HUAZHONG UNIVERSITY OF SCIENCE & TECHNOLOGY

NUS National University of Singapore

Space-Oblivious Compression and Wear Leveling for Non-Volatile Main Memories

Haikun Liu, Yuanyuan Ye, Xiaofei Liao, Hai Jin, Yu Zhang, Wenbin Jiang, Bingsheng He*

School of Computer Science and Technology
Huazhong University of Science and Technology
***National University of Singapore**

CGCL

SCTS

Outline

- **Background and Motivations**
- **Our Solution: Space-Oblivious Compression and Wear Leveling**
- **Evaluation**
- **Related Works**
- **Conclusion**

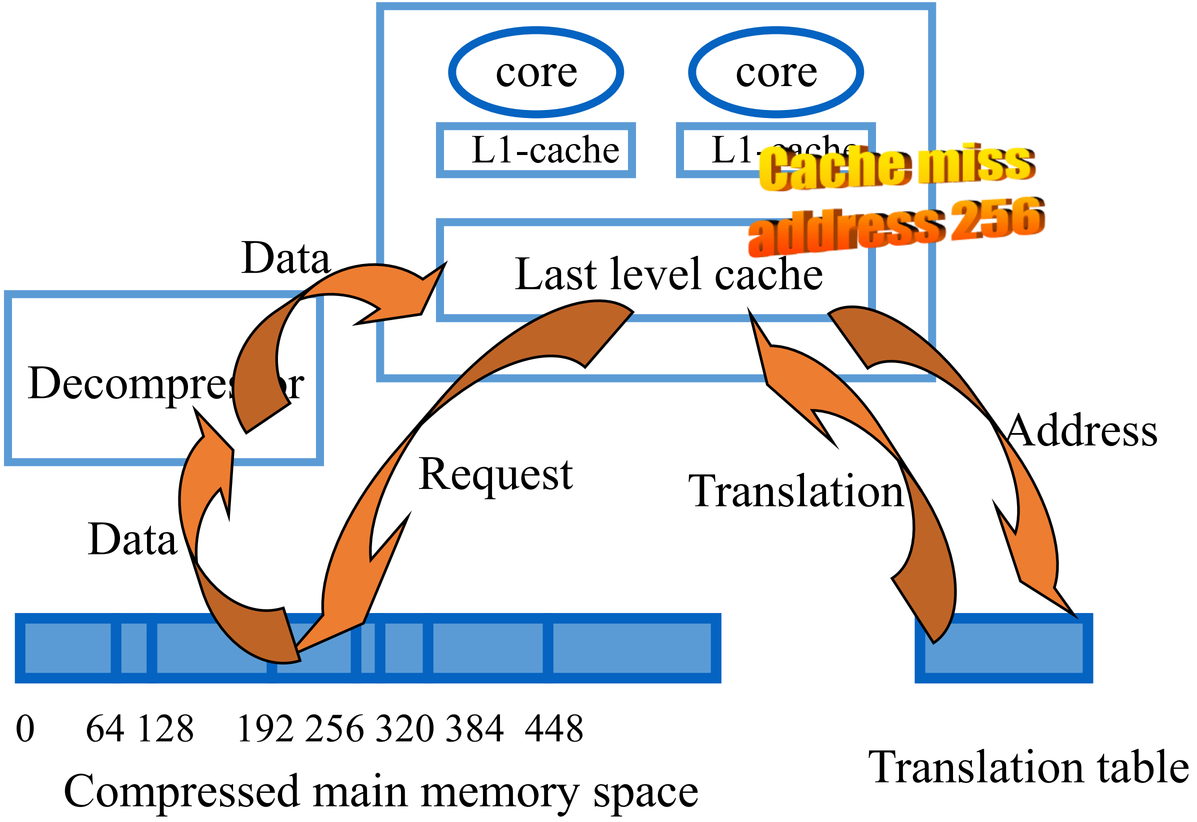
The disadvantages of NVMMs

- Non-Volatile Main Memory (NVMM) has limited write endurance
  - Pros: high density, near-zero static power, non-volatility
  - Cons: limited write endurance, higher write latency and write power

	DRAM	NVM (PCM)	NAND Flash
Read latency	~10 ns	10-100 ns	5−50 µs
Write latency	~10 ns	100-1000 ns	2-3 ms
Write endurance	10^{15}	10^{8}--10^{10}	10^{5}
Non-volatility	No	Yes	Yes
Write power	~0.1 nJ/b	~1 nJ/b	0.1-1 nJ/b

