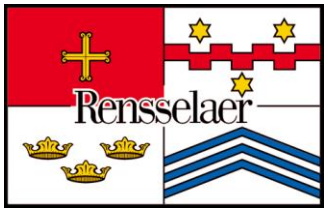
Reducing Solid-State Storage Device Write Stress Through Opportunistic In-Place Delta Compression

Xuebin Zhang, Jiangpeng Li, Hao Wang,
Kai Zhao and Tong Zhang
xuebinzhang.rpi@gmail.com
ECSE Department, Rensselaer Polytechnic Institute (RPI)

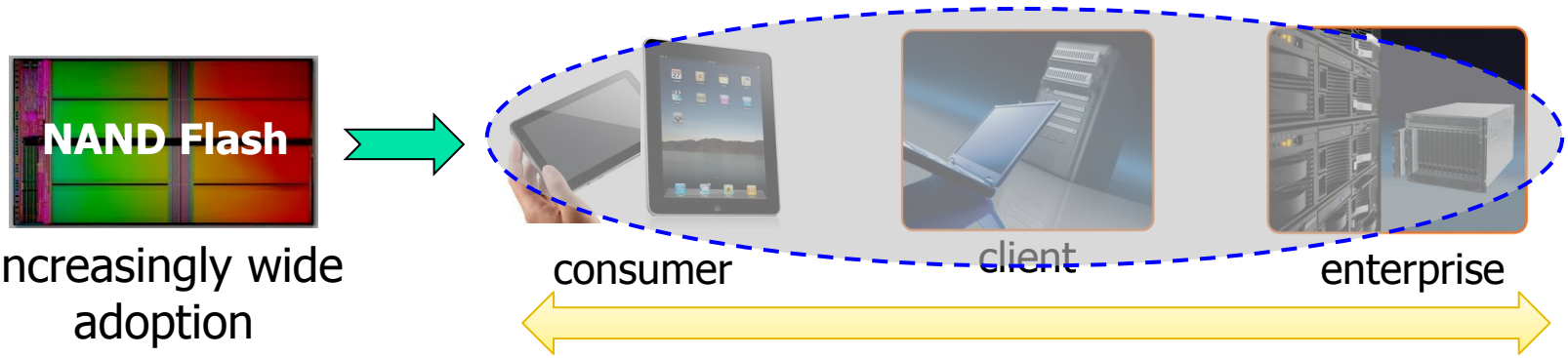


Outline

- Introduction and Motivation
- Progressive Programmability of SLC Flash Pages
- In-place Delta Compression Realization in SSDs
- Evaluation Results and Overhead Analysis
- Future Work and Summary

NAND Flash Memory

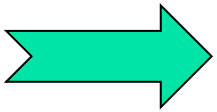
- Increasing Adoptions and Decreasing Cost of NAND Flash Memory



- Endurance: One of Existing Problems of NAND Flash Memory
  - Limited lifetime : SLC: 8000; MLC: 2000; TLC: <1000.
  - Make it worse: Bad management in software stack.

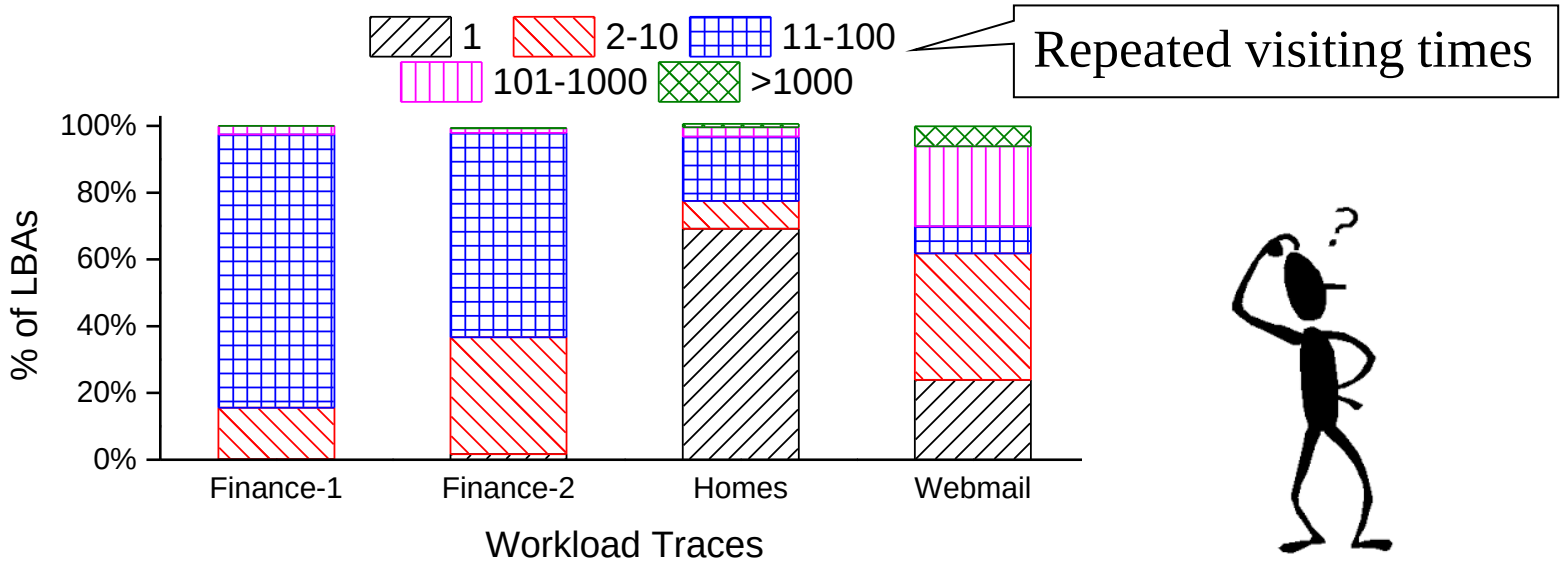
Content Temporal Locality in Storage Systems

- Small (<4kB) update will trigger a 4kB write in Flash (write amplification)
  - 60% writes are smaller than one page (4kB), even less than 10 bytes

