Australian National University

# Tolerating Holes in Wearable Memories

Tiejun Gao[1], Stephen M Blackburn[1], Karin Strauss[2], Kathryn McKinley[2], Jim Larus[2], Doug Burger[2]

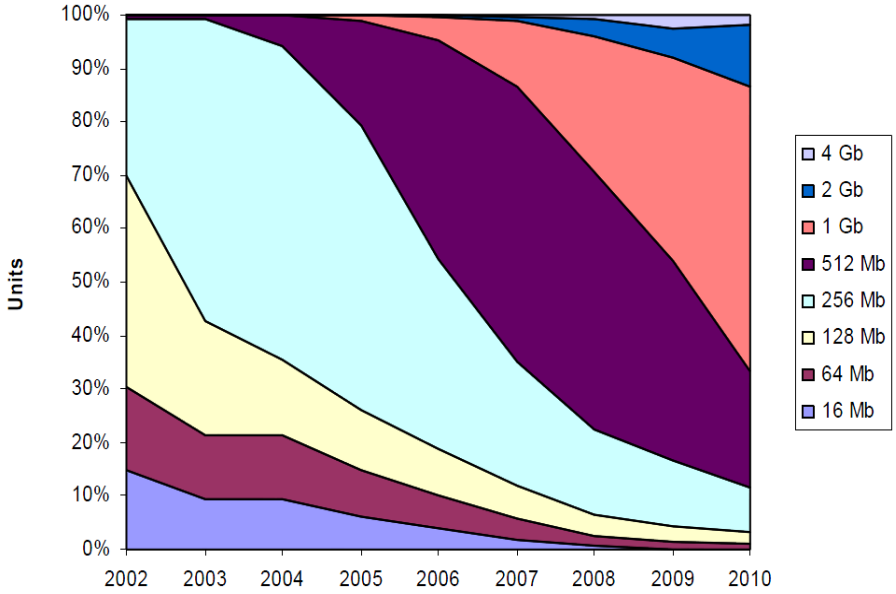
1: The Australian National University

2: Microsoft Research

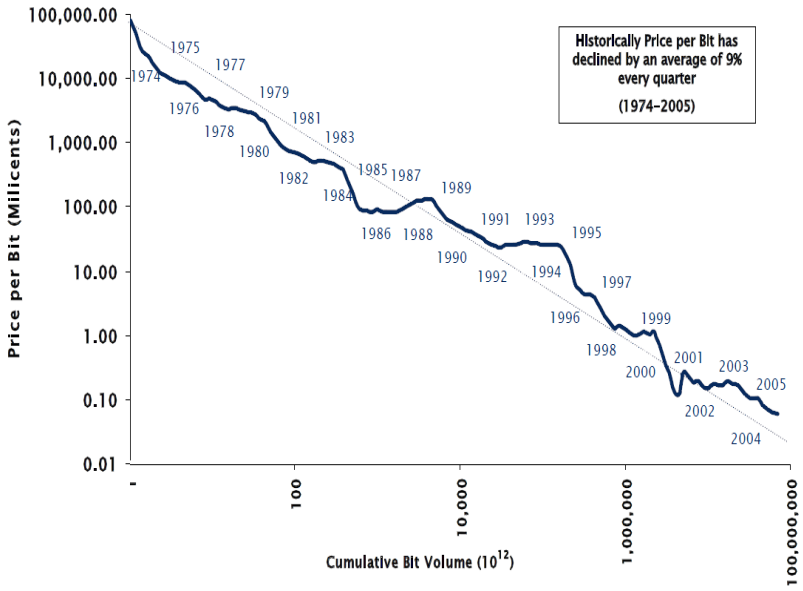
# Background: Wearable Memory

# DRAM

- Demands for higher capacity and lower cost
- The trend may be broken in the near future




DRAM Density Trend




Price per bit Trend

# Background: Wearable Memory

## DRAM

- Charge-based volatile memory

## Problems

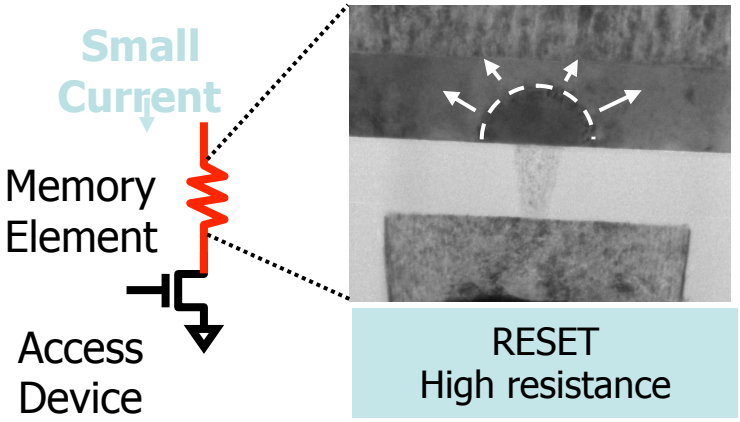
- Charge leakage
- Cosmic particle hitting
- High refresh rate

# Alternatives to DRAM

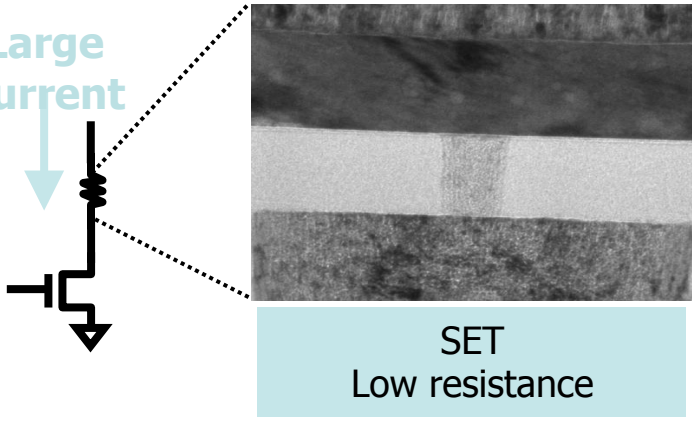
- Resistive memories
  - Non-volatile
  - Stable
  - Low refresh rate
- Different material system
  - Binary transition metal oxides
  - Solid-state electrolytes
  - Phase change chalcogenides