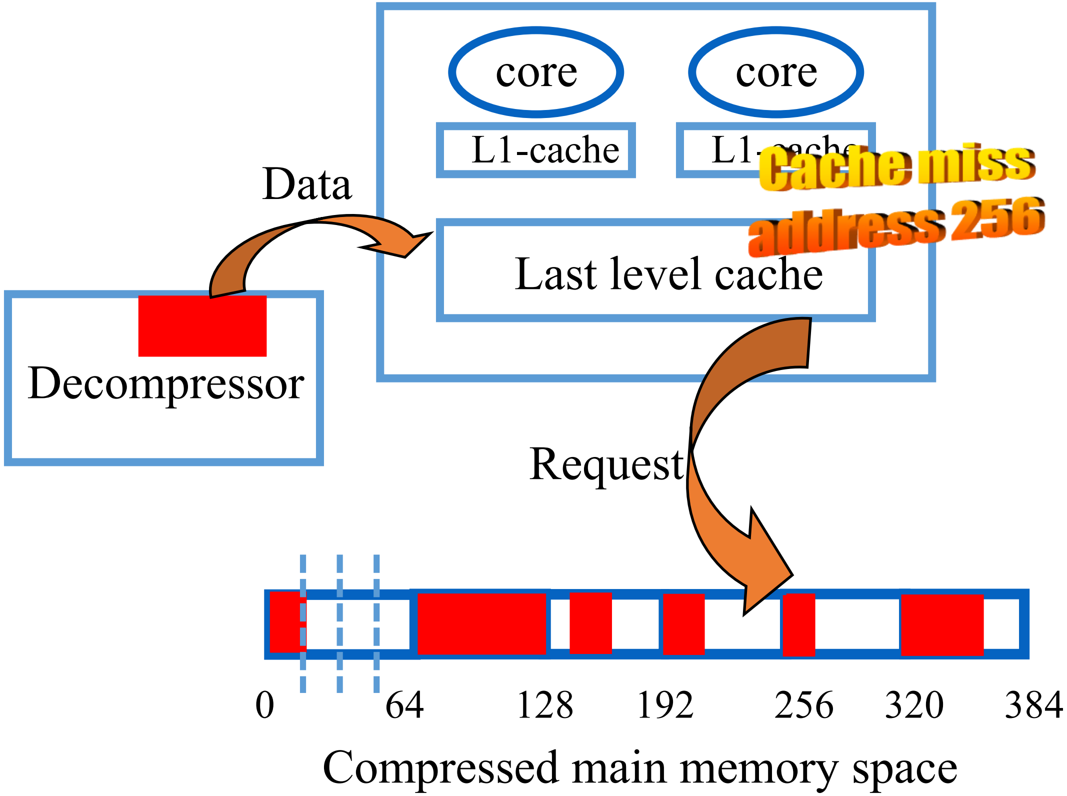
- NVMM lifetime extension techniques
  - Memory compression techniques can reduce bit writes on NVMMs.
  - Wear leveling techniques can balance bit-writes among all NVMM cells.

Memory Compression for Space Saving



- Memory compression techniques (Pros)
  - Save memory space
  - Reduce memory bandwidth consumption
- Memory compression techniques (Cons)
  - An additional memory access for address translation
  - increased memory access latency
  - Complicated Hardware extension

Memory Compression for Wear Leveling

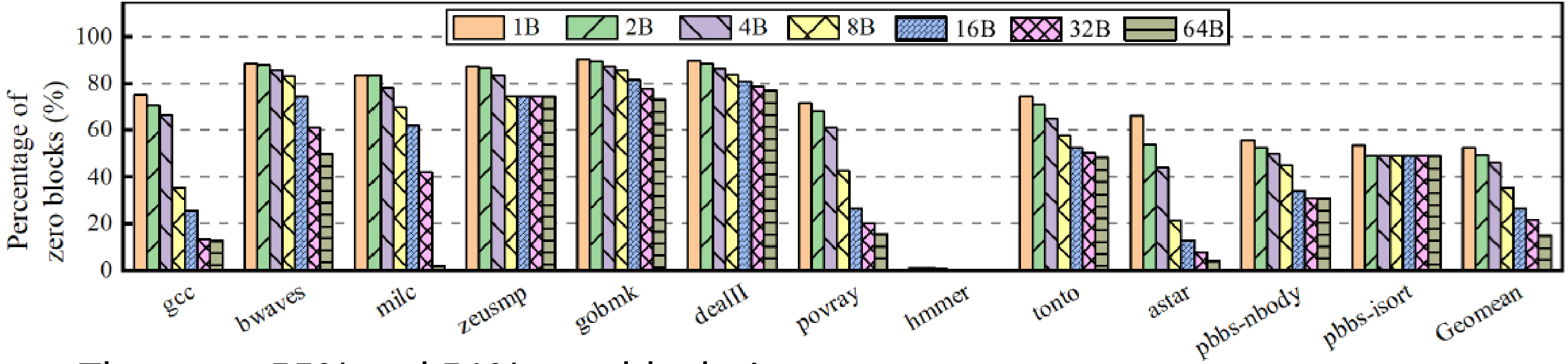


- Memory compression for Wear Leveling
  - Reduce bit writes in NVMMs
  - Reduce memory bandwidth consumption
  - No address translations
  - Space saved by memory compression can be exploited for intra-block wear leveling
  - Trivial hardware extension

Significant Redundancy in Memory

- Application memory usually contain a large fraction of zero blocks

0x00000000	0x0000000B	0x00000003	0x00000004	…

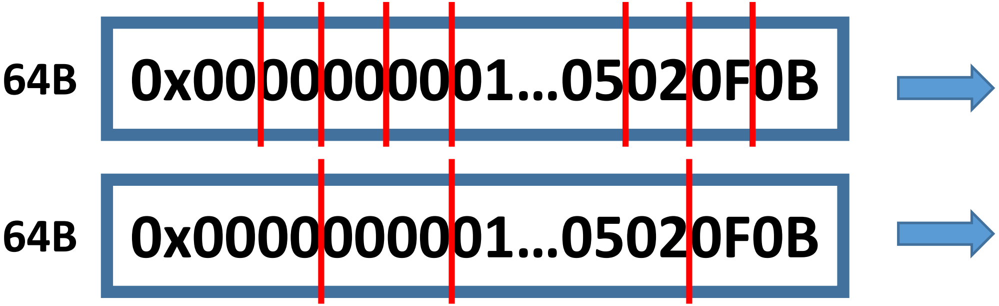


- There are 55% and 51% zero blocks in memory on average when the data sizes are 1B and 2B.
- Even 15% of 64B blocks are all zeros.

→ A smaller block improves compressibility for zero-based memory compression

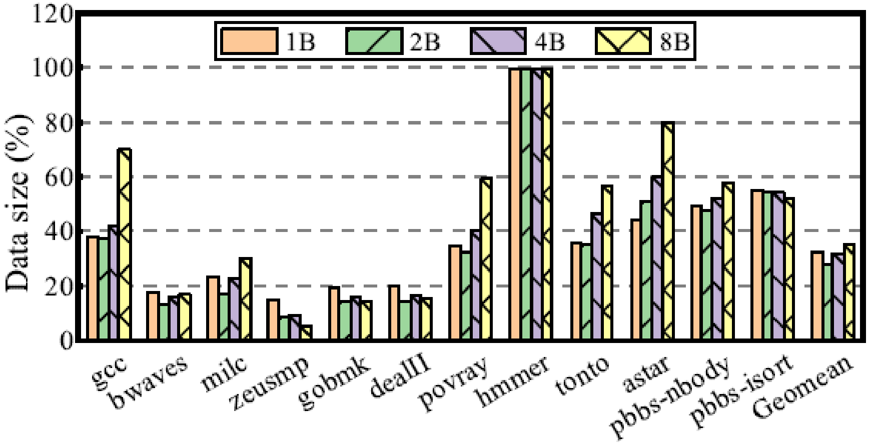
Significant Redundancy in Memory

- How to determine the optimal block size for compression?



64 bits encoding for zero-based memory compression

32 bits encoding for zero-based memory compression

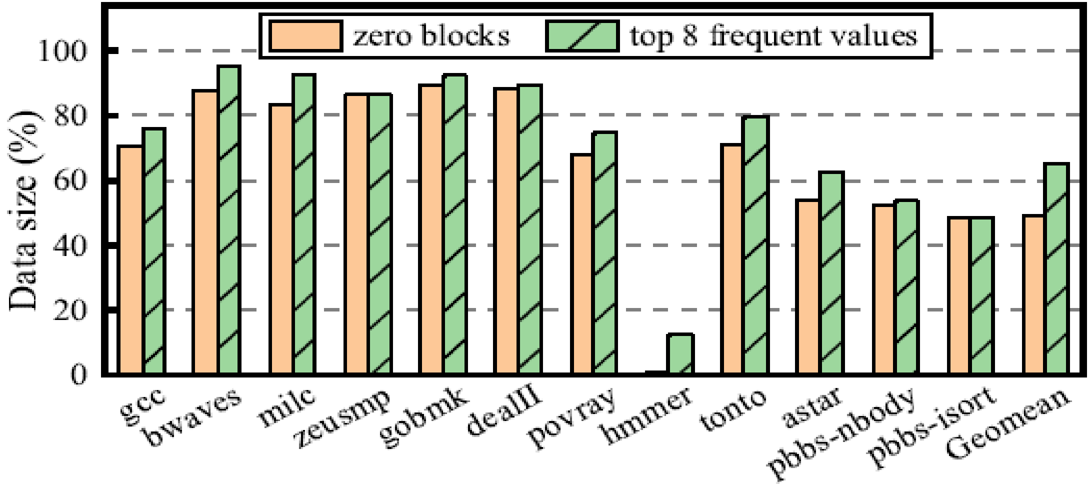


- Small sub-blocks potentially improve the compression ratio, but increase the size of compression metadata.
- We find that the size of compressed data including compression metadata is minimized when the block size is set as 2B.

Significant Redundancy in Memory

- Application memory usually contain many frequent values

0x00000001	0x00000001	0x00000002	0x00000001	…



The fraction of zero blocks and the top 8 frequent values in application's memory when the block size is 2B.

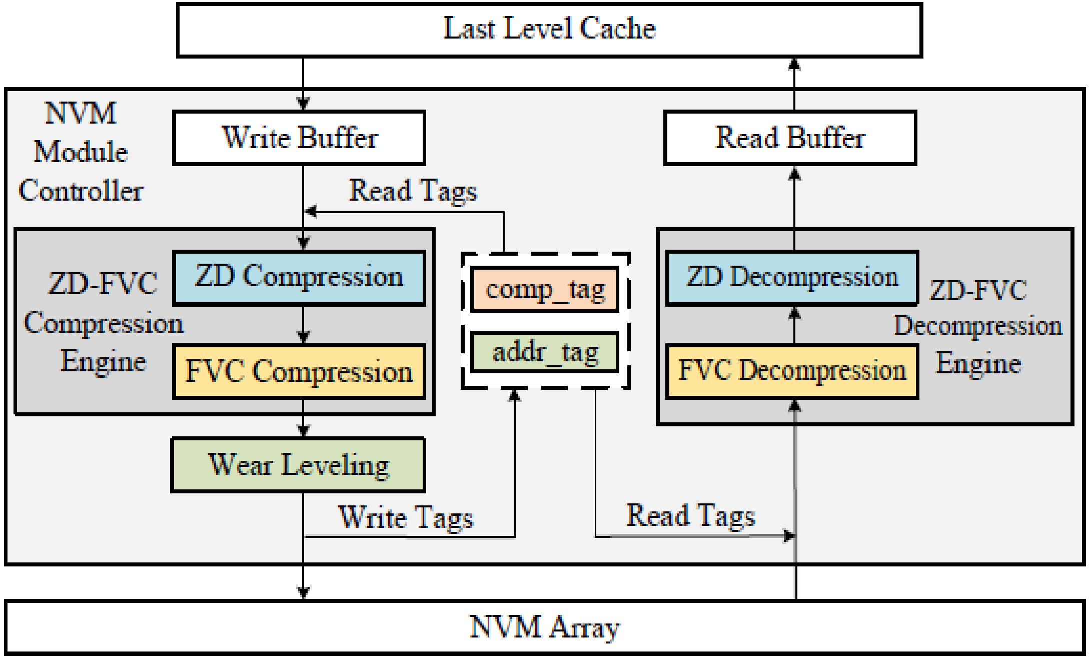
- The top 8 frequent values are 0, 1, 2, 4, 3, -1, 5, and 8.
- The zero values account for a majority of frequent values.