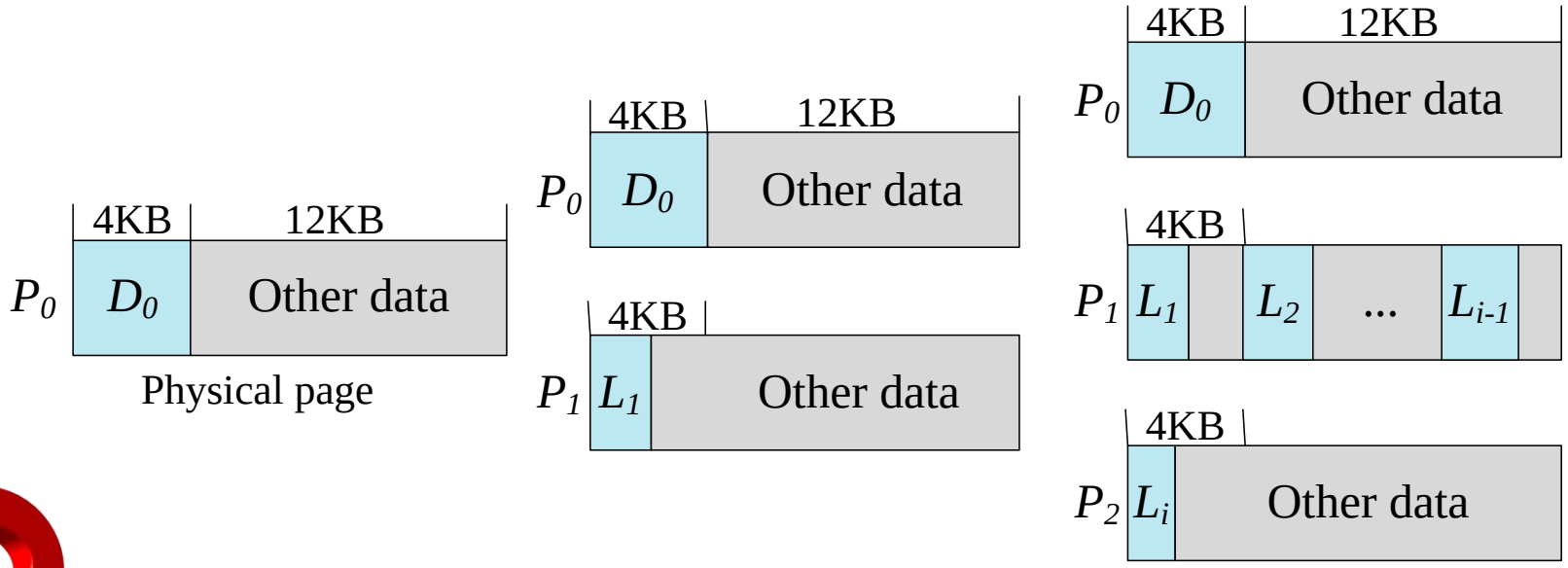


Redundancy between consecutive updates

- Frequently repeated in-place update within a short time period



Delta Compression: Problems in Current Practice



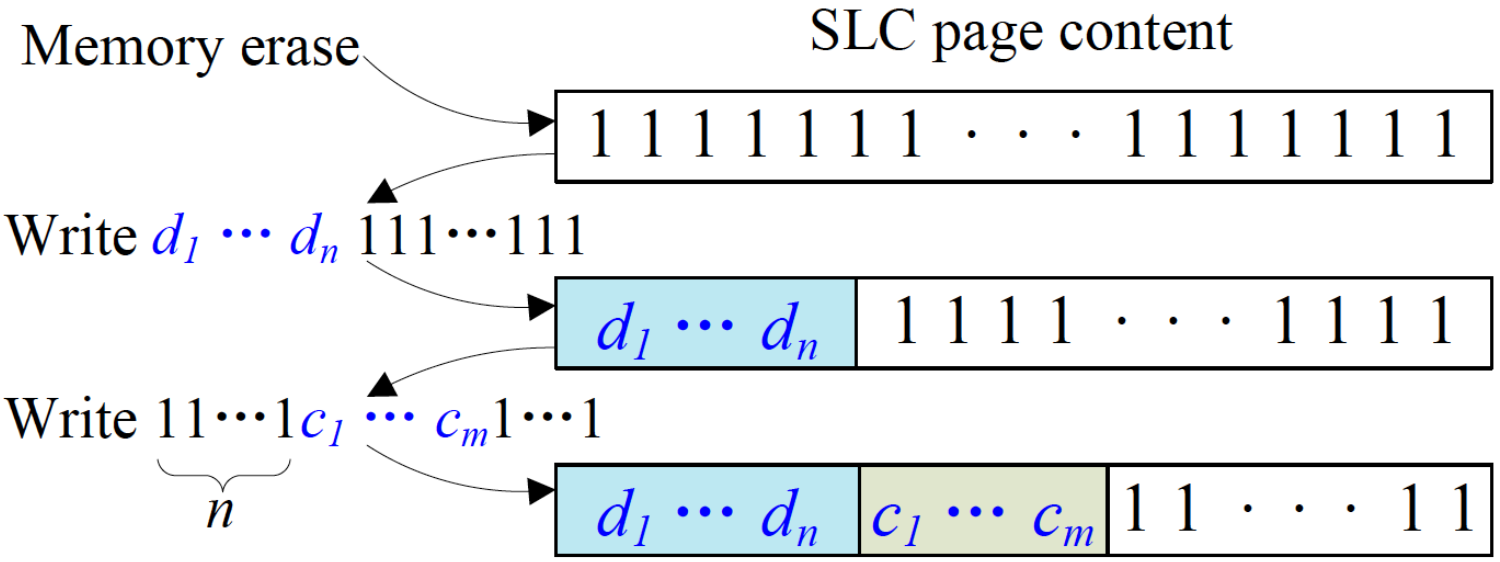
- Latency: need to read multiple pages to recover the latest version.
- Storage: need to keep a mapping table to store the delta's location.

Outline

- Introduction and Motivation
- Progressive Programmability of SLC Flash Pages
- In-place Delta Compression Realization in SSDs
- Evaluation Results and Overhead Analysis
- Summary and Future Work