# Phase Change Memory (PCM)

- Non-volatile resistive memory
- Switching by heating using electrical pulses



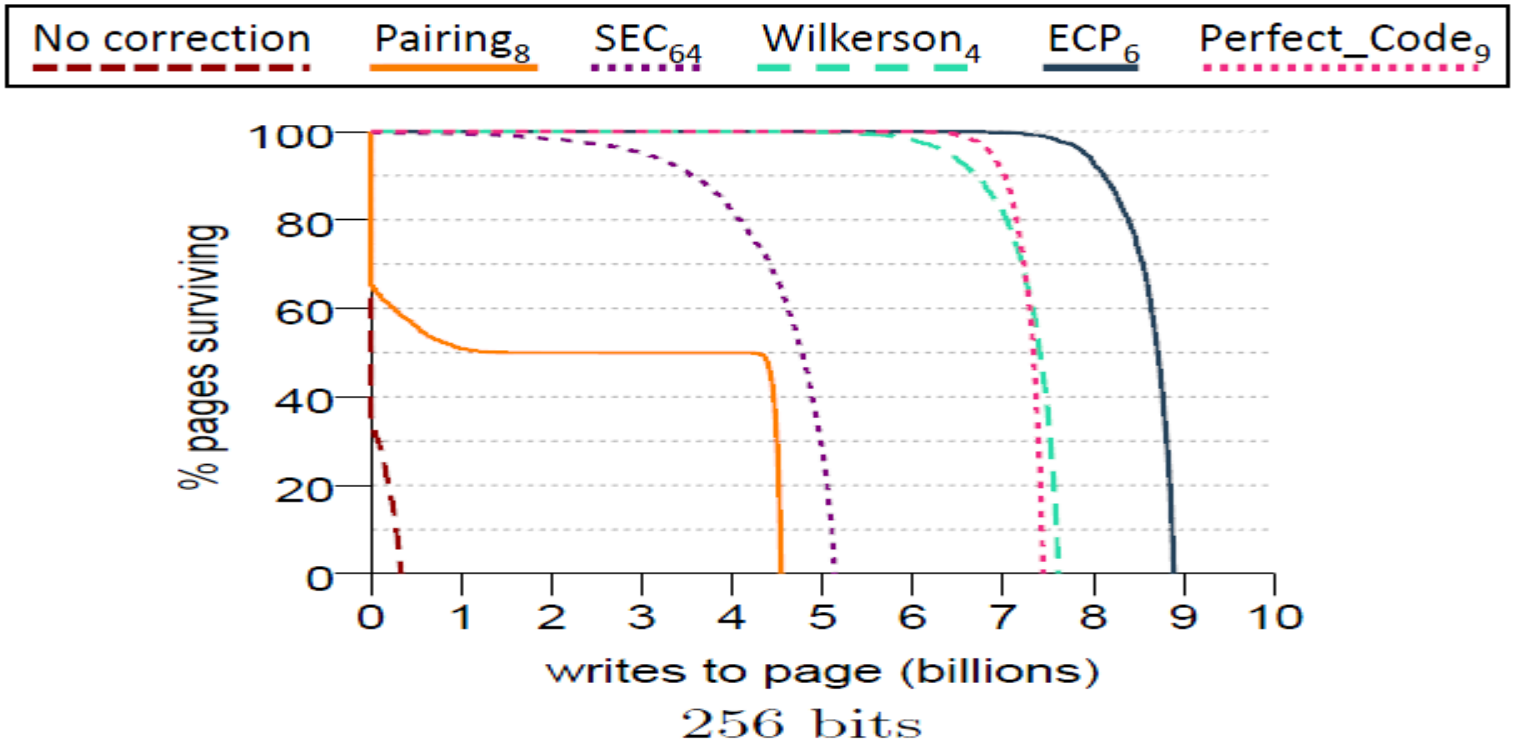

Amorphous state




Crystalline state

# Background: Wearable Memory

- Typical write limit: PCM $10^8$, DRAM $10^{15}$
- PCM line (64B)
- Hardware error correction mechanism



# Problem and opportunity

- Current Memory Failure Model
  - Discard entire page for one failed line (98% of the memory is wasted)
  - Pages die very fast when hardware error correction resources run out.
  - Fewer surviving pages support more writes
- How about we use the failed pages?

# Failures are exposed to applications

- Live data should never occupy failed memory
  - Code and data segment
  - Heap
- Significant changes needed for native program

# Managed runtime

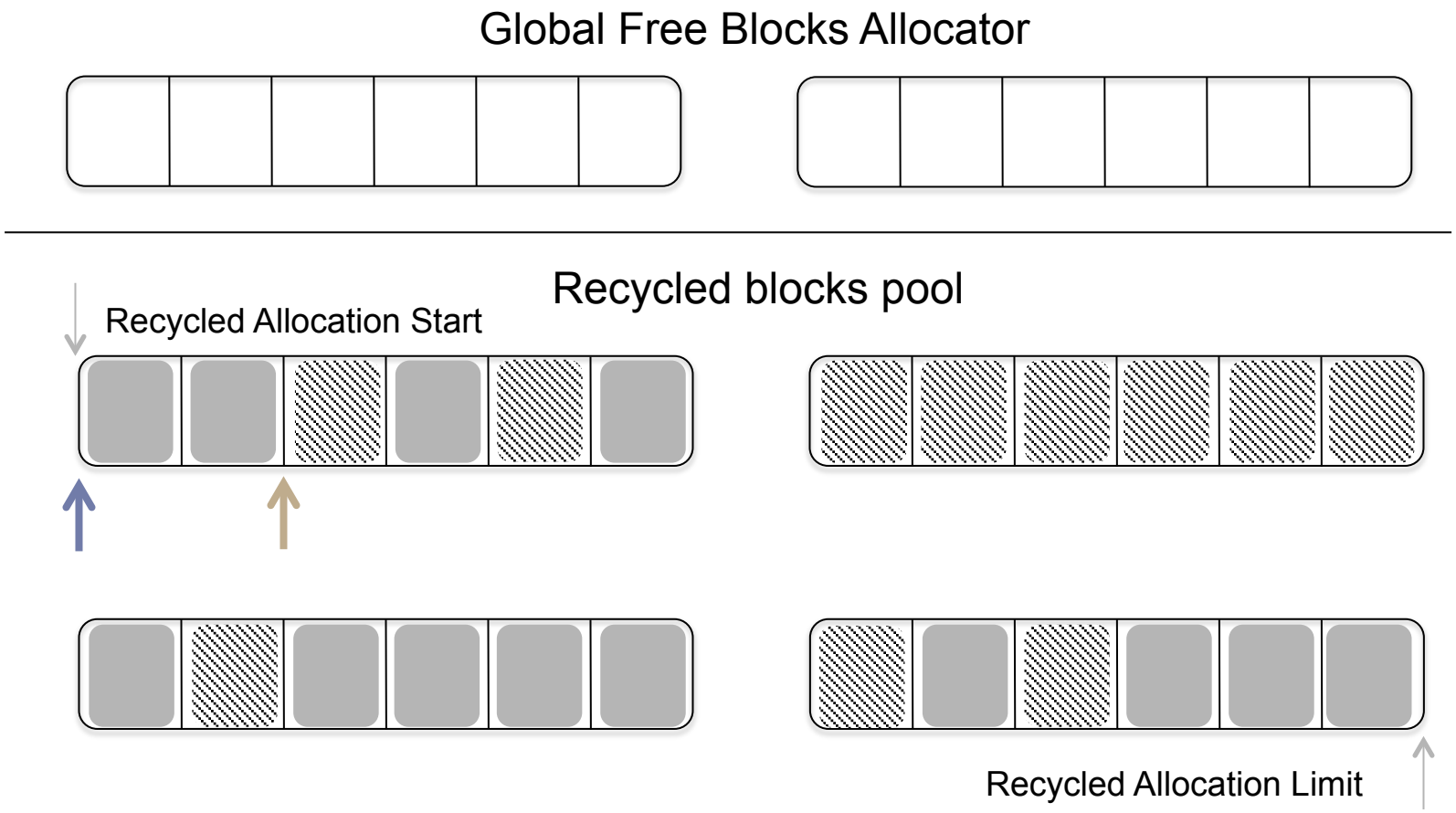
- Safe pointer discipline
  - References, not addresses
- Dynamic compilers
  - Lay out code around failures
- Garbage collection
  - Move data transparently and correctly
- No change is needed for applications

# Background: Immix in JikesRVM

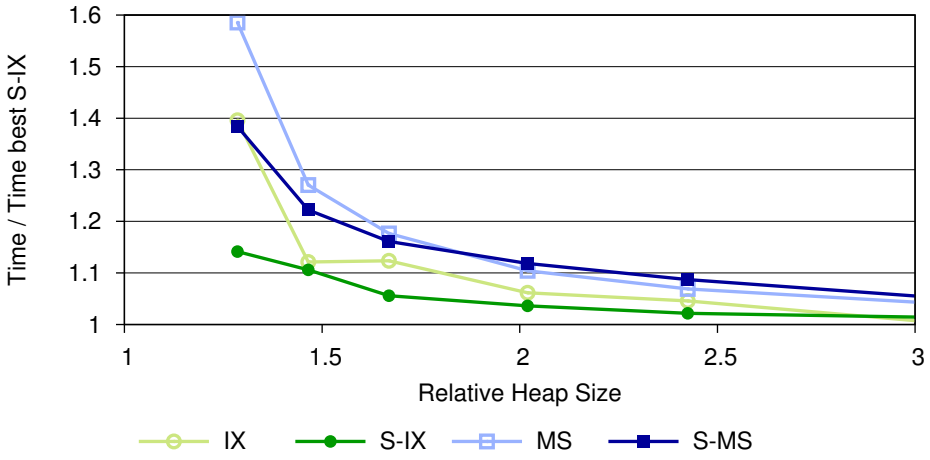
# Immix algorithm in JikesRVM

- Different granularities
  - Coarse block (32KB) and fine line (256B)
- Good space efficiency, collection time and locality
  - Bump pointer allocation in blocks
  - Linearly scan the line map and identify free and partially free blocks
  - Utilize recyclable blocks

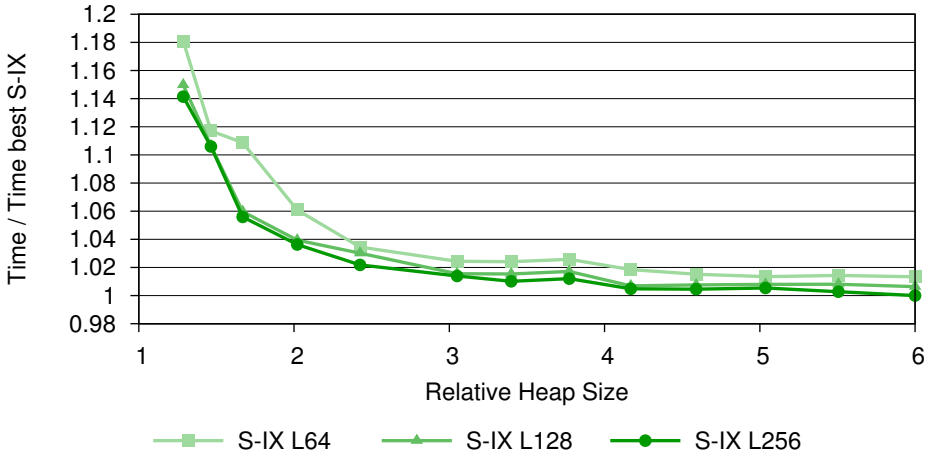
# How Immix works



# Immix algorithm in JikesRVM




StickyImmix outperforms MarkSweep and StickyMarkSweep




256B immix line size outperform 128B and 64B

# Failure-aware Memory Management

# Hardware and OS support

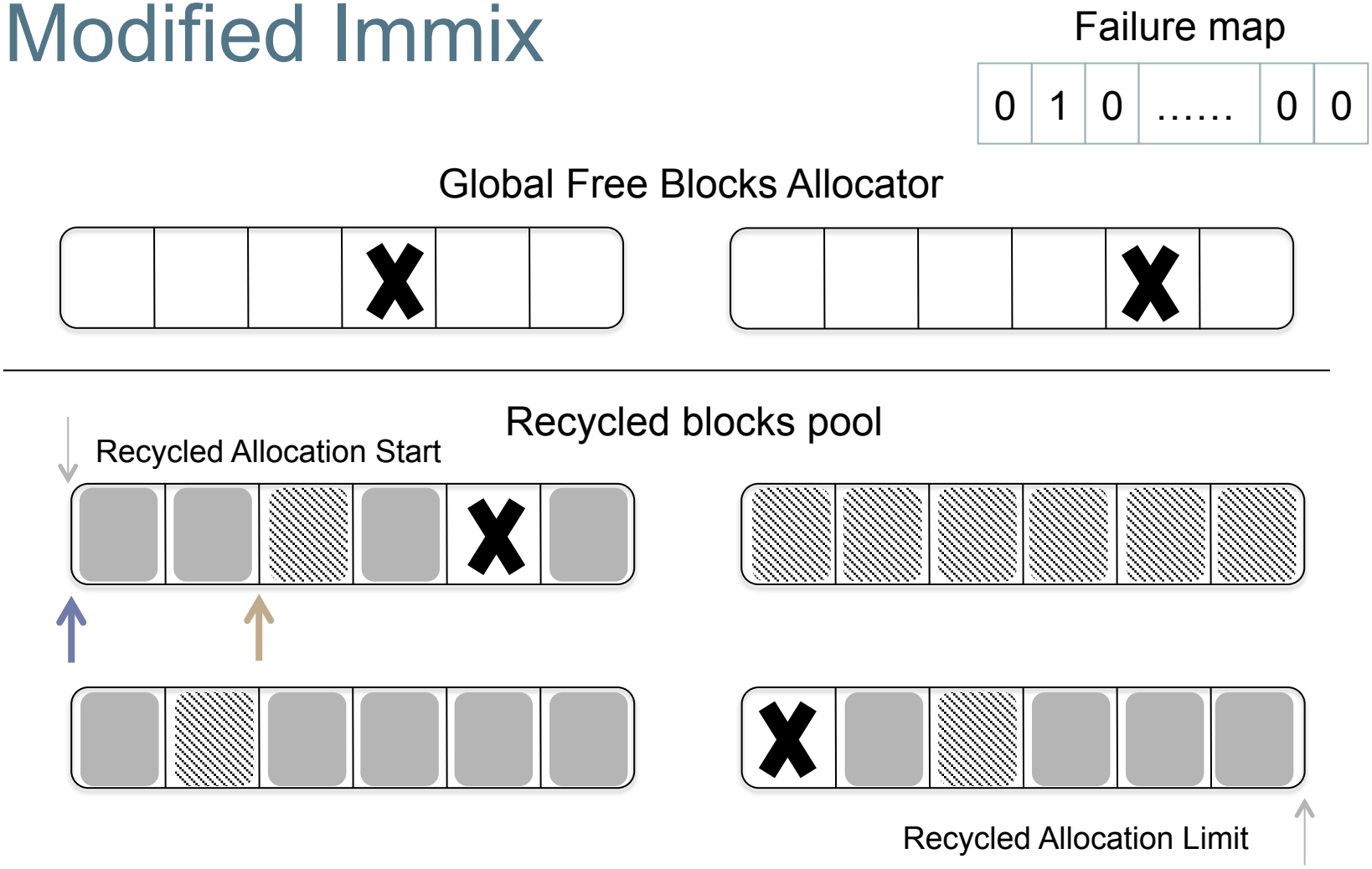
- Hardware support
  - Interrupt CPU to inform software of failures
  - Maintain correct data in failure buffer
- Operating System
  - Both PCM and DRAM are in use
  - Notify applications when failures occur
  - Replace faulty pages

# Static and dynamic failures

- Static failure
  - Known and recorded when VM starts
  - Regarded as used immix lines
- Dynamic failure
  - Trigger a full GC
  - Copy affected objects
  - Update the failure map
  - Use perfect pages if necessary.

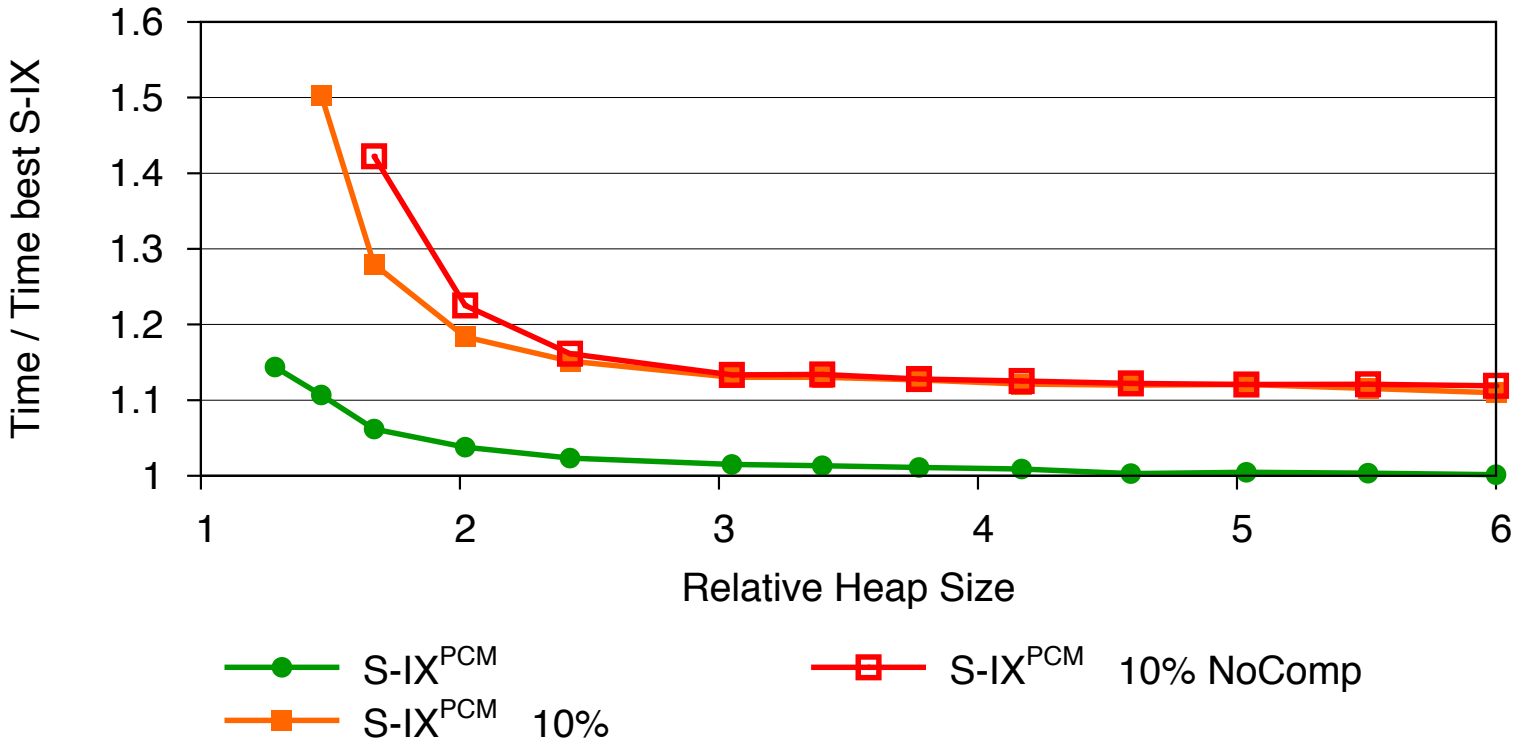
# Failure-aware Memory Management

## Modified Immix



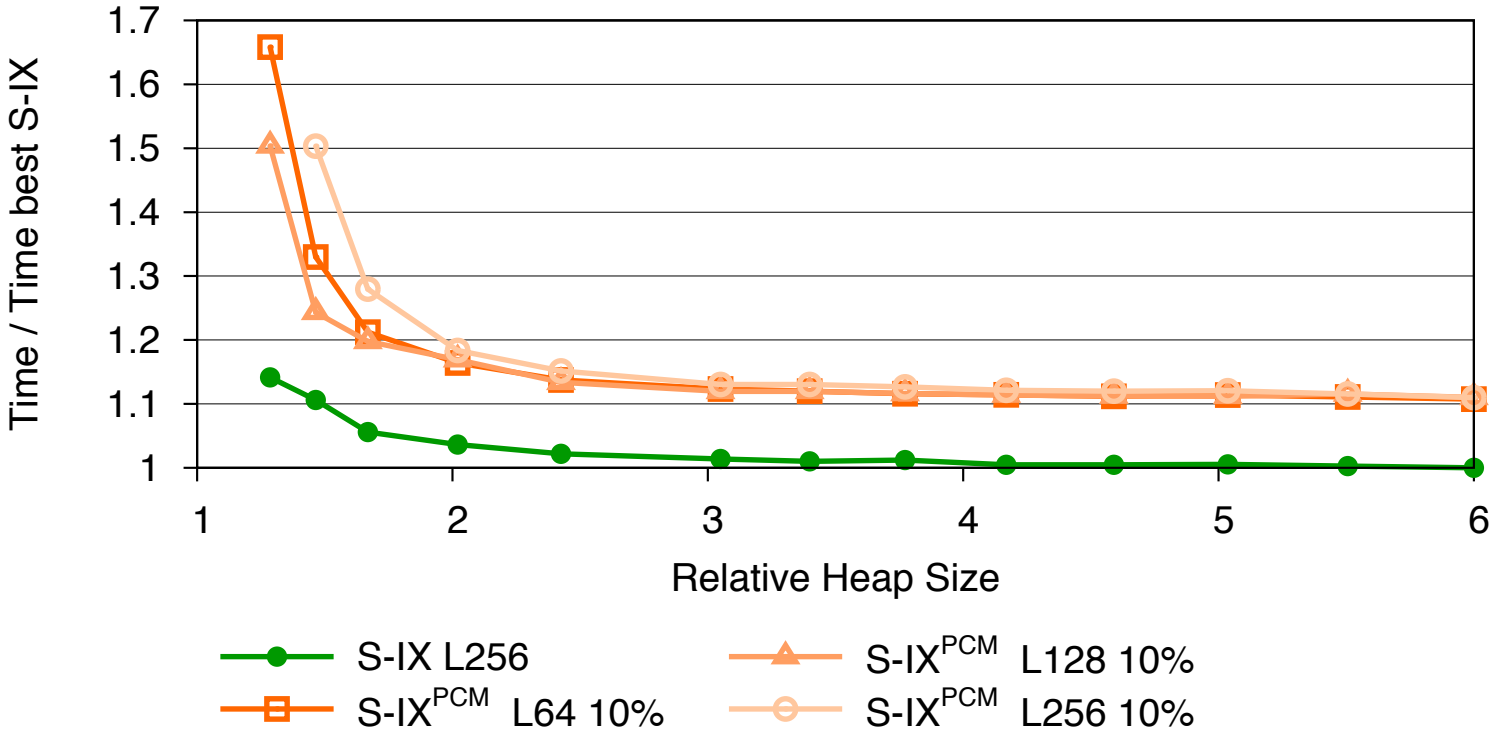
# Influence of failures

- Memory loss



# Influence of failures

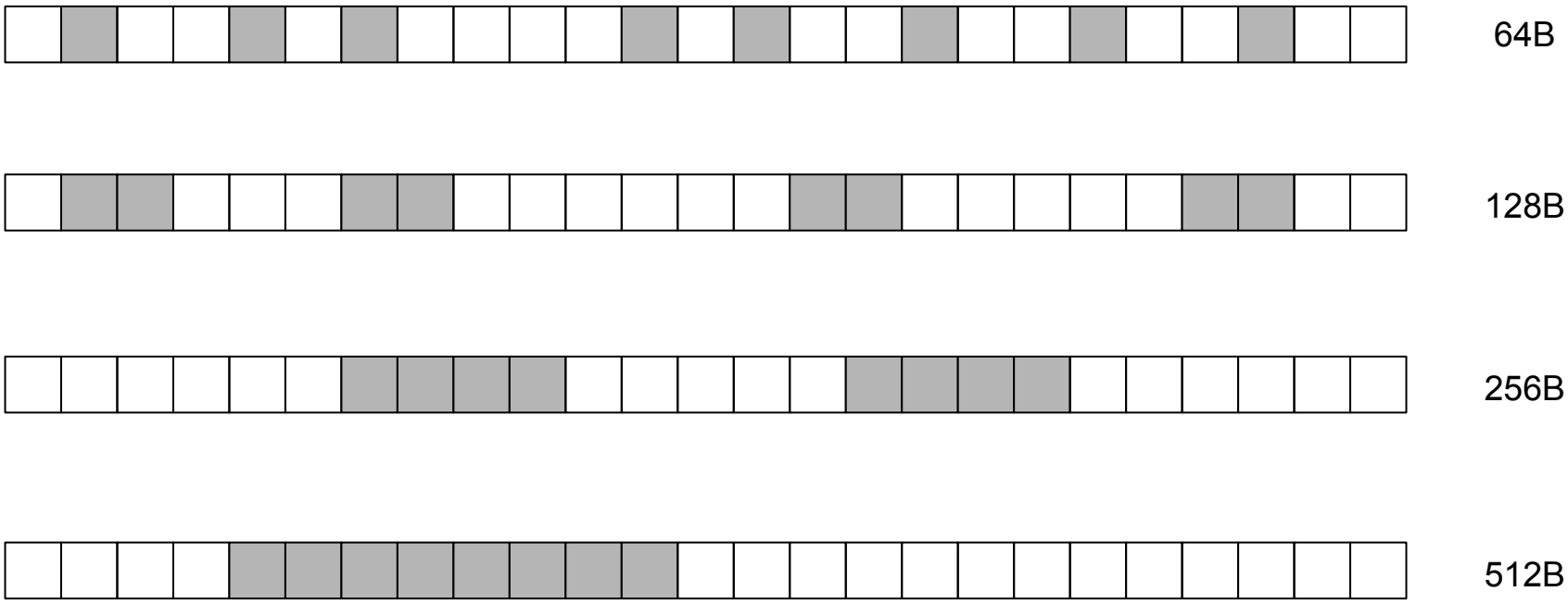
- Fragmentation



# Failure Cluster Mechanism

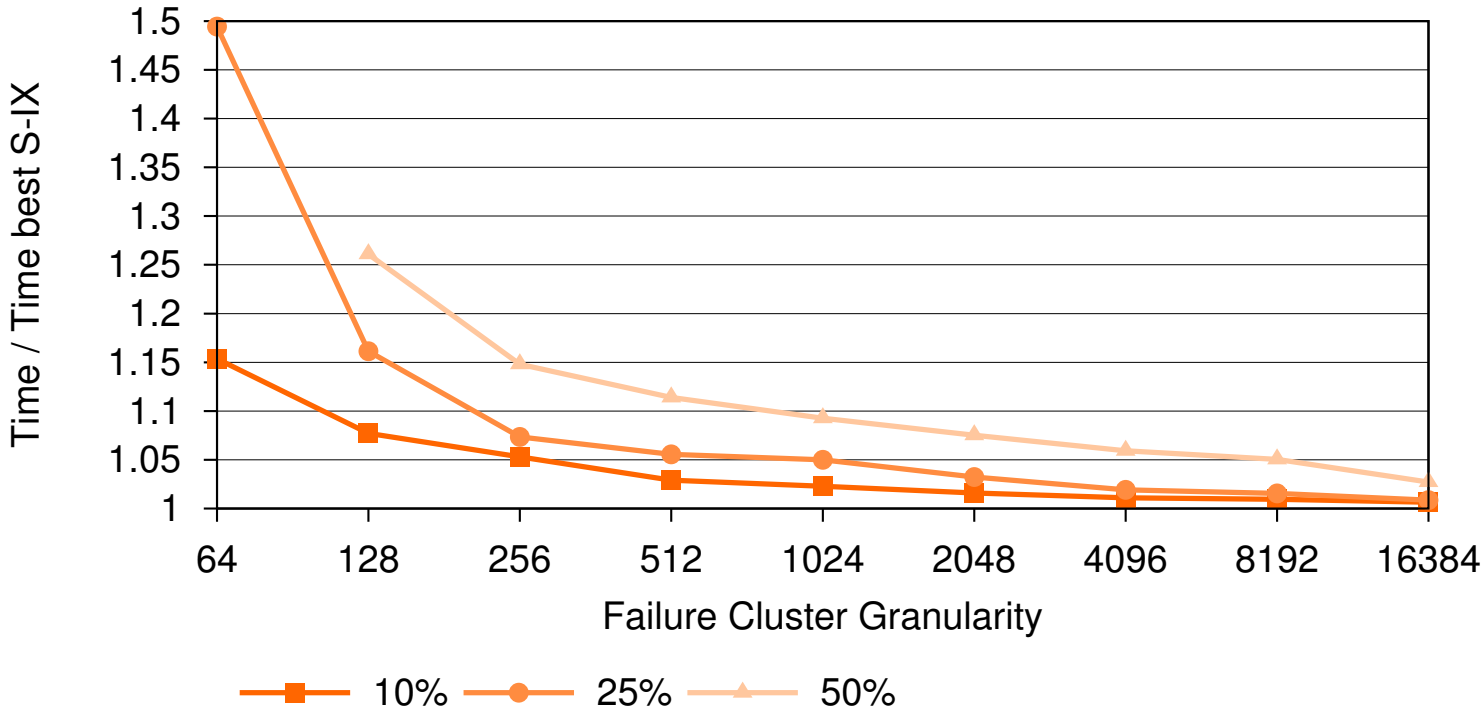
# Influence of failures

- Fragmentation

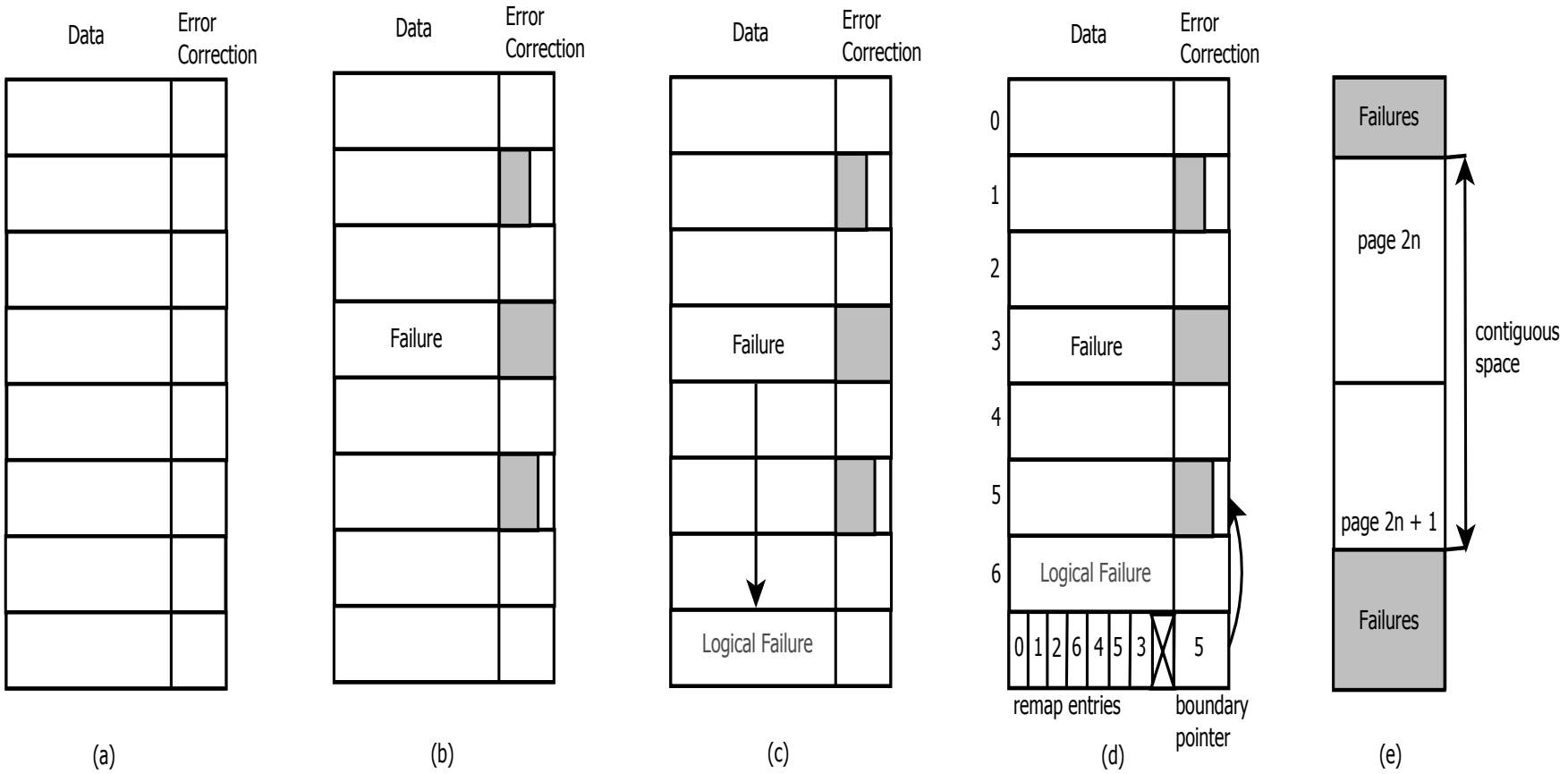


# Influence of failures

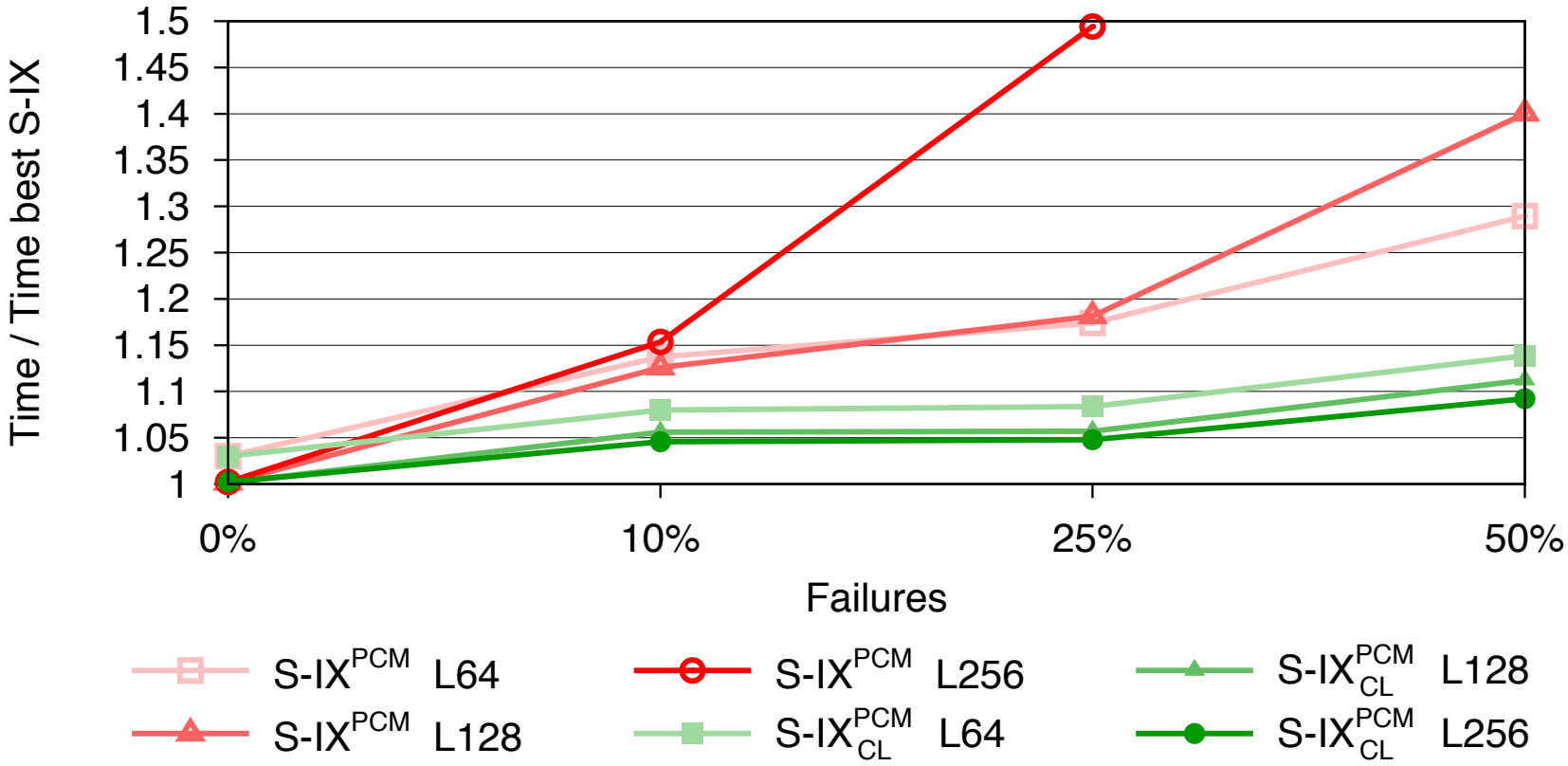
- Failure Cluster



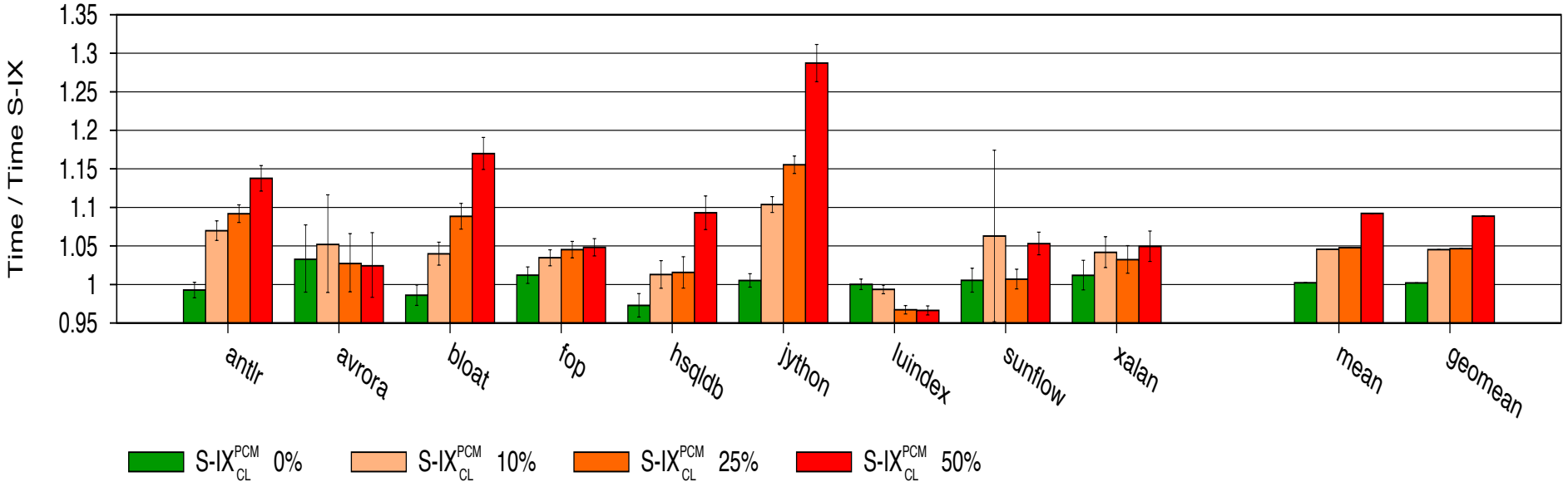
# Failure Cluster Hardware Mