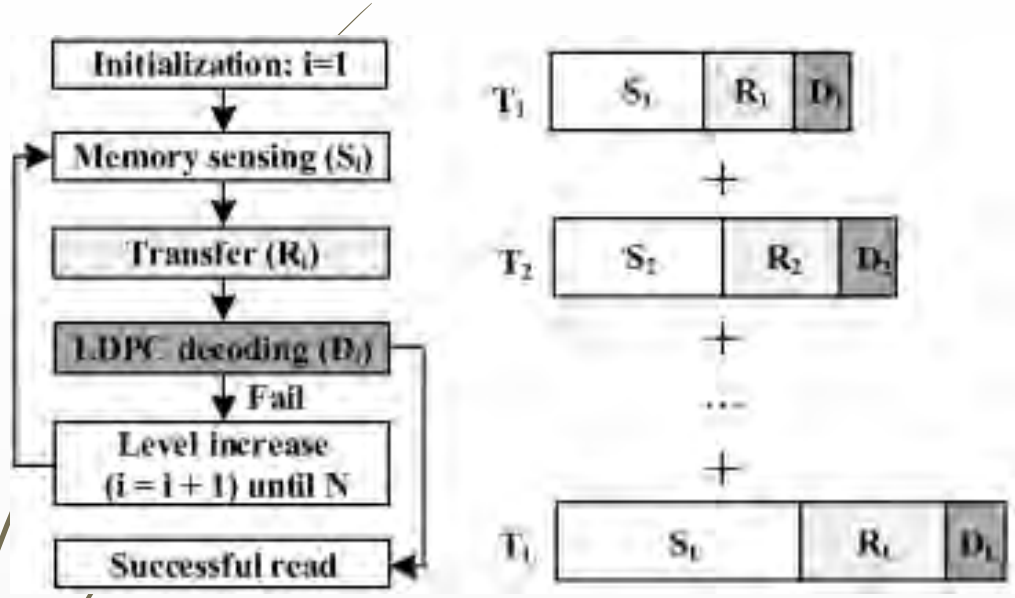
Source: Seagate error correction technlogy, http://www.seagate.com/cn/zh/tech-insights/shield-technology-master-ti/

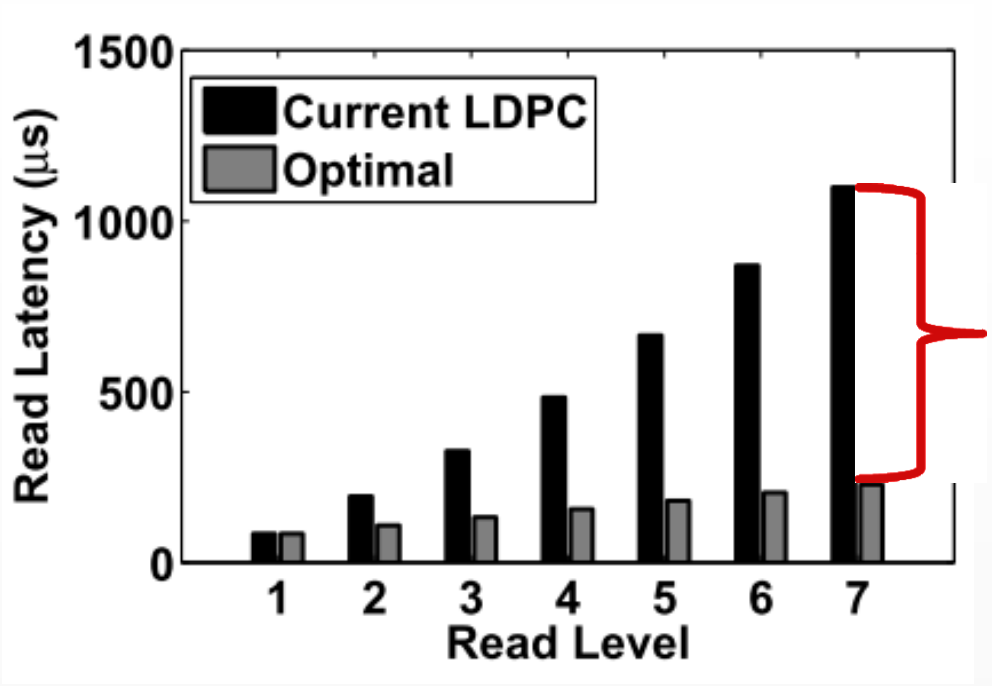
Current Progressive Read-retry LDPC Implementation



Reference: Zhao et. al., FAST 2012

Latency Accumulation Problem—double increases



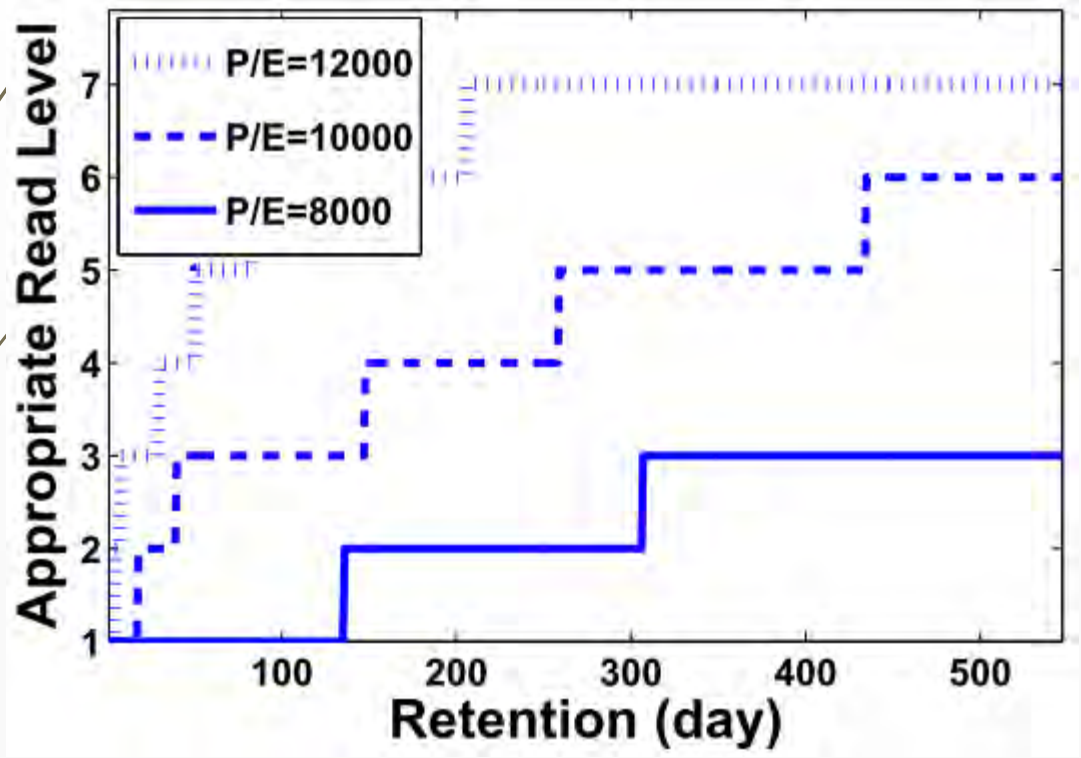


Overall latency is $\sum_{i=1}^{L} T_i$ with $T_i = S_i + R_i + D_i$

There is a large latency gap between LDPC reads with high read levels and the optimal case.
Gap causes: 1) higher latency for higher read level; 2) accumulation of read levels.

We aim to find the optimal read level and narrow this latency gap!

Observation: Temporal Read Level Locality of LDPC Codes



$$\begin{cases} \mu = K_0 K_1 (x - x_0) ln(1 + t/t_0) N^{0.5} \\ \sigma^2 = K_0 K_2 (x - x_0) ln(1 + t/t_0) N^{0.6} \end{cases}$$

Gaussian error model with parameters: $K_0 = 0.333$, $K_1 = 4 \times 10^{-4}$, $K_2 = 2 \times 10^{-6}$ and $x_0 = 1.4$

Reference: Pan et. al., HPCA 2012

The read level for one page lasts for a long time, during which all reads have the same read level, called ***temporal read level locality***.

Outline

- Background and Motivation
- ***Design of LaLDPC***
- Evaluations
- Summary

LaLDPC: Exploiting Temporal Read Level Locality

LaLDPC objective

- A new decoding scheme to assist LDPC decoders and to solve read latency accumulation

Basic idea of LaLDPC

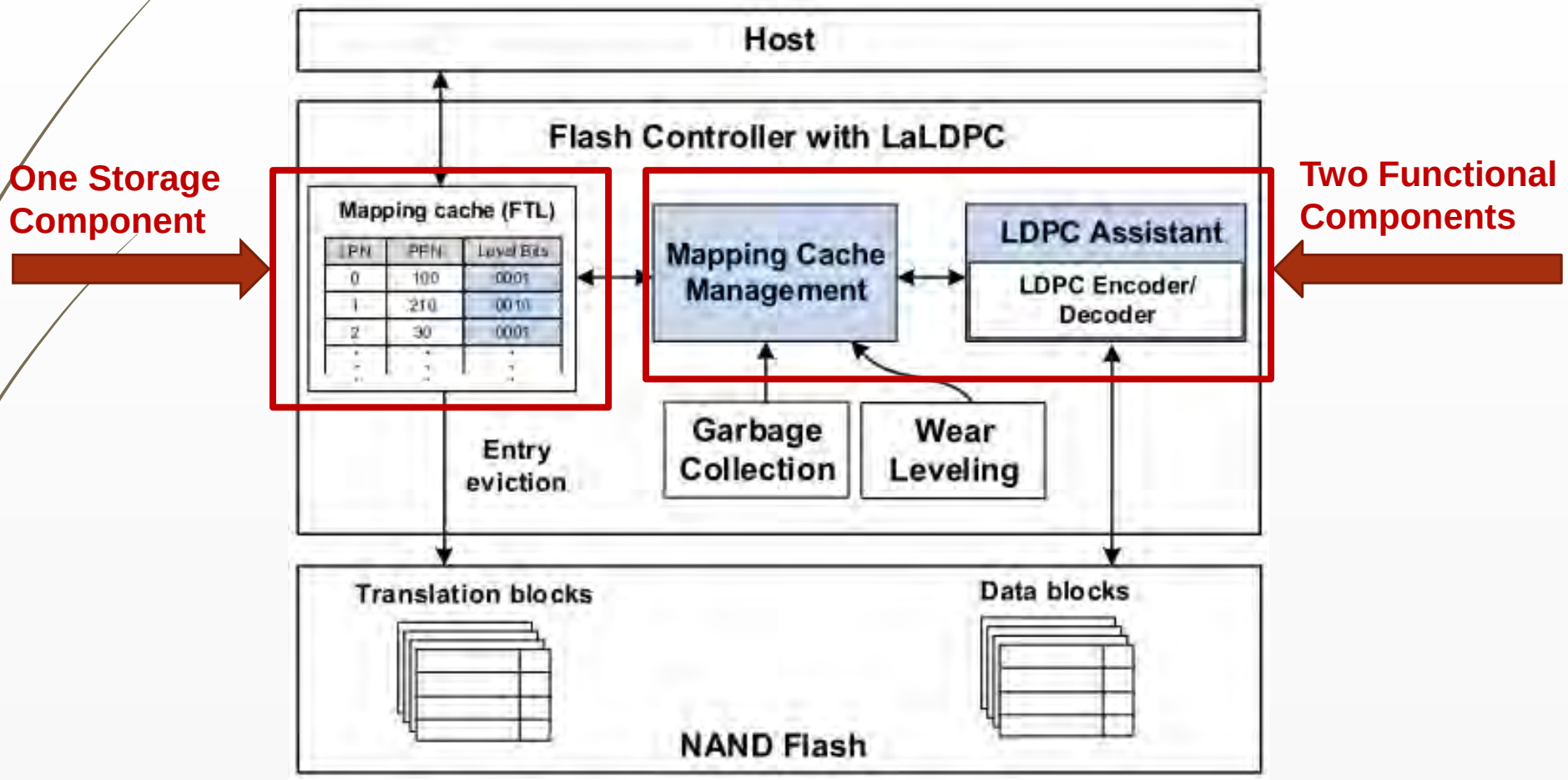
- Store the LDPC read level of previous reads for each page;
- Apply stored read level as the beginning level of LDPC read-retry process in the following reads.

Questions

- Where to store the read levels?
- When and how to use these read levels?

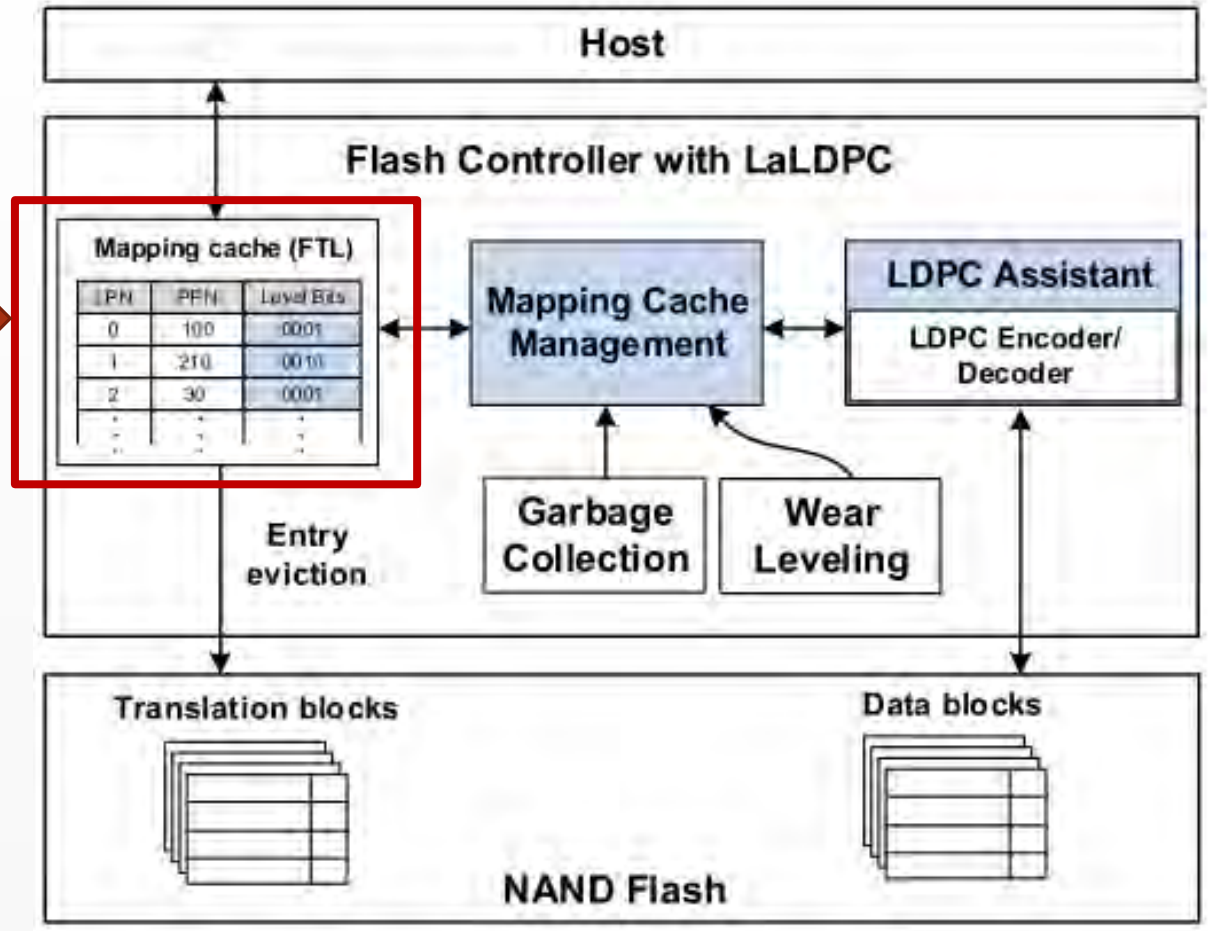
We take DFTL as an example to implement LaLDPC.

Design of LaLDPC: Architecture Overview



Design of LaLDPC: Architecture Overview

One Storage Component



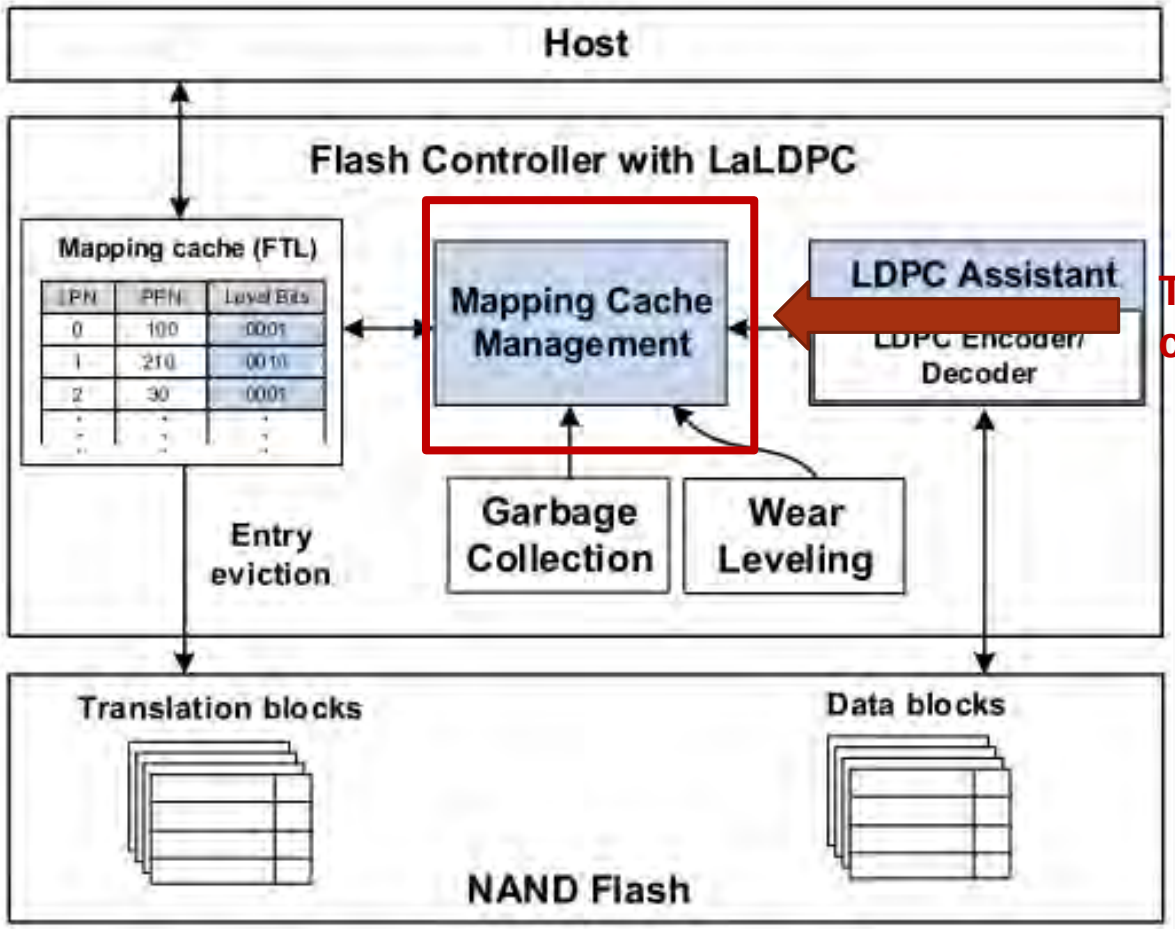
Design of LaLDPC: Storage Component

- The read levels are stored into the flash translation layer in mapping cache;
- Each mapping cache entry stores one read level represented by four bits;
- Read level ranges from 1 to 7.

Mapping cache (FTL)

LPN	PPN	Level Bits
0	100	0001
1	210	0010
2	30	0001
⋮	⋮	⋮

Design of LaLDPC: Architecture Overview



The first functional component

Design of LaLDPC: Mapping Cache Management (1/3)

Mapping cache management in two aspects

1. **Manage read levels of cache entries**
2. **Manage cache entry evictions**

Design of LaLDPC: Mapping Cache Management (2/3)

Mapping cache management in two aspects

1. **Manage read levels of cache entries**
2. **Manage cache entry evictions**

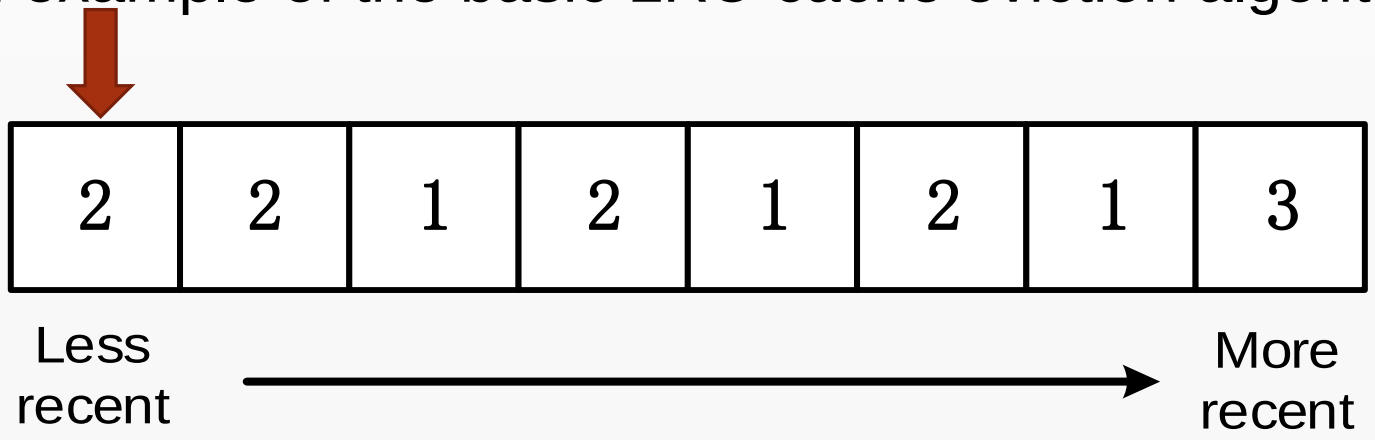
- **Initialize** read levels as 1 in the creation of mapping cache entries;
- **Update** read levels to be the latest read level when read happens;
- **Reset** read levels to 1 when write and garbage collection happens.

Design of LaLDPC: Mapping Cache Management (3/3)

Mapping cache management in two aspects

1. Manage read levels of cache entries
2. **Manage cache entry evictions**

An example of the basic LRU cache eviction algorithm in DFTL:



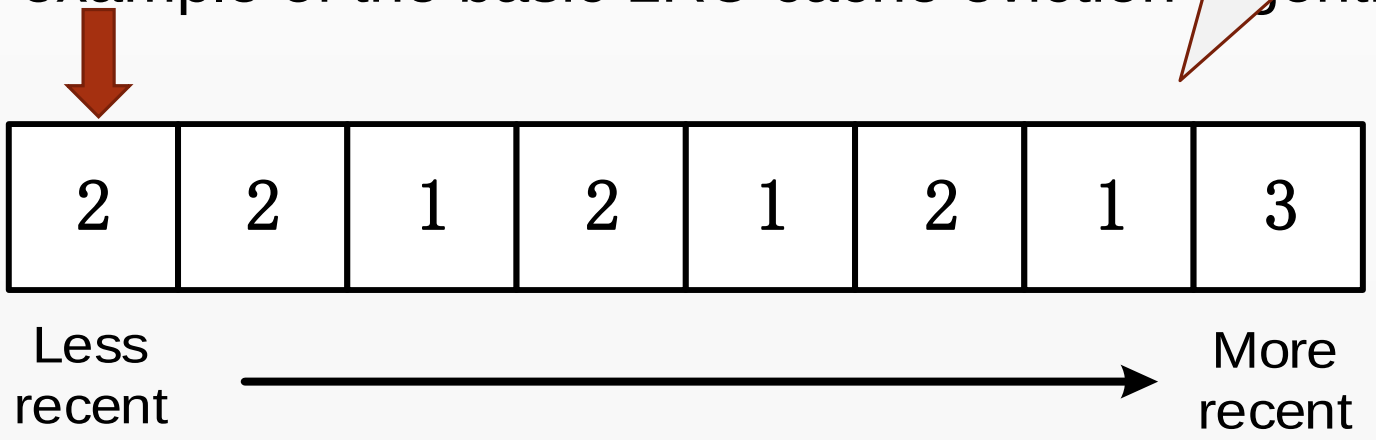
Design of LaLDPC: Mapping Cache Management (3/3)

Mapping cache management in two aspects

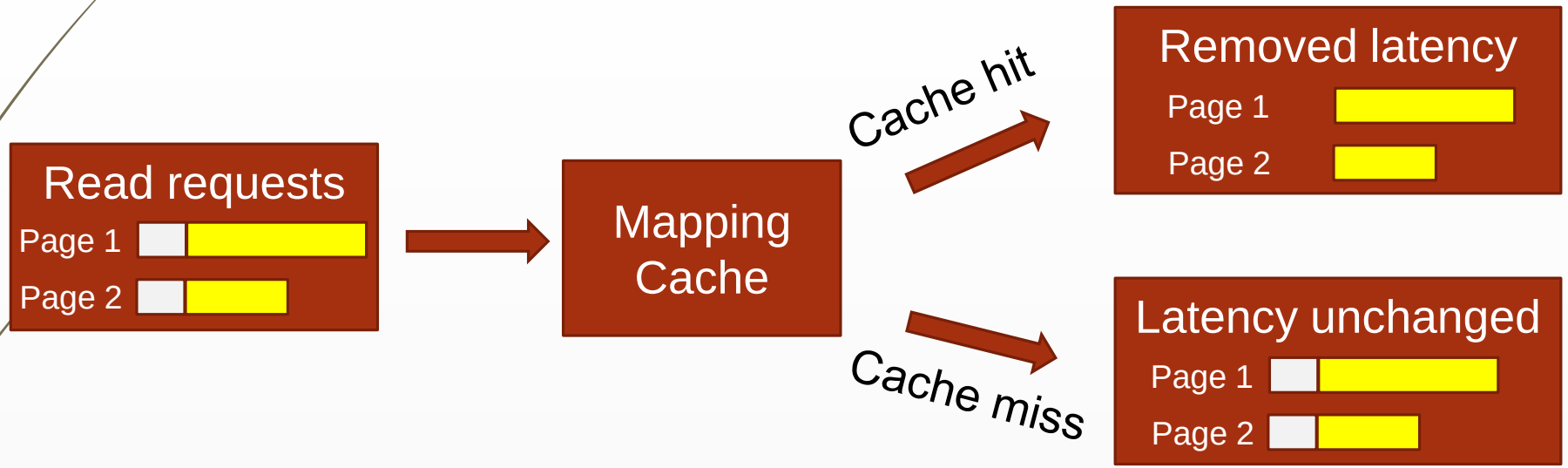
1. Manage read levels of cache ent
2. **Manage cache entry evictions**

LRU can't be aware of LDPC read latency! A new cache eviction algorithm is developed

An example of the basic LRU cache eviction algorithm in DFTL:



Design of LaLDPC: why a new cache eviction algorithm?



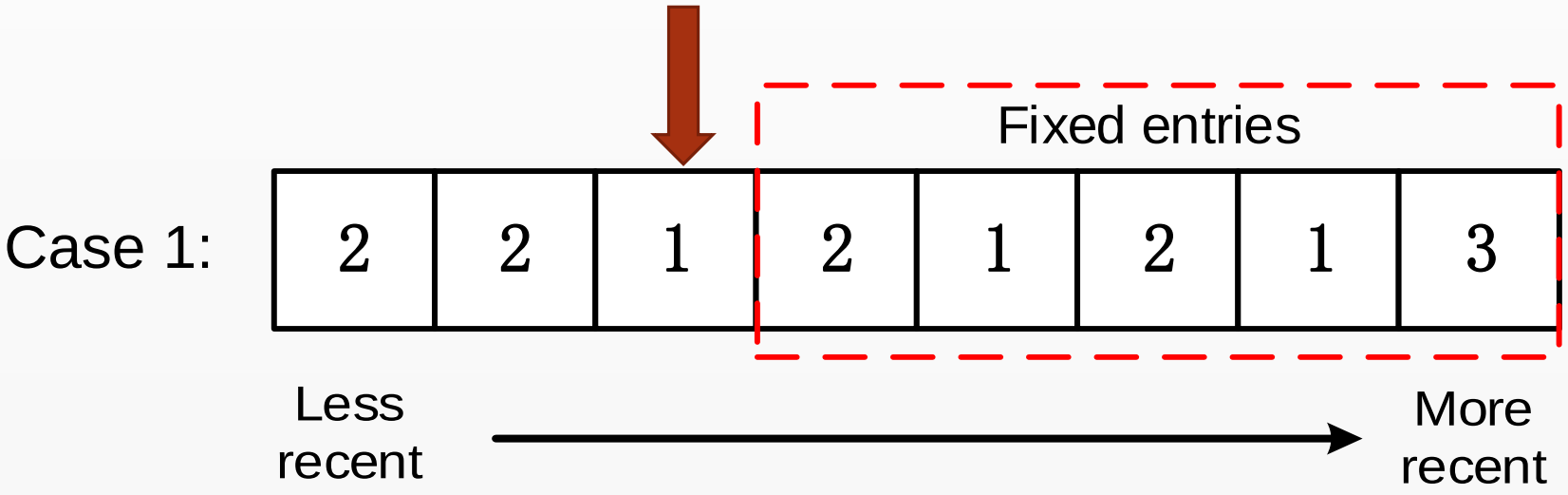
- LaLDPC applied on pages that involve long read latency can achieve more latency benefits;
- Only when cache hits happen, latency reduction can made by LaLDPC.

Improve cache hit ratio of pages with long read latency and keep them in mapping cache as long as possible!

Design of LaLDPC: a new cache eviction algorithm with awareness of read latency (1/2)

The rules to find the cache entry to evict:

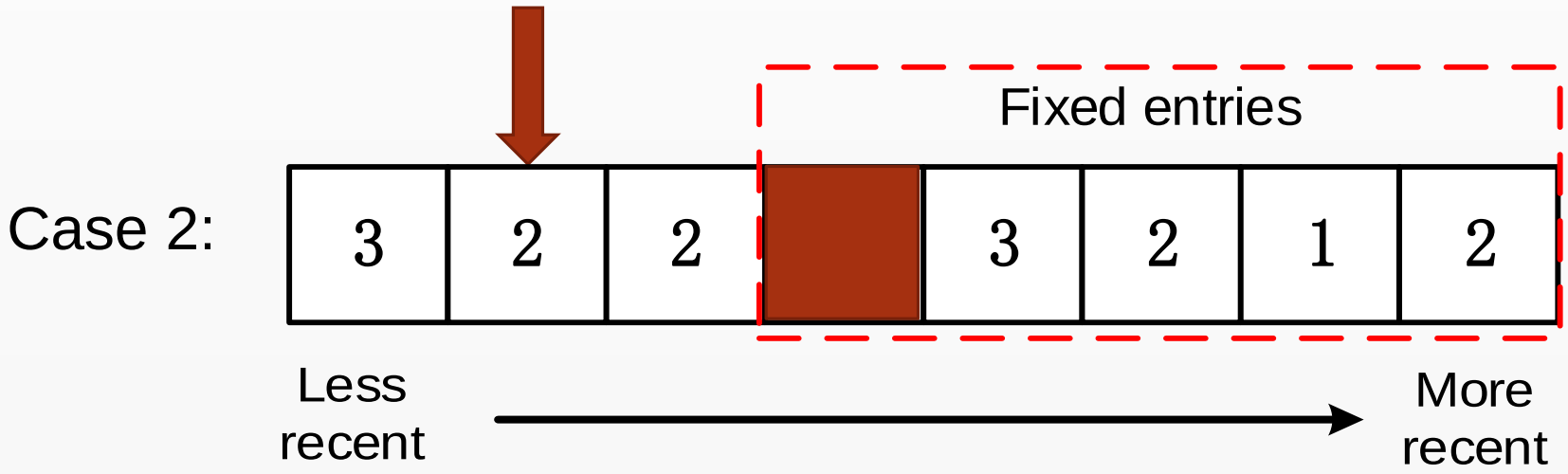
1. With the smallest read level – latency awareness;
2. The least recent entry – LRU property;
3. Not in the fixed entry set – keeping part of access locality.



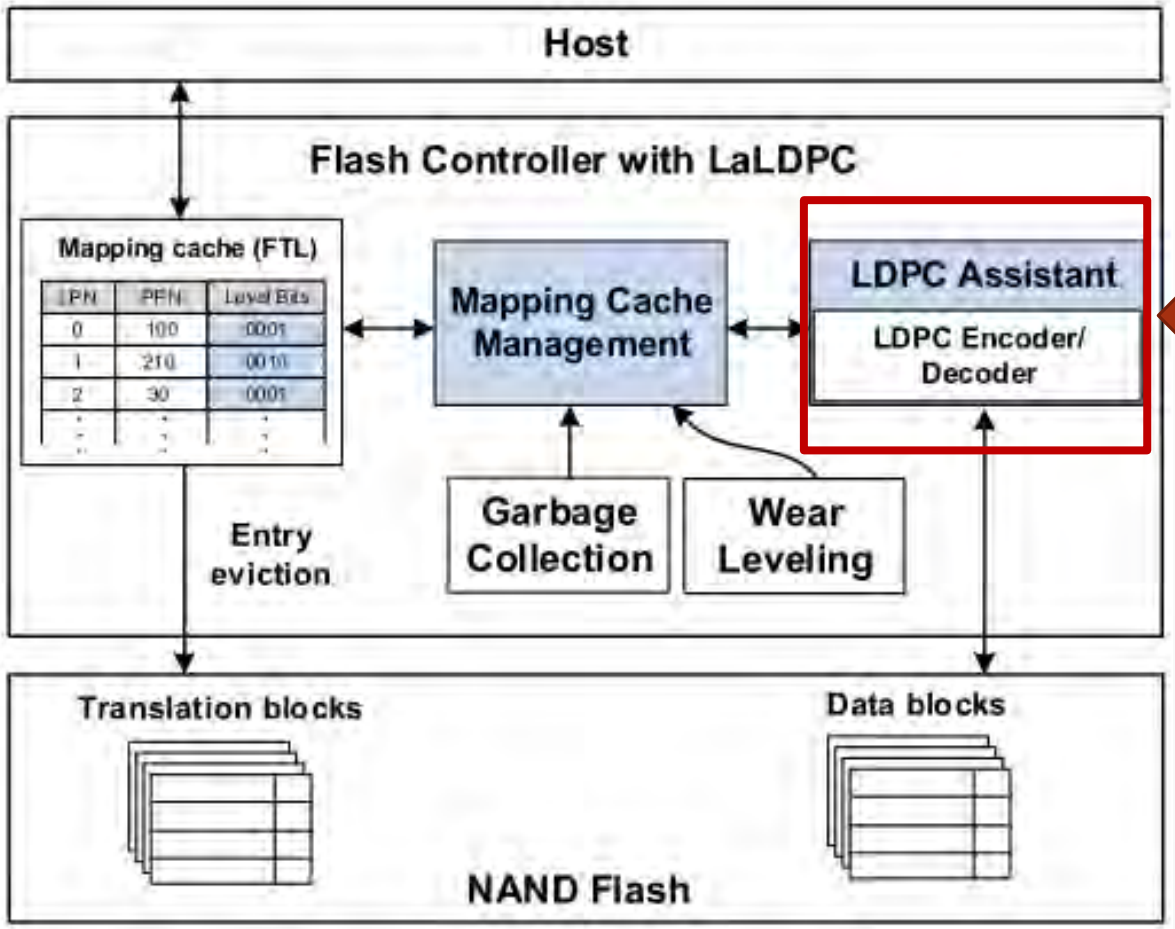
Design of LaLDPC: New Cache Eviction Algorithm with awareness of read latency (2/2)

The rules to find the cache entry to evict:

1. With the smallest read level – latency awareness;
2. The least recent entry – LRU property;
3. Not in the fixed entry set – keeping part of access locality.

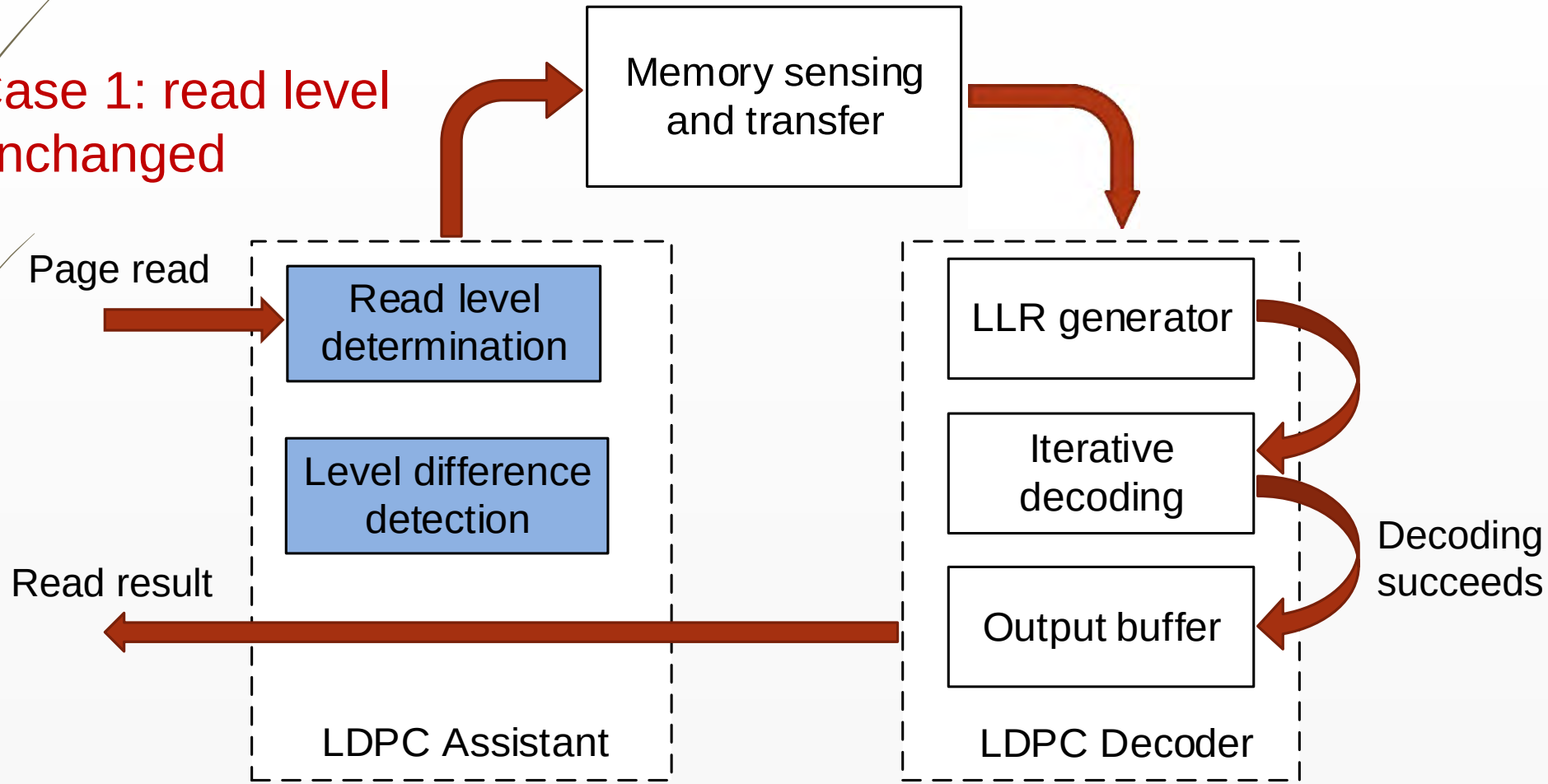


Design of LaLDPC: Architecture Overview

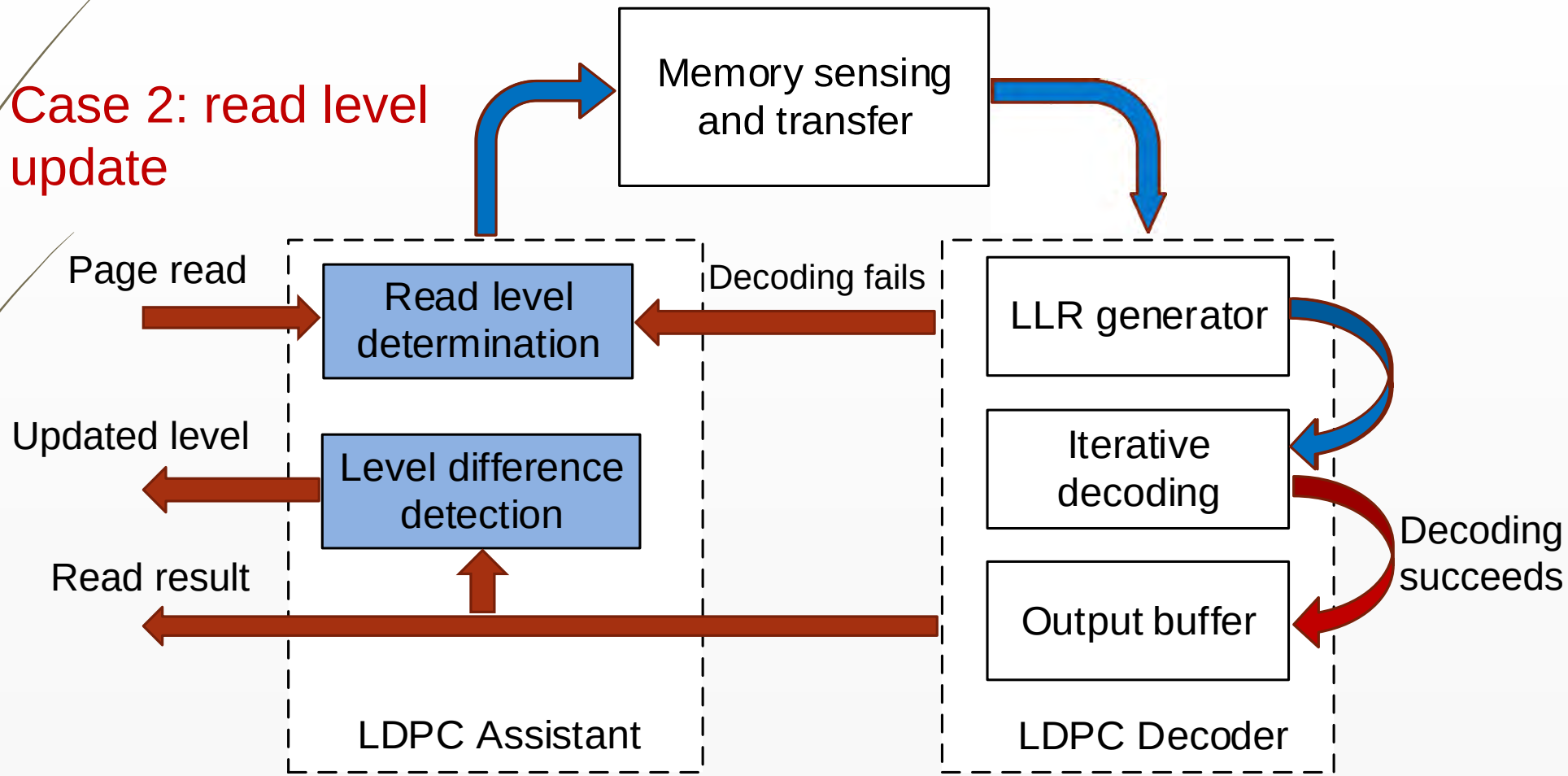


The second functional component

Design of LaLDPC: LDPC Assistant Component (1/2)



Design of LaLDPC: LDPC Assistant Component (2/2)



Design of LaLDPC: Storage Overhead

- The storage overhead in LaLDPC is taken by the read levels in mapping cache entries;
- Assuming the size of one mapping cache entry is 8 Bytes, the portion of space taken by the four level bits is:

$$4 /(8*8) * 100\% = 6.25\%$$

- For mapping cache with the size of 256MB, level bits take 16MB storage space.

Outline

- Background and Motivation
- Design of LaLDPC
- ***Evaluations***
- Summary

Evaluations: Experiment Setup

SSD configuration

- 32GB SSD with 15% over-provision is configured with 8 packages, each of which has 8 planes;
- Each plane contains 1024 blocks and each block has 64 pages with size of 4KB;

Latency parameters for MLC flash

- Page write latency: 900μs ;
- Block erase latency: 3.5ms;
- Read latencies in the table.

read level	RBER	read latency (μs)
1	< 0.005	85
2	$[0.005, 0.006)$	109
3	$[0.006, 0.008)$	133
4	$[0.008, 0.009)$	157
5	$[0.009, 0.01)$	181
6	$[0.01, 0.012)$	205
7	$[0.012, 0.013]$	229

Evaluations: Methods and Parameter Settings for Comprehensive Experiments

- ***LDPC-in-SSD:*** the current progressive LDPC method;
- ***Ideal:*** LDPC method with known read levels;
- ***LaLDPC*** $_{LRU}$: LaLDPC method with LRU cache eviction algorithm;
- ***LaLDPC*** $_{new}$: LaLDPC method with the new cache eviction algorithm.

Three parameters are comprehensively configured for the basic experiment and sensitivity studies:

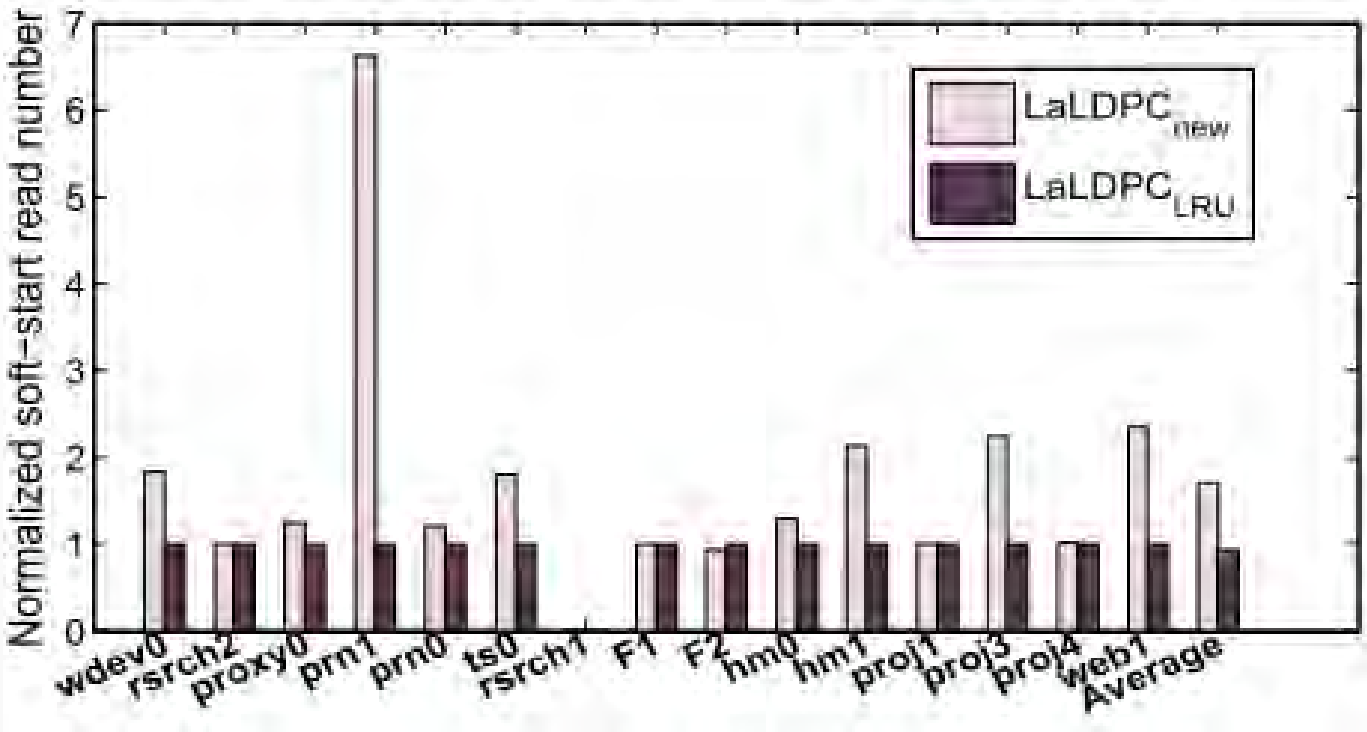
- Cache size;
- Fixed entry length of mapping cache;
- Flash life stages.

Evaluation Results: Important Workload Statistics (1/3)

workload	soft read ratio	workload	soft read ratio
wdev0	36%	F2	100%
rsrch2	0.2%	hm0	41%
proxy0	17%	hm1	21%
prn1	4%	proj1	0.5%
prn0	22%	proj3	58%
ts0	36%	proj4	5%
rsrch1	91%	web1	46%
F1	99%	-	-

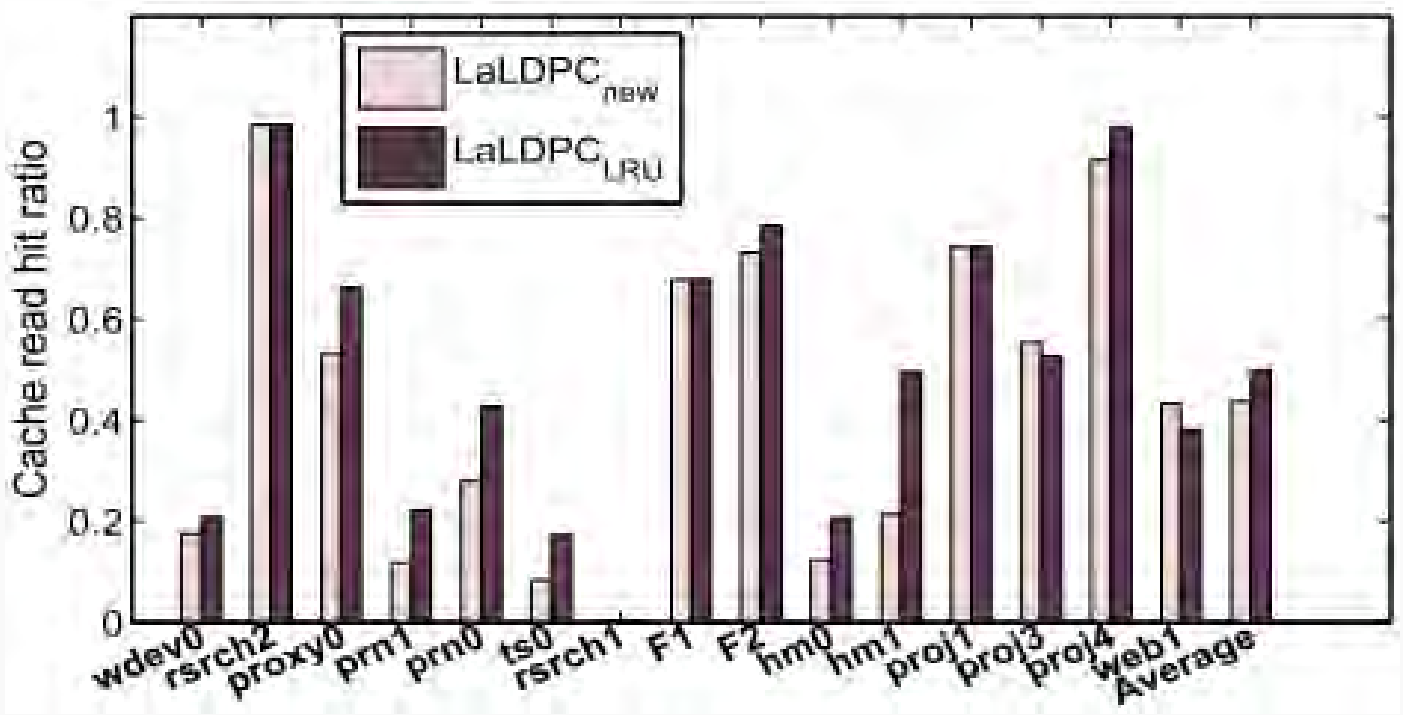
Soft read ratio reflects the potential performance improvement of workloads because only latency of soft reads can be further reduced.

Evaluation Results: Important Workload Statistics (2/3)



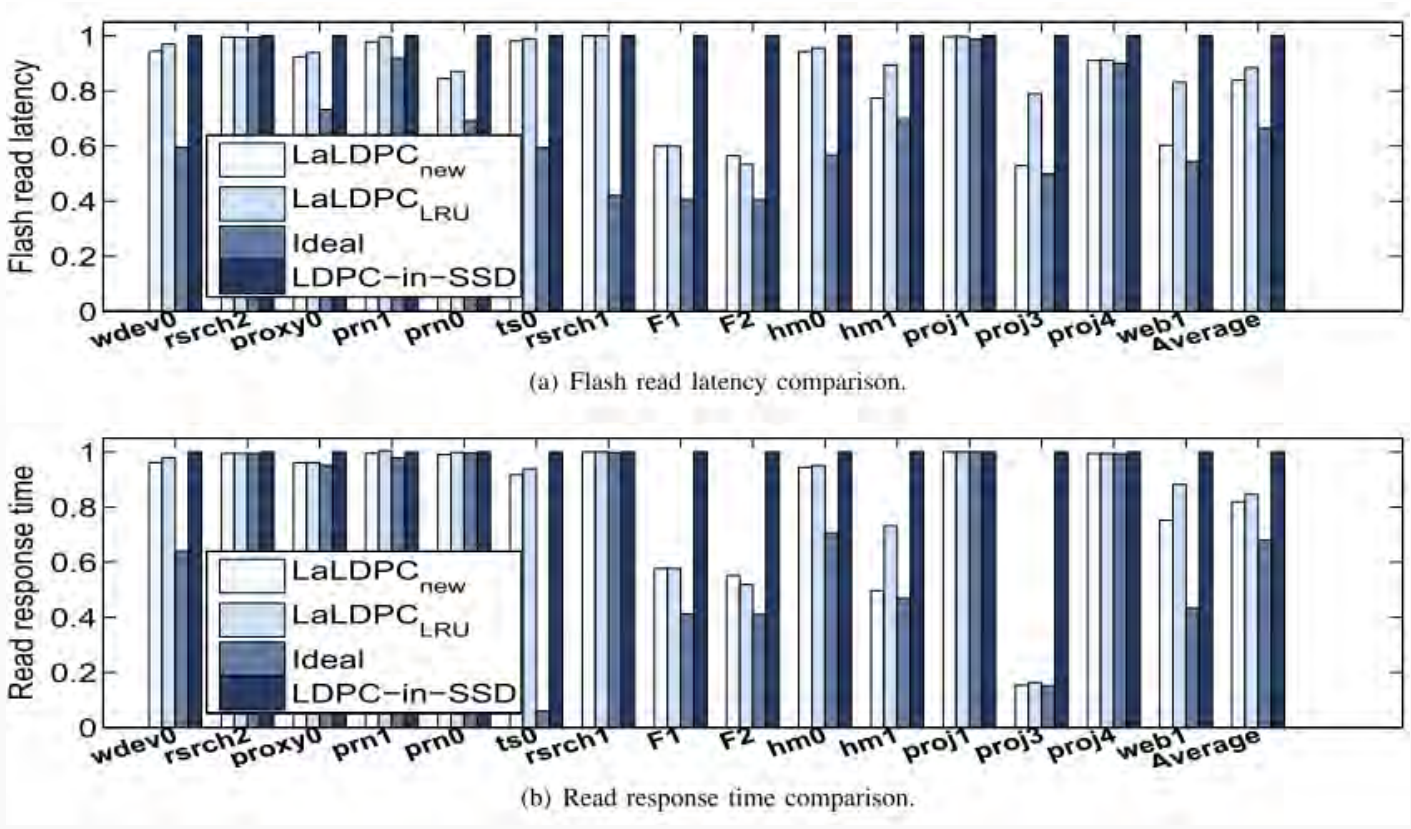
The ratio of soft-start reads reflects that how many reads in the workloads can be optimized from the two LaLDPC methods.

Evaluation Results: Important Workload Statistics (3/3)



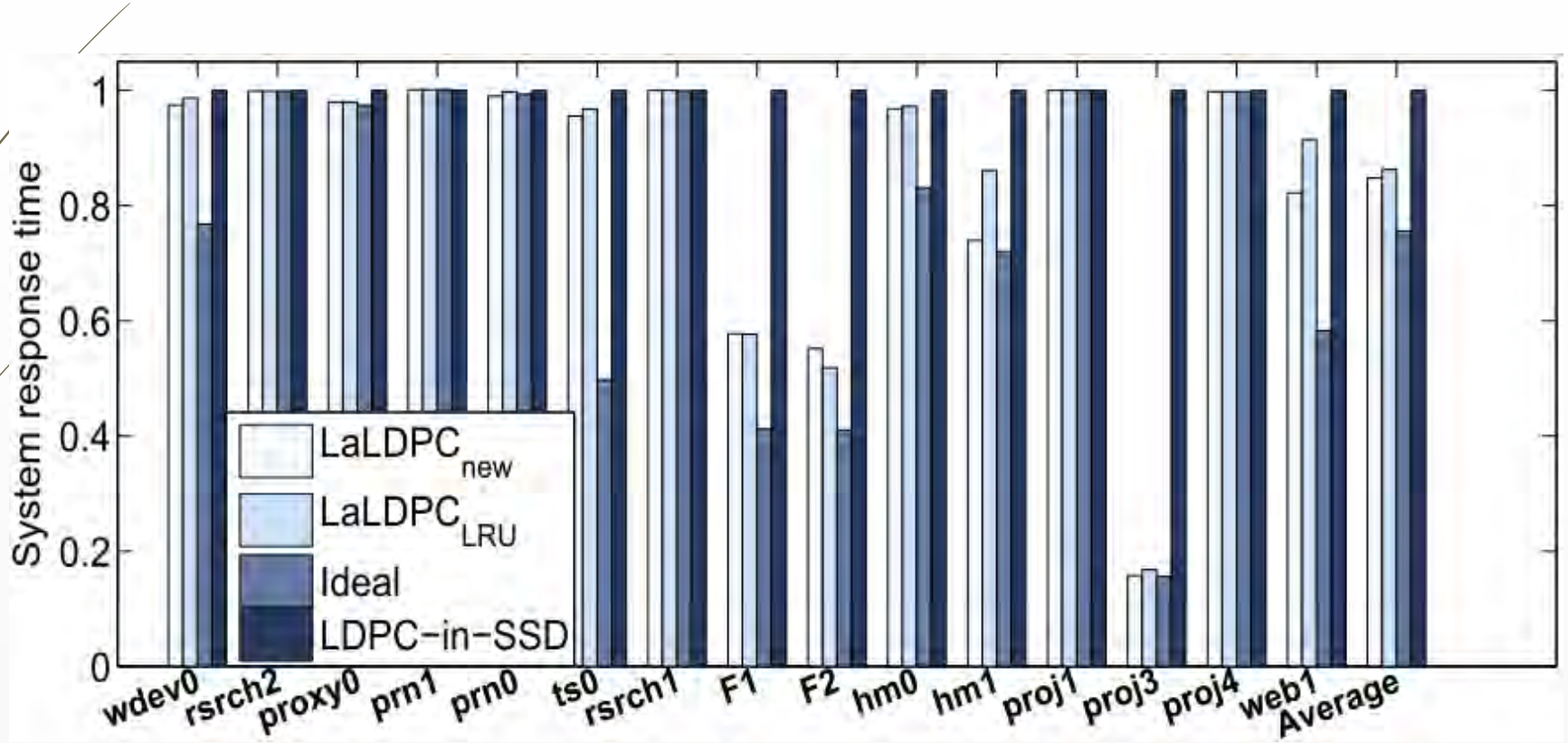
$LaLDPC_{new}$ shows decreased cache hit ratios than $LaLDPC_{LRU}$ because of losing part of aceess locality.

Evaluations: Read Performance



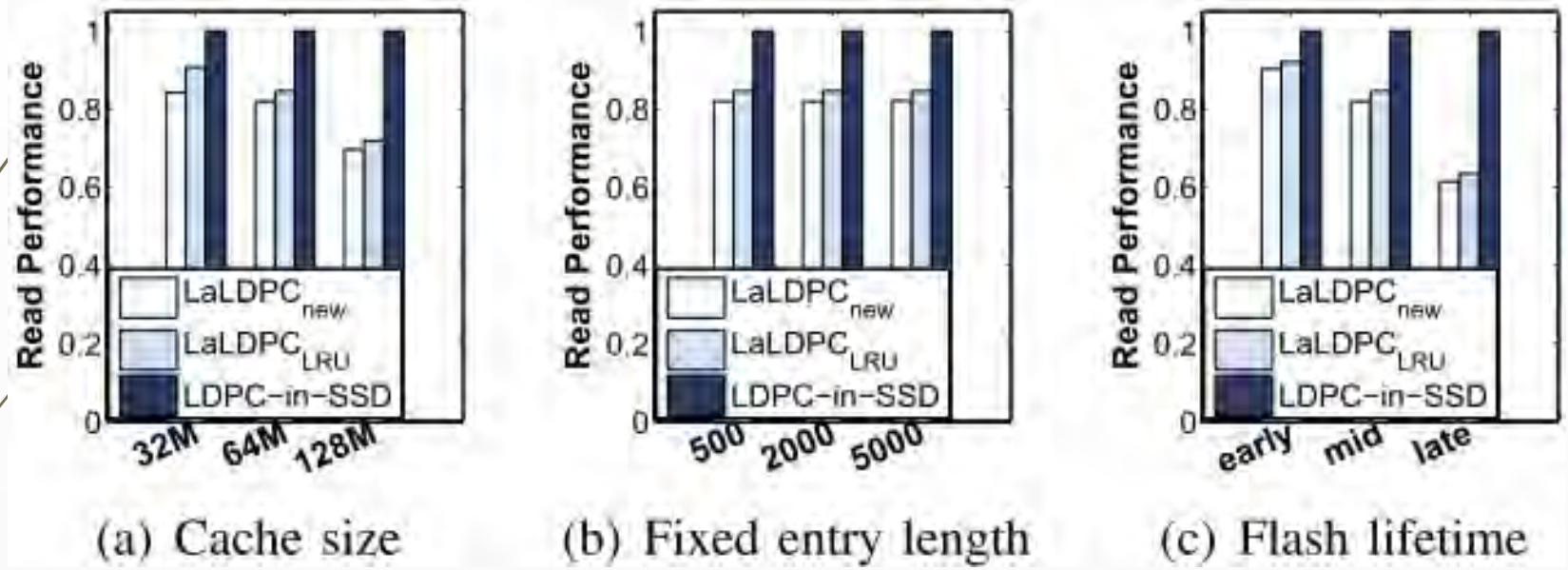
LaLDPC$_{new}$ can remove 56% of redundant read performance by comparing with the ideal results and can improve read performance of LDPC-in-SSD by 18%.

Evaluations: System Response Time



About 24% of system response time in LDPC-in-SSD can be reduced.

Evaluations: Sensitivity Study Results



(a) Cache size (b) Fixed entry length (c) Flash lifetime

a. When a larger mapping cache is used, more performance benefits of $LaLDPC_{new}$ and $LaLDPC_{LRU}$ can be achieved;

b. When more entries are fixed in mapping cache, latency awareness is reduced and advantages of the new cache eviction algorithm $LaLDPC_{new}$ are decreased;

c. For SSDs in the late life stage, higher performance improvements can be achieved.

Outline

- Background and Motivation
- Design of LaLDPC
- Evaluations
- ***Summary***

Summary

- We study read performance slowdown caused by latency accumulation of LDPC codes and discover the temporal read level locality;
- We propose a latency-aware LDPC method. The awareness has been reflected in two aspects as follows:
  1) LaLDPC can be aware of the latency of a LDPC read by leveraging read levels of previous reads;
  2) The new cache eviction algorithm in LaLDPC can be aware of read latencies of cached pages and brings more benefits on read performance.
- We evaluate the effectiveness of LaLDPC with extensive experiments.

Thanks for your attention!
Any Questions?

dyjcityu2013@gmail.com