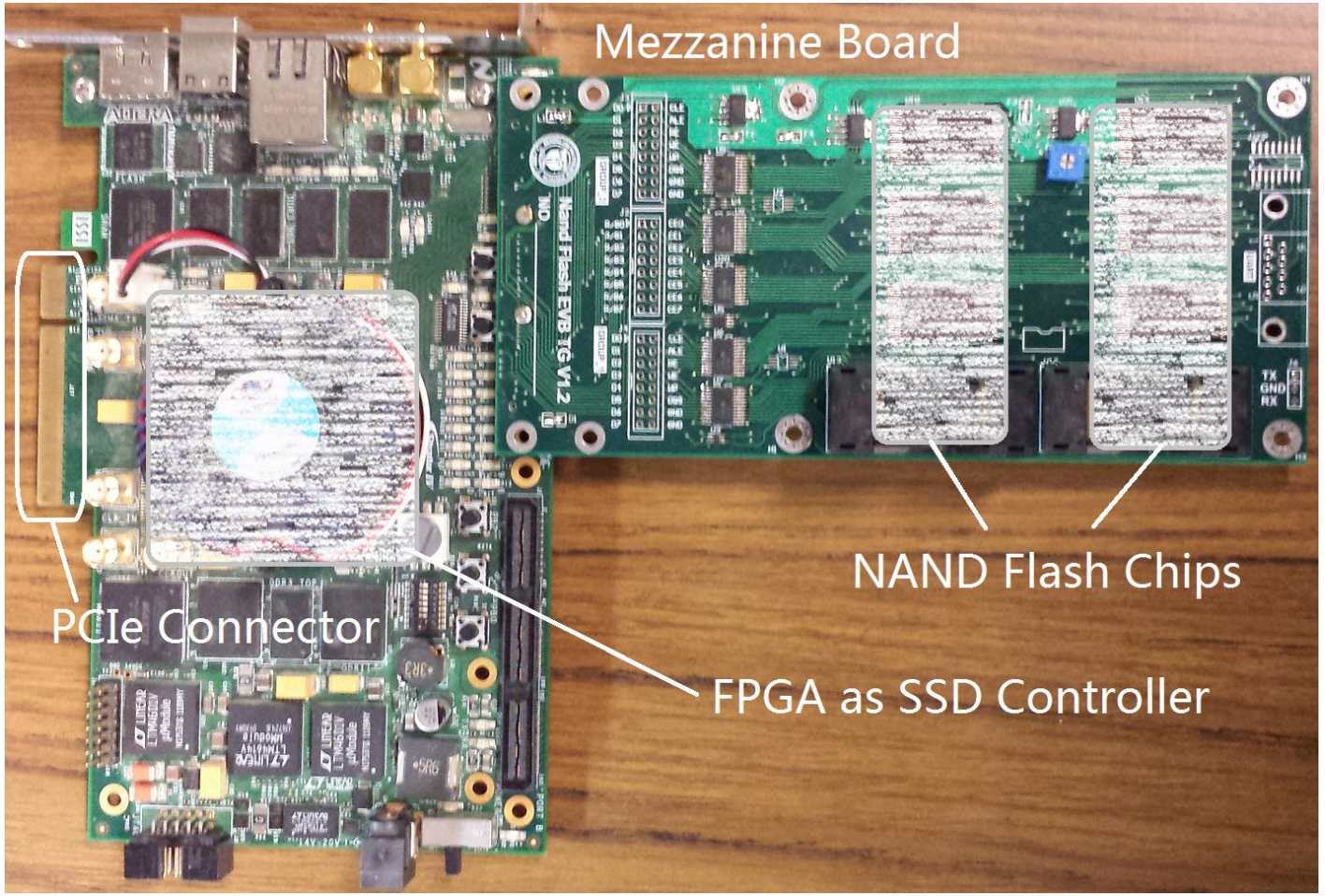
NAND Flash Progressive Programming

- Append new data in the same page (for SLC flash pages)
  - Writing '1' to flash cell will not change its state
  - Overwrite new data to original physical page



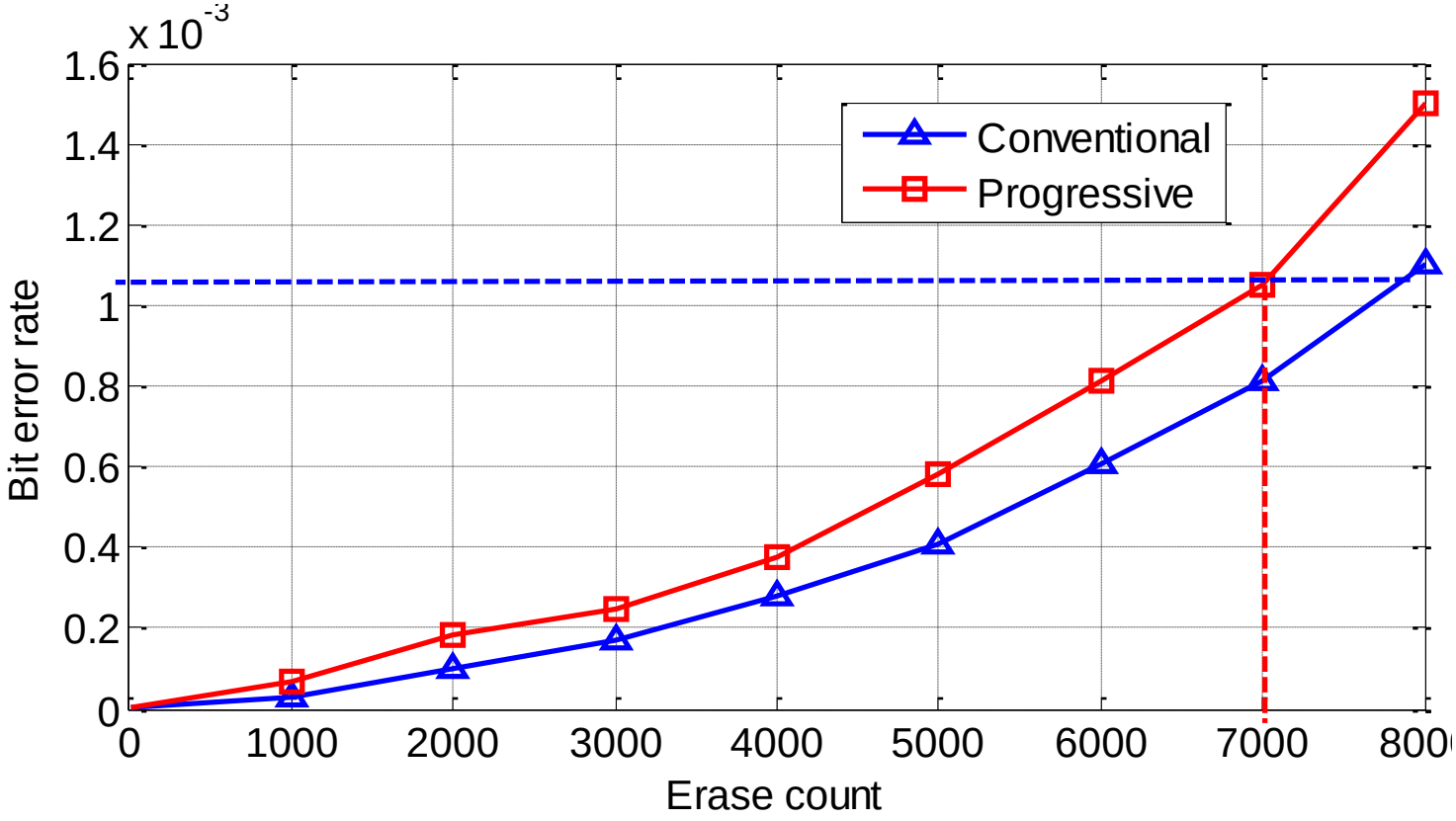
NAND Flash Progressive Programming

- Hardware Platform: PCIe interface, FPGA as the Flash controller



NAND Flash Progressive Programming

- Validity checking of multiple programming before erasing
  - Conventional: one program – one erase
  - Progressive: eight program – one erase