→ Non-uniform encoding scheme for frequent value compression

Outline

- Background and Motivations
- **Our Solution: Space-Oblivious Compression and Wear Leveling**
- Evaluation
- Related Works
- Conclusion

NVMM Compression Architecture

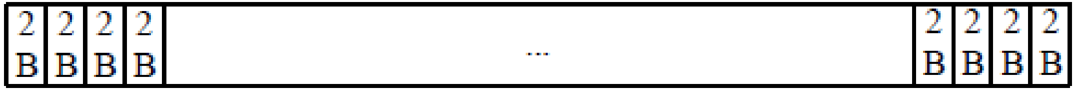


■ ZD-FVC Compression

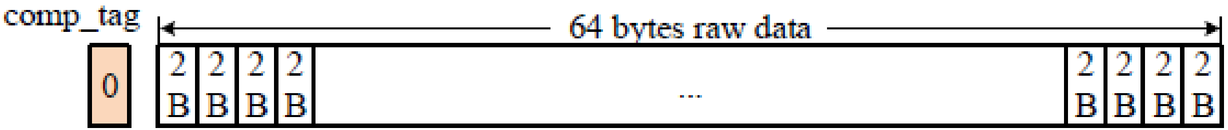
- Integrate Zero Deduplication (ZD) and Frequent Value Compression (FVC) together
- A wear leveling policy is achieved by exploiting the memory space saved by memory compression.
- Use reserved bits of error-correcting code (ECC) to store 2-bit compression tags (comp tag) and 2-bit wear leveling tags (addr tag)

Zero Deduplication

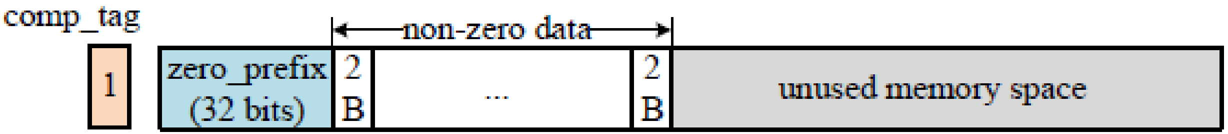
- We divide a cache line into 32 sub-blocks, and use 32 bits (called zero_prefix) to identify the zero-valued sub-blocks
- The number of zero bits in the zero_prefix should be larger than 2 because the zero prefix spends 4 bytes



(a) 64B cache line, split into 32 blocks of 2B data



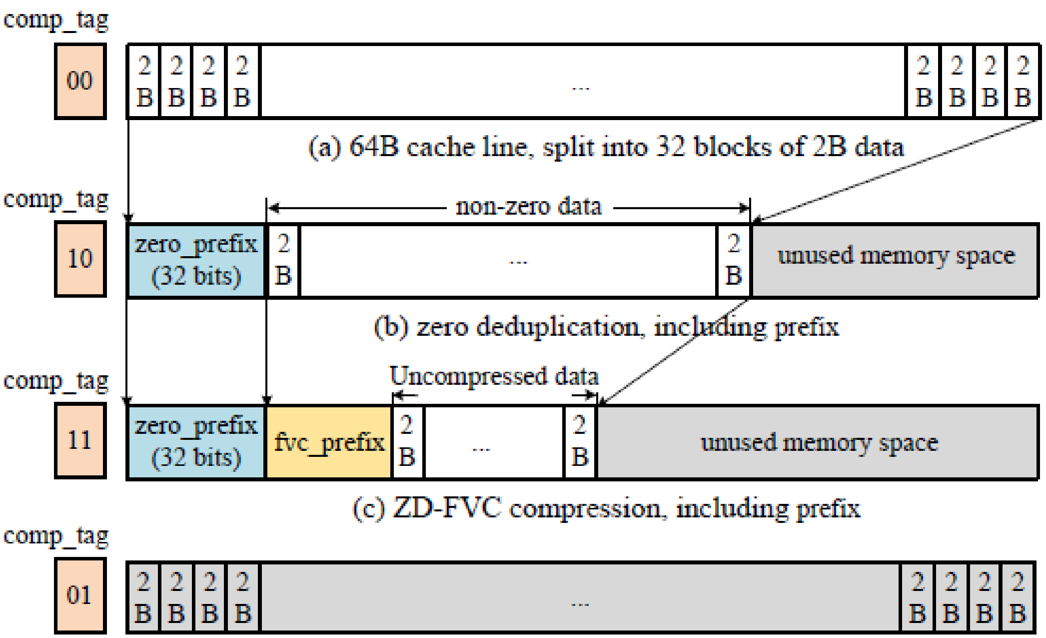
(b) comp_tag = 0, data is not compressible



(c) comp_tag = 1, data is compressible

Integrating ZD with FVC

Frequent Values (2B)	-1	1	2	3	4	5	8	Other
Encoding (3 bits)	000	001	010	011	100	101	110	111



(a) 64B cache line, split into 32 blocks of 2B data

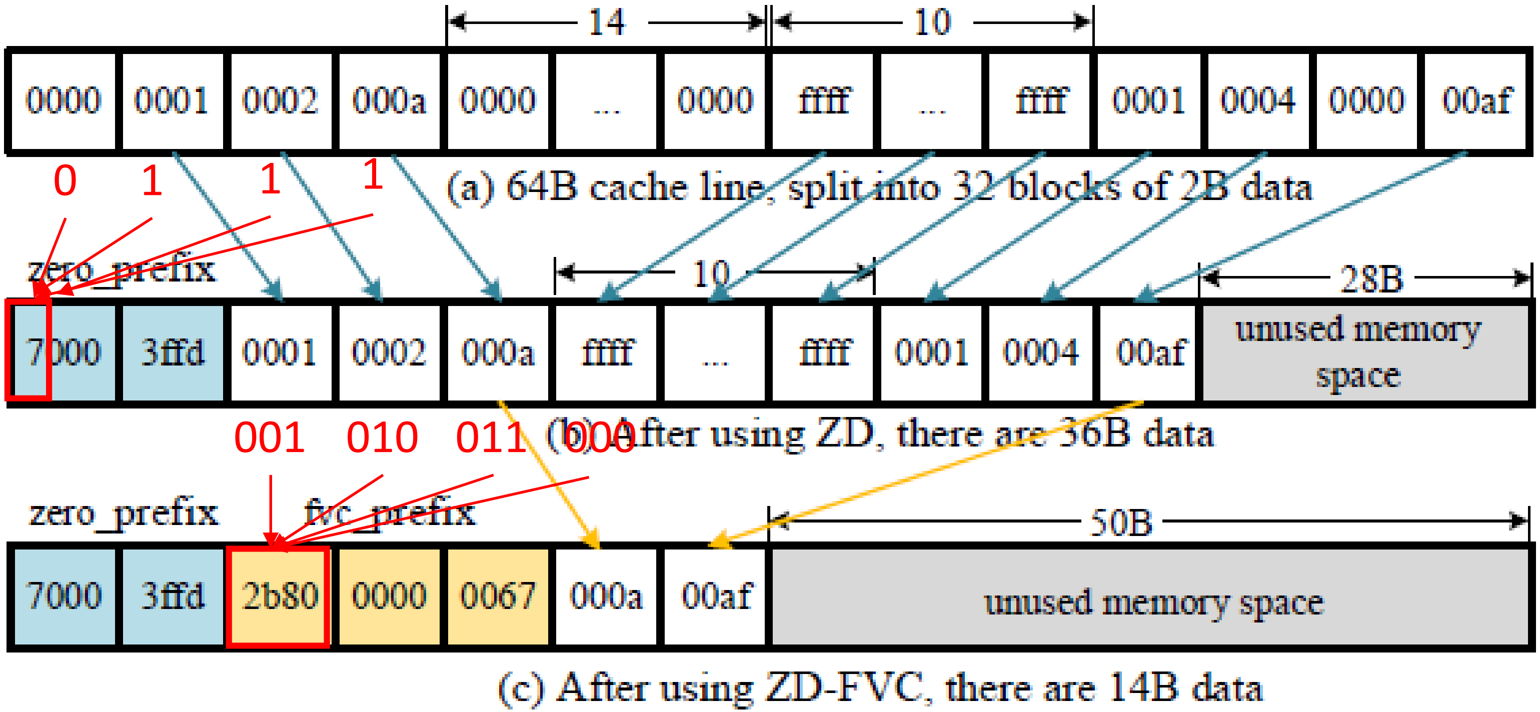
(b) zero deduplication, including prefix

(c) ZD-FVC compression, including prefix

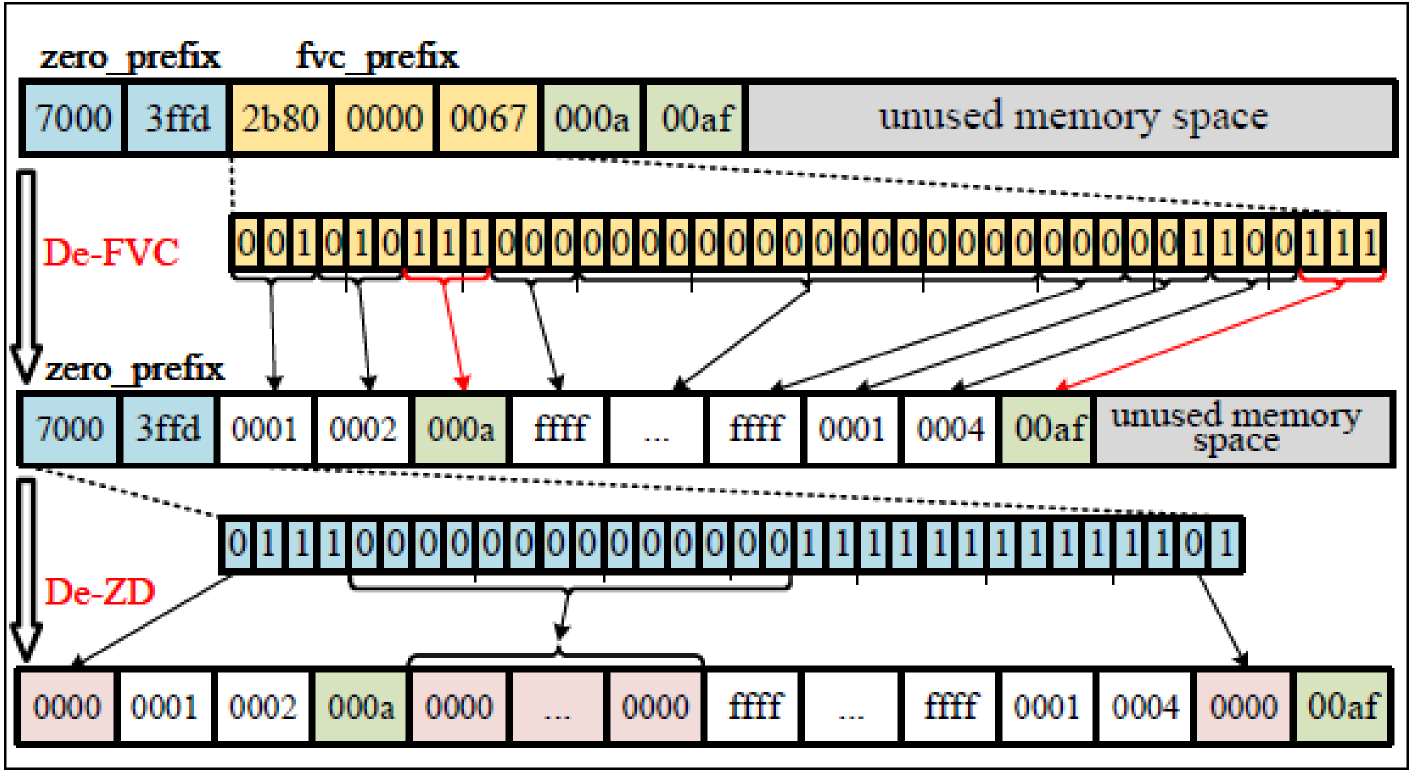
(d) comp_tag = 01, all 32 blocks are zero bytes

- We extend the comp_tag to 2 bits to identify different compression schemes.
- Storage overhead of compression codes
  - 1 bit for each zero sub-block;
  - 4 bits for each non-zero sub-block (ZD and FVC use 1 bit and 3 bits in the zero prefix and fvc prefix);
  - ZD-FVC is better than FVC if the proportion of zero sub-blocks exceed 34%

An Example of ZD-FVC

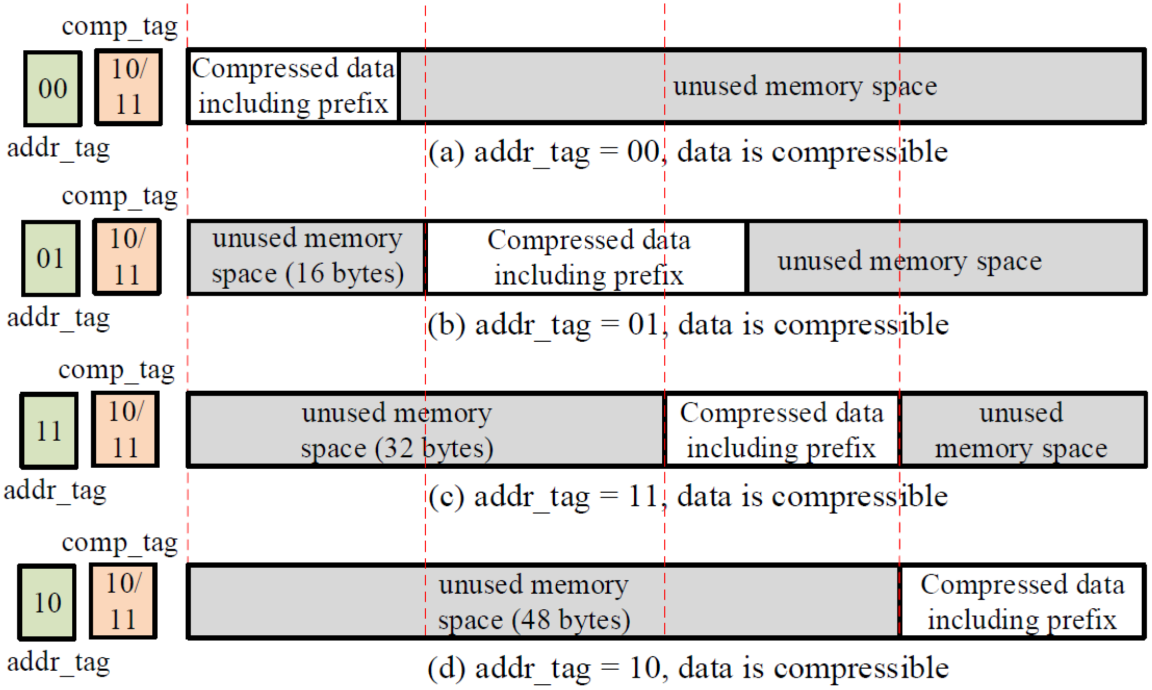


Decompression of ZD-FVC



Wear Leveling

- divide the 64-byte memory block into four sections evenly
- use 2-bit addr tag to locate the starting address of compressed data



The current data address (addr tag) is determined by the value of *comp_tag*, the previous *addr_tag*, and the size of compressed data.

Outline

- Background and Motivations
- Our Solution: Space-Oblivious Compression and Wear Leveling
- **Evaluation**
- Related Works
- Conclusion

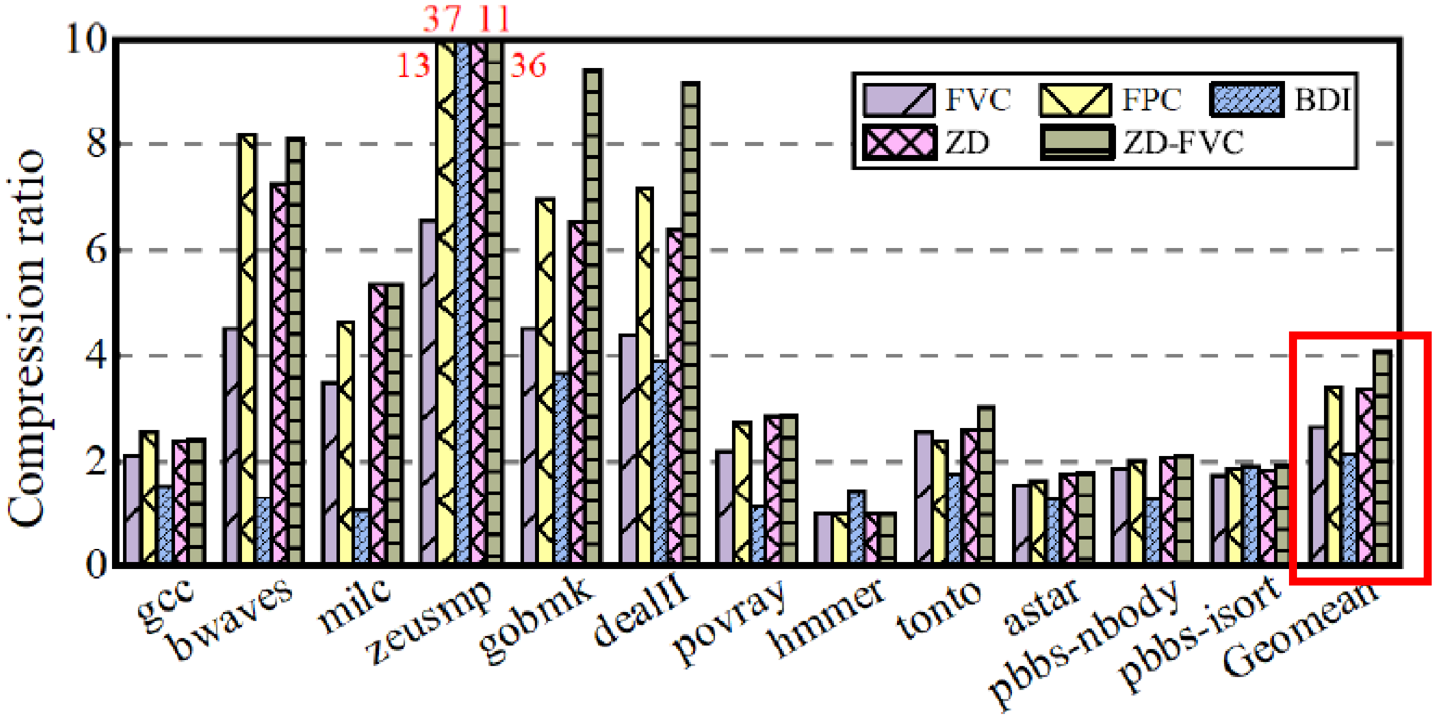
Experimental setting

- **Simulators:** Gem5 + NVMain

CPU	out-of-order, 2 GHz, 8 cores
L1 cache	32 KB separated icache and dcache, 2 cycles
L2 cache	1 MB, 20 cycles
L3 cache	16 MB, 50 cycles
PCM	Capacity: 4 GB Controller: FRFCFS scheduler Bus Frequency: 400 MHz Timing (tCAS-tRCD-tRP-tRAS): 5-22-60-17 (cycles) Energy: 81.2 PJ/bit for read, 1684.8 PJ/bit for write

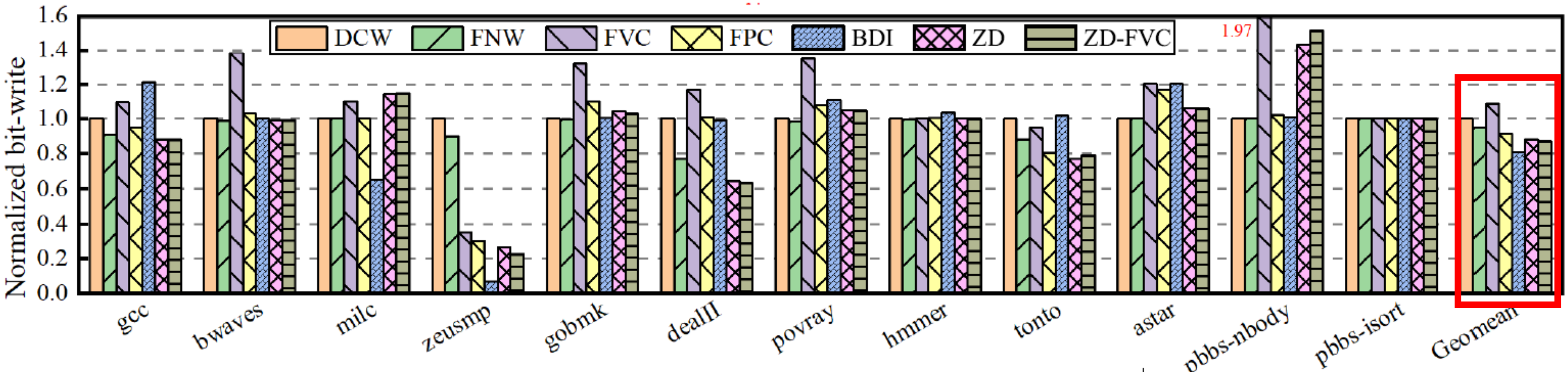
- **Benchmarks:** SPEC CPU 2006 benchmark, Problem Based Benchmark Suite (PBBS)
- **Comparisons:** Data Comparison Write (DCW), Flip-N-Write (FNW), Frequent Value Compression (FVC), Frequent Pattern Compression (FPC), and Base-Delta-Immediate Compression (BDI)

Memory Compression Ratio



The average compression ratio of ZD-FVC is about 4.

Bit-write Reduction



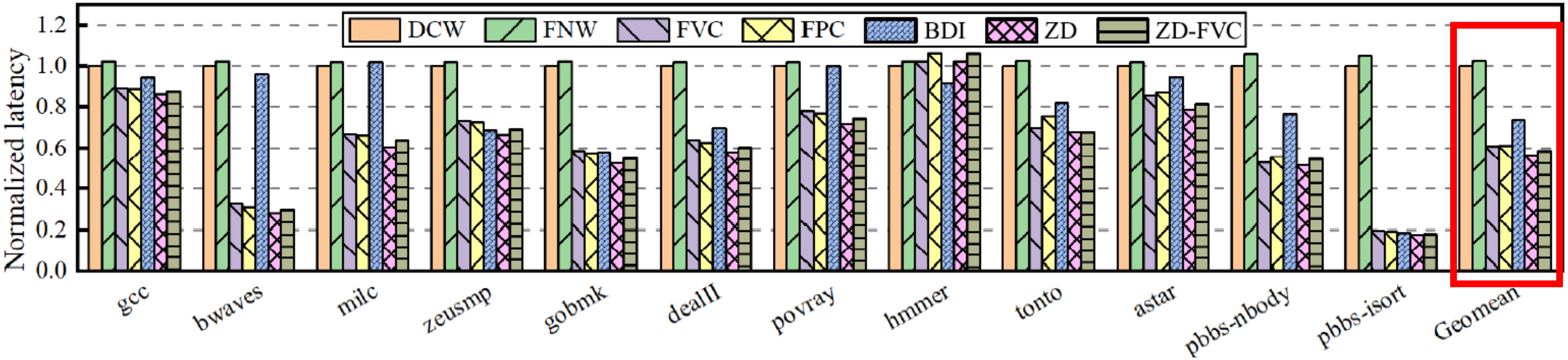
ZD-FVC can reduce the bit-writes by 15% on average compared with DCW (a typical differential write scheme).

NVMM Access Latency

Schemes	Write (cycles)	Read-1[a] (cycles)	Read-2[b] (cycles)
DCW	2	0	0
FNW	4	1	2
FVC	4	1	5
FPC	8	1	5
BDI	8	1	2
ZD	4	1	5
ZD-FVC	8	1	7

[a]Data is not compressed. [b]Data is compressed.

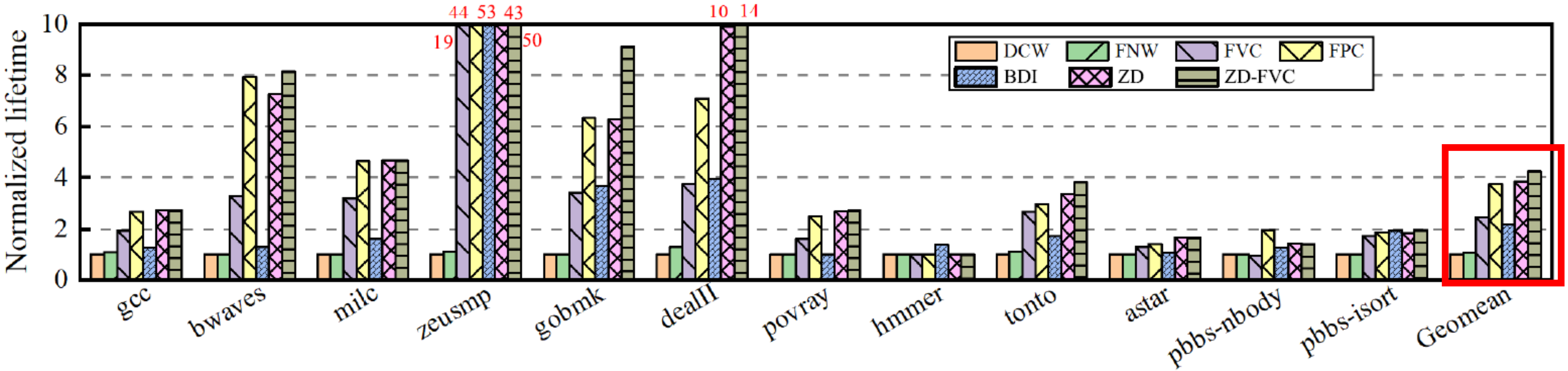
ZD-FVC can reduce the accumulated NVMM access latency by 42% compared with DCW.



NVMM Lifetime Improvement

$$lifetime = \frac{C \times R}{N}$$

C: the capacity of NVMM
R: memory compression ratio
N: the number of bit-writes



ZD-FVC can significantly improve the lifetime of NVMM by 3.3X compared with DCW. Because Memory compression can increase the available NVMM capacity to some extent.

Conclusion

- **Problem:** Limited write endurance is a major drawback of Non-Volatile Main Memory (NVMM) technologies.
- **Observation:** Memory blocks of many applications usually contain a large amount of zero bytes and frequent values.
- **Key ideas:** 1) We propose a non-uniform compression encoding scheme that integrates Zero Deduplication with Frequent Value Compression (called ZD-FVC) to reduce bit-writes on NVMM. 2) We leverage the memory space saved by compression to achieve intra-block wear leveling.
- **Results**: The new NVMM architecture can integrates memory compression and wear leveling together seamlessly, and can improve the lifetime of NVMM by 3.3X.

Thank you! Questions?