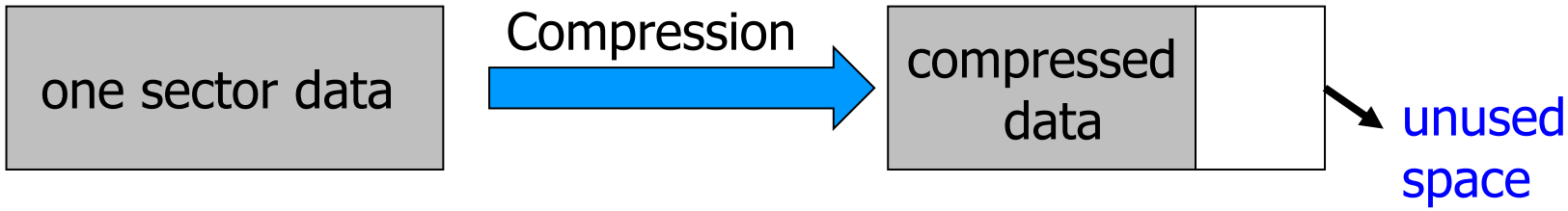


Outline

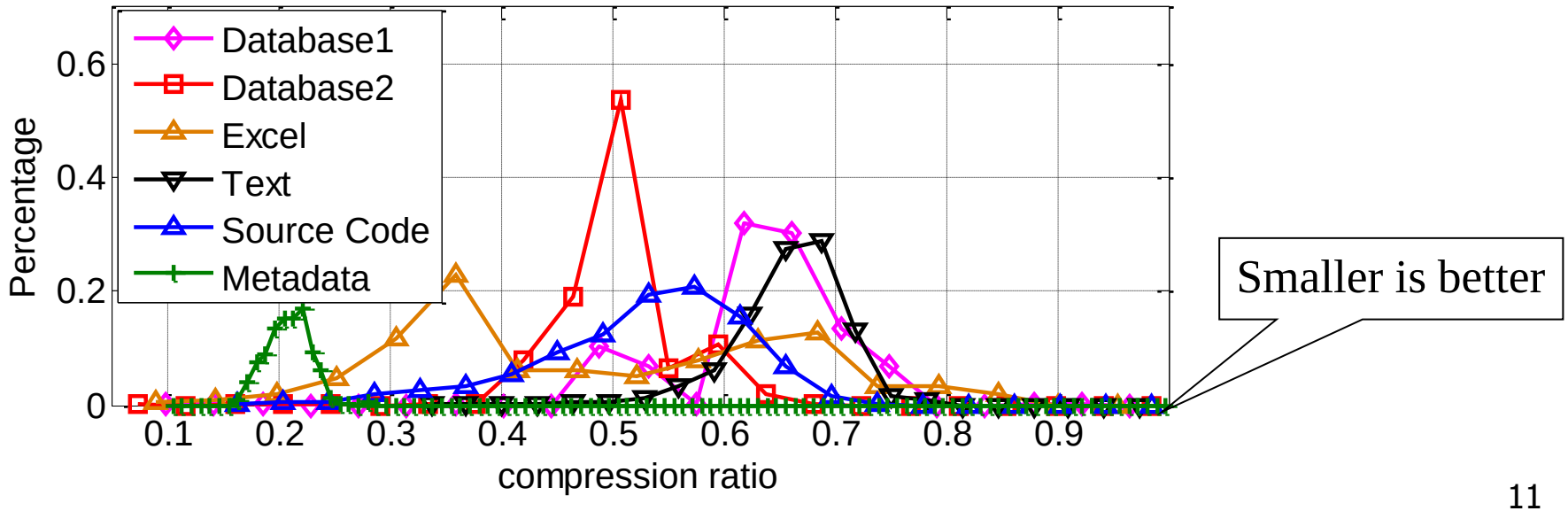
- Introduction and Motivation
- Progressive Programmability of SLC Flash Pages
- In-place Delta Compression Realization in SSDs
- Evaluation Results and Overhead Analysis
- Summary and Future Work

Proposed Solution: Combine Data/Delta Compression

- Use intra-sector lossless data compression to make space for deltas

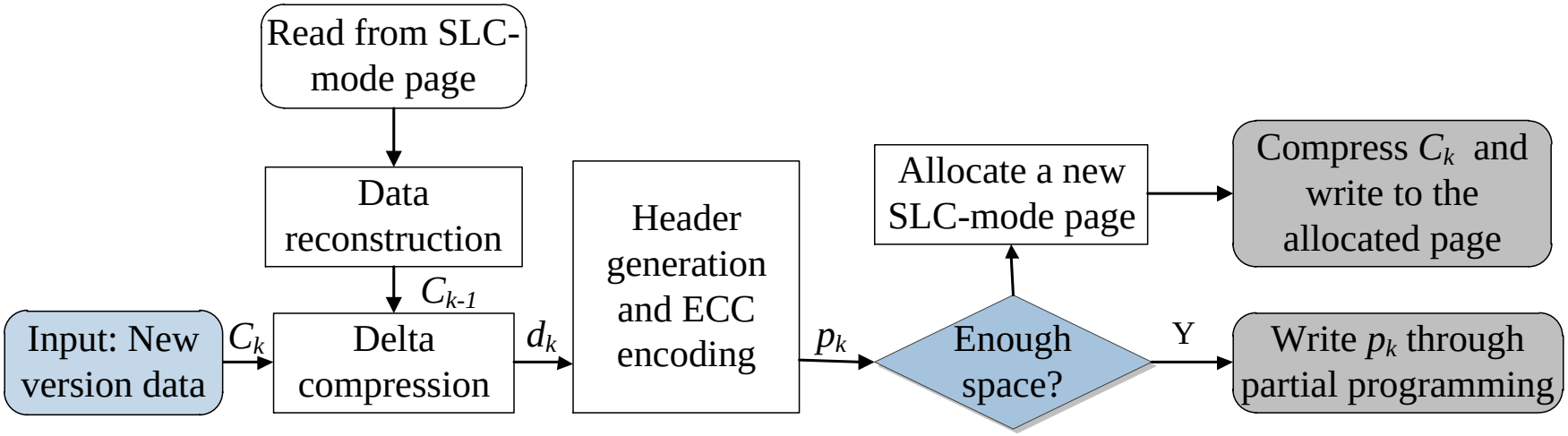


- Per-sector Compression ratio distribution of different file types

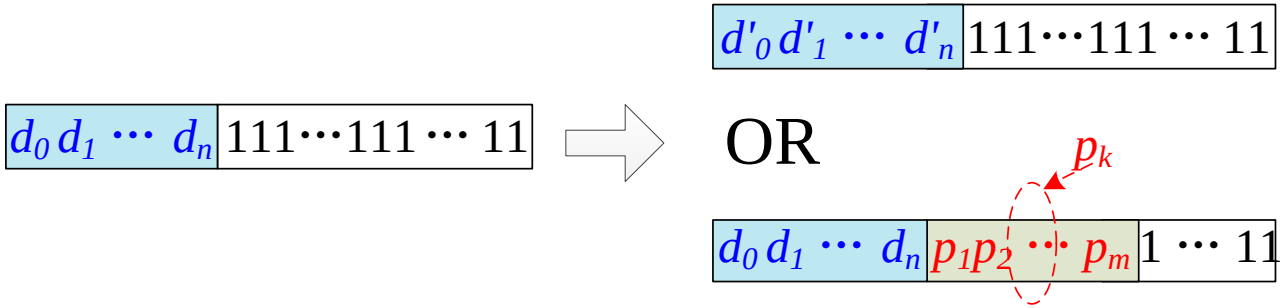


Proposed Solution: In-place Delta Compression

- Overall flow diagram to update one new version of sector data

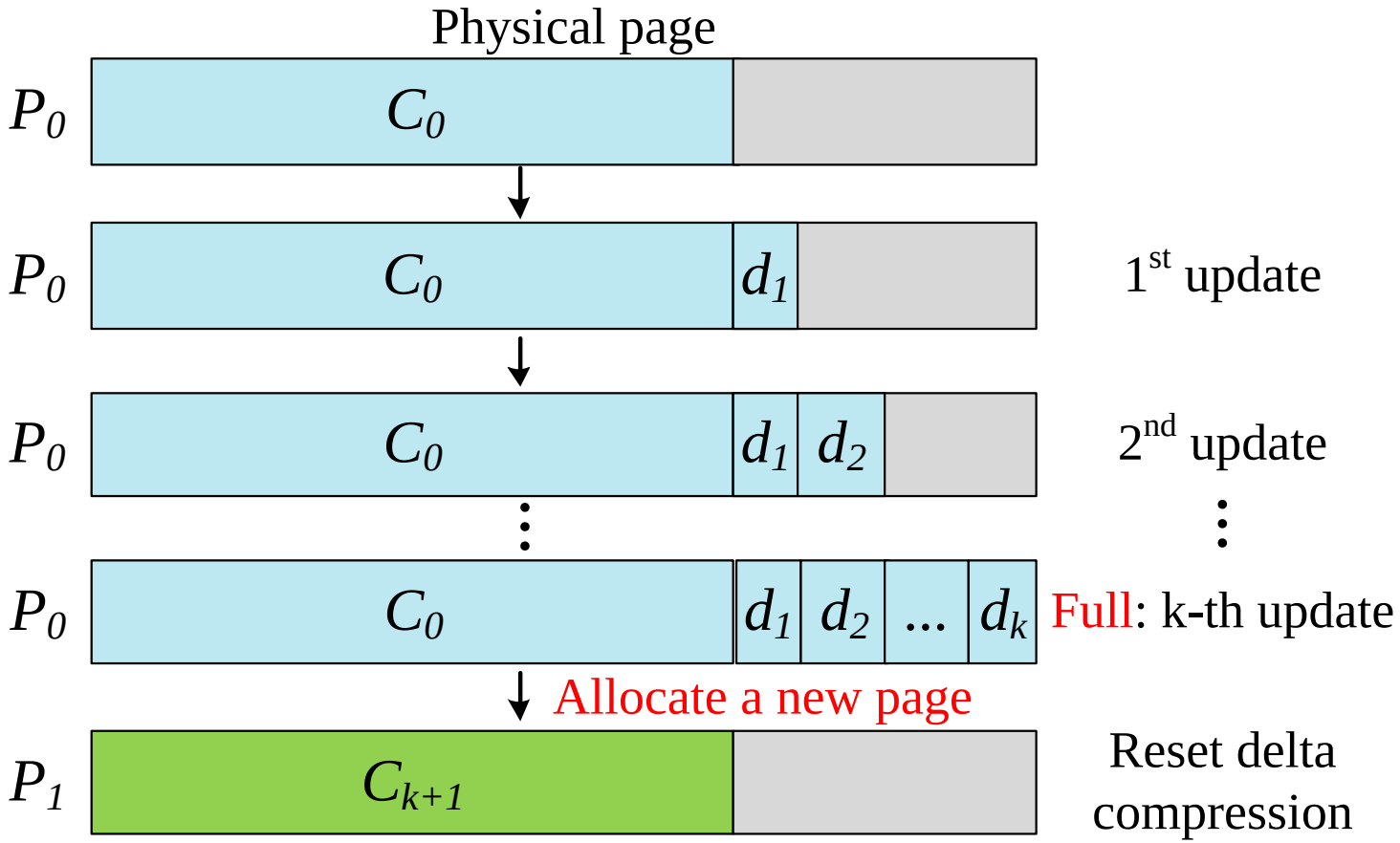


- Sector content change



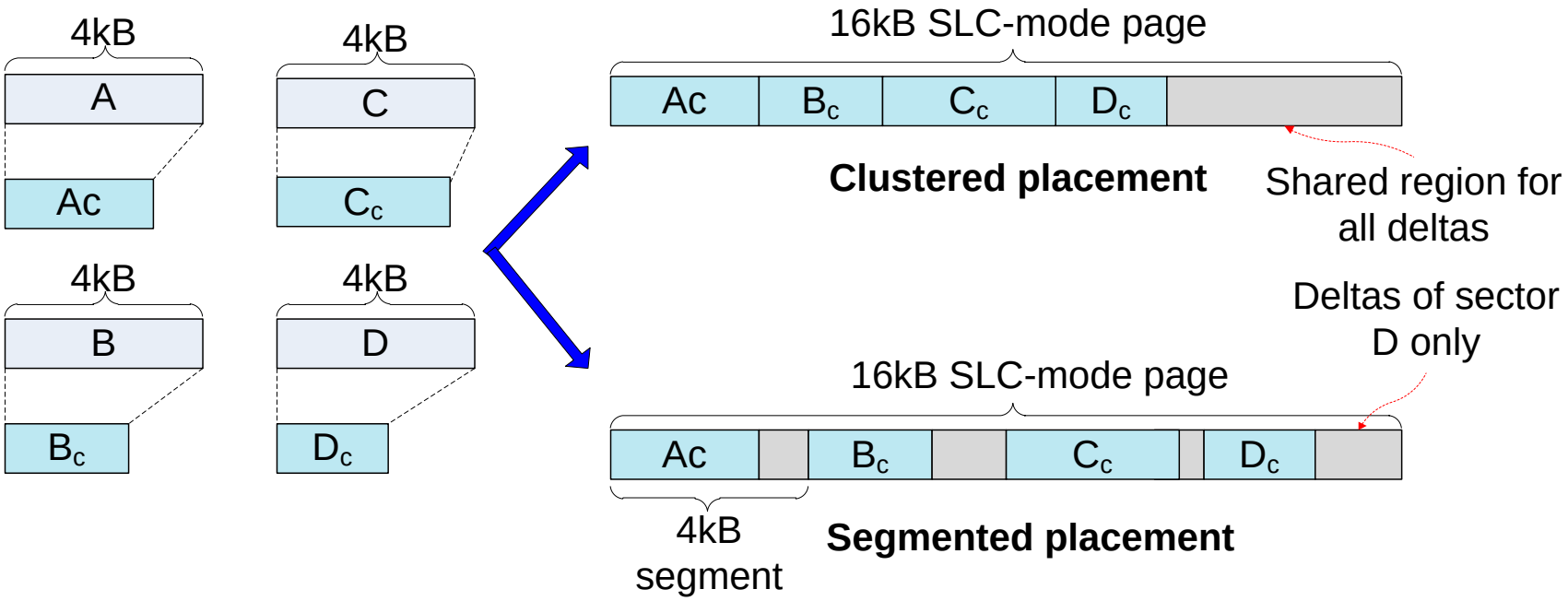
Proposed Solution: In-place Delta Compression

- Consecutive in-place updates



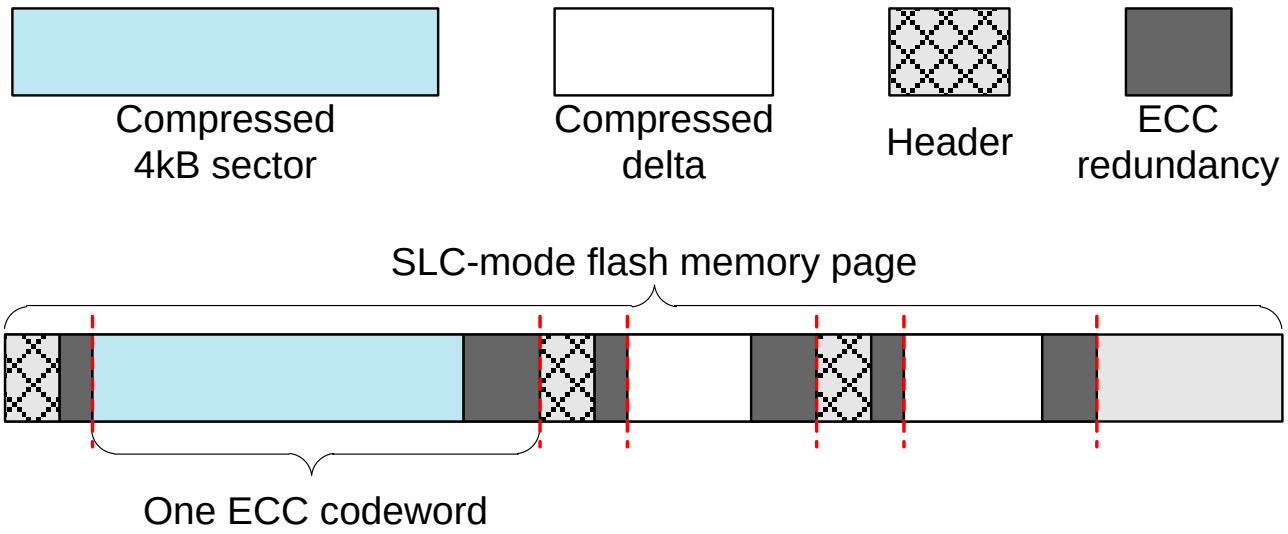
Proposed Solution: Data Placement

- Two different data placement strategies
  - Clustered: shared region for deltas of this physical page
  - Segmented: independent regions for deltas of each sector only



Proposed Solution: ECC Management

- ECC management for different types of data elements
  - Compressed original sector data
  - Compressed deltas
  - Header for each elements

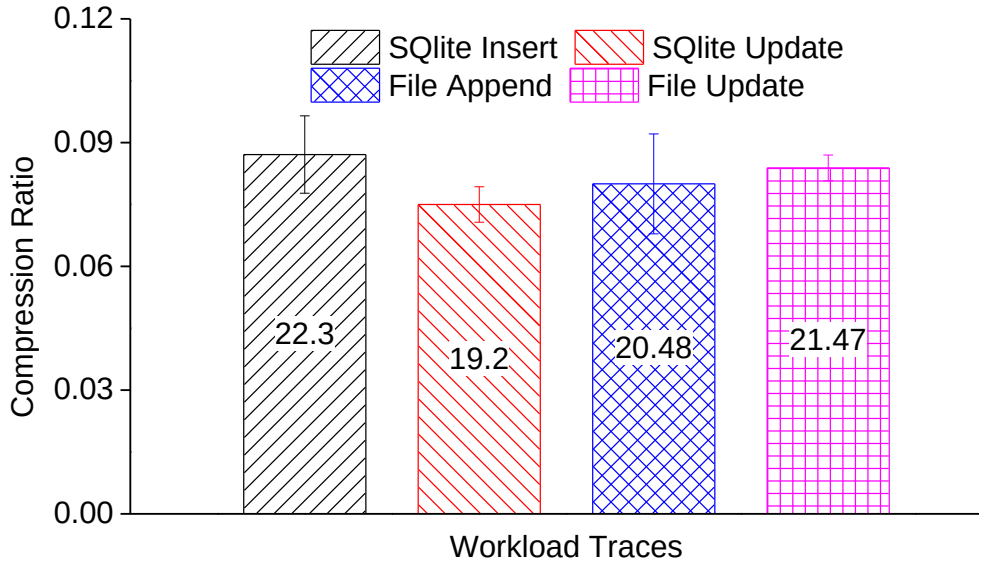


Outline

- Introduction and Motivation
- Progressive Programmability of SLC Flash Pages
- In-place Delta Compression Realization in SSDs
- Evaluation Results and Overhead Analysis
- Summary and Future Work

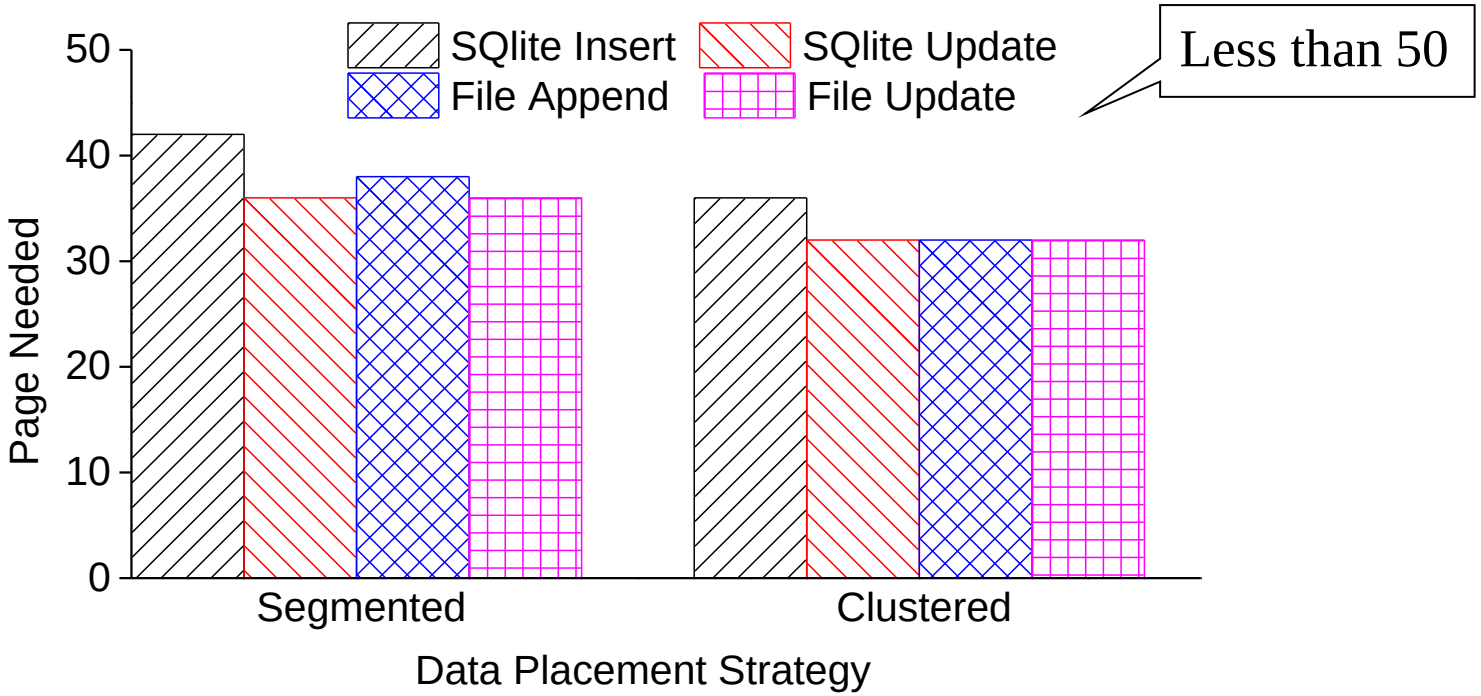
Evaluation: Write Stress Reduction

- Case study: file system metadata
  - Use a benchmark to generate database/files operations
  - Implement a metaAnalyzer to grasp metadata from file system
  - Analyze the collected consecutive versions of metadata
- Delta compression efficiency: (full metadata size is 256 bytes)



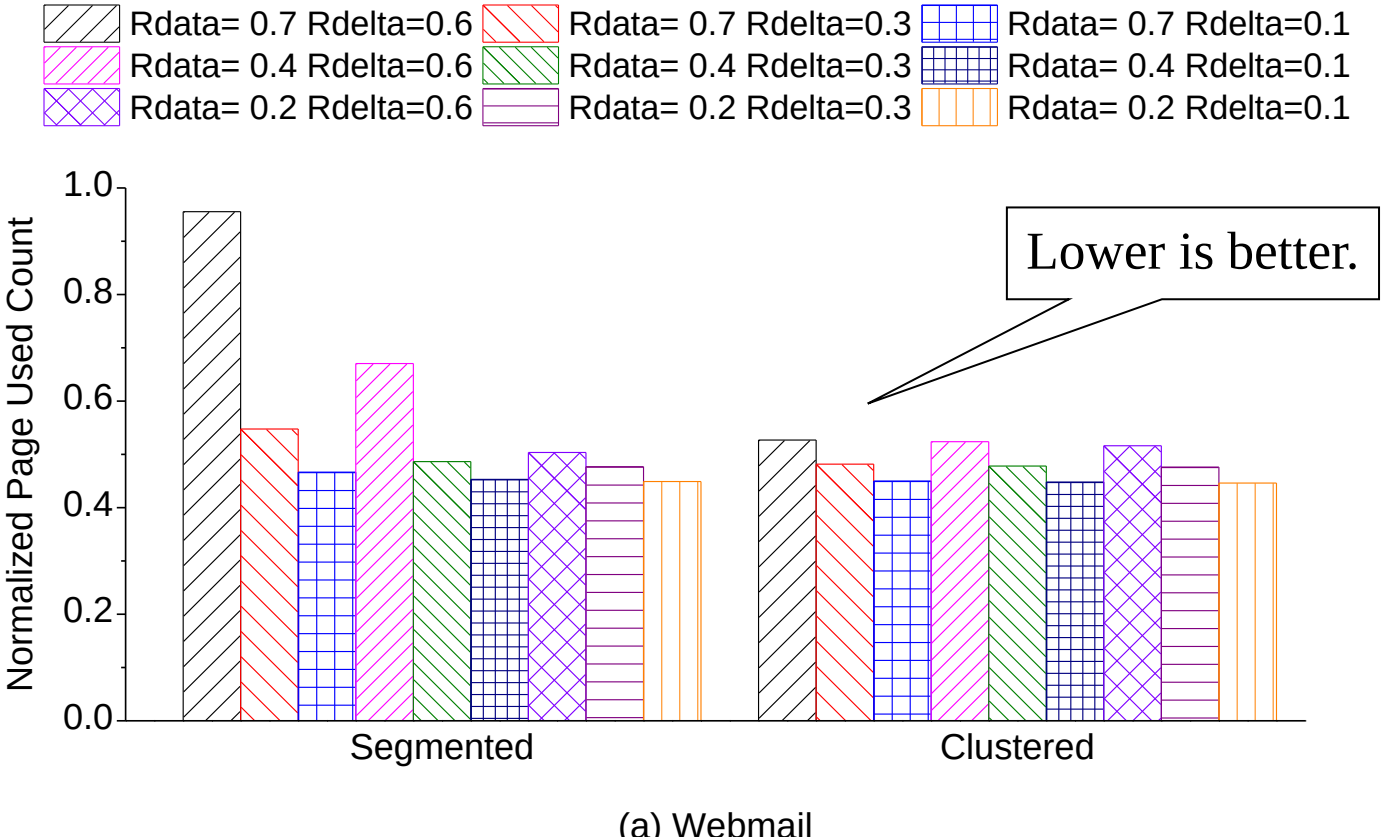
Evaluation: Write Stress Reduction

- Write stress reduction to store 1000 consecutive versions



Evaluation: Write Stress Reduction

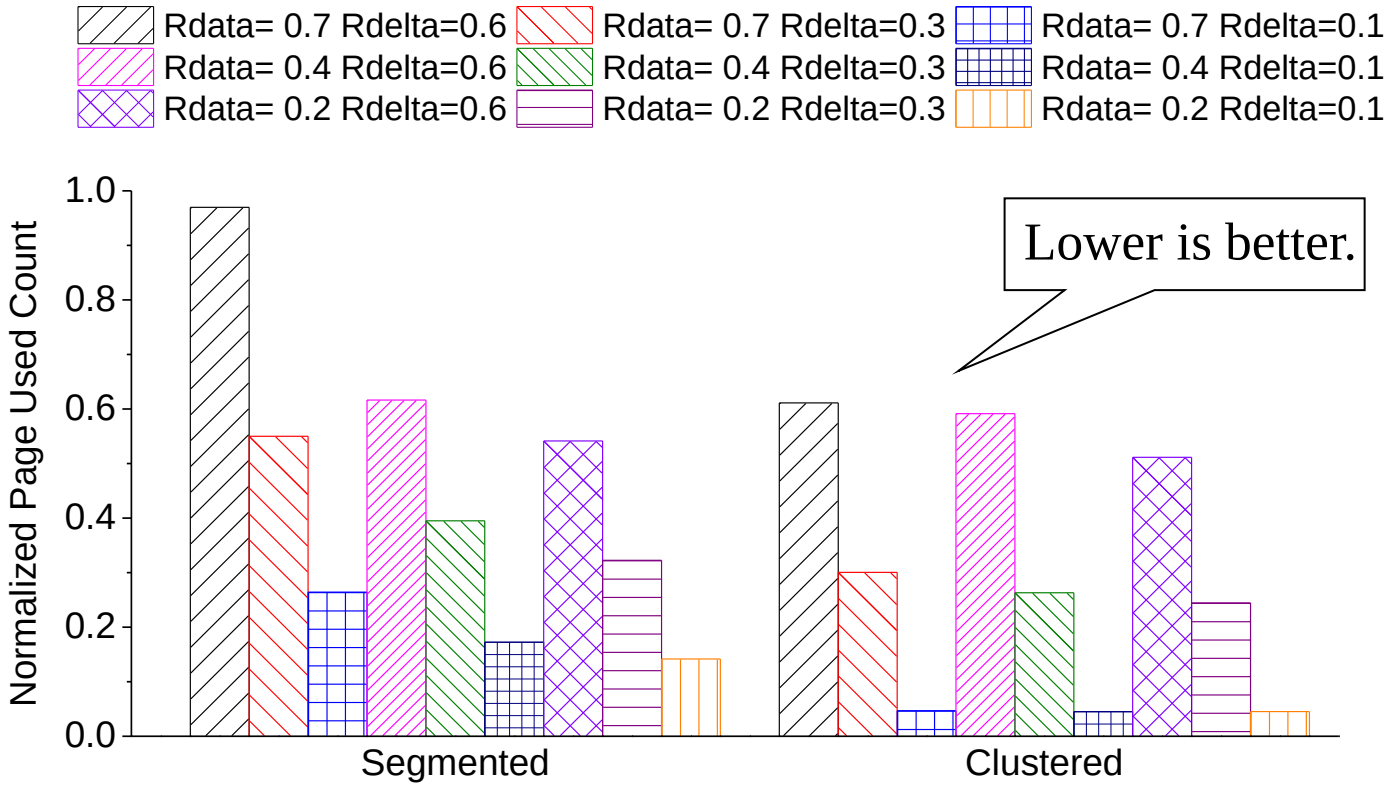
- General cases analysis
  - Set the data/delta compression ratio as Gaussian distribution
  - Simulation driven by traces of different workloads



(a) Webmail

Evaluation: Write Stress Reduction

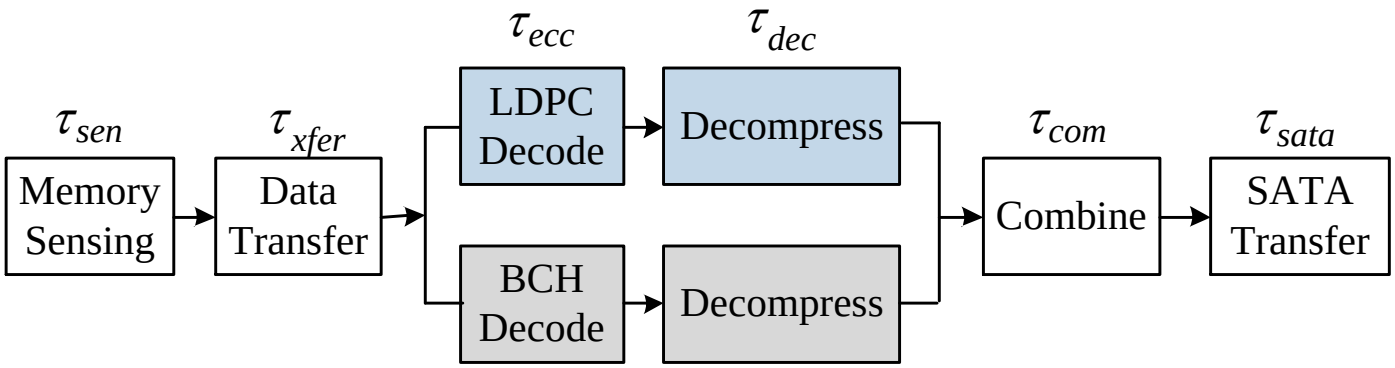
- General cases analysis



(d) TPC-C

Evaluation: Read Latency Overhead

- Read operation's flow diagram in proposed solution



- Read latency model
  - Conventional Practice without data/compression:

$$\tau_{read} = \tau_{sen} + \tau_{xfer}(4kB) + \tau_{LDPC}^{(dec)} + \tau_{sata}$$

  - Proposed solution:

$$\tau_{read} = \tau_{sen} + \tau_{xfer}(n \cdot 4kB) + \max(\tau_{LDPC}^{(dec)}, \tau_{BCH}^{(dec)}) + \max(\tau_{\text{sec}}^{(dec)}, \tau_{delta}^{(dec)}) + \tau_{com} + \tau_{sata}$$

Evaluation: Latency Overhead and Silicon Cost

- Flash memory parameter configurations
  - Flash memory sensing latency: 40 us
  - Data transfer from flash to controller (ONFI 4.0): 800MB/s
  - LDPC/BCH decoding throughput: 1GB/s
- Read latency overhead

Operation	Technique	Average-case (us)	Worst-case (us)
Read	Conventional	54	
	Clustered	76	102
	Segmented	56	63

- Silicon cost
  - Total involved silicon area is 0.39 mm^2 at 22nm, while in genral, a whole controller's silicon area size is around 10-20 mm^2.

Outline

- Introduction and Motivation
- Progressive Programmability of SLC Flash Pages
- In-place Delta Compression Realization in SSDs
- Evaluation Results and Overhead Analysis
- Summary and Future Work

Extensions and Future Work

- Use a hardware accelerator to offload computation intensive tasks
  - ✓ Will not bring troubles to controller because of small silicon size and power consumption
  - ✓ Benefit a lot from the perspective of performance
- Exploit the byte-addressability to realize the delta compression in NVM
  - ✓ Limited endurance, expensive cost
  - ✓ Inherent support of byte-addressability

Conclusion

- SLC Flash page can support "Progressive Partial Programming": different portions of the same flash page can be programmed at different time.
- Lossless data compression can be utilized to make space for deltas between consecutive versions of updates.
- SSD write stress can be reduced by up to 80% with proposed solution without significant overhead.

Thank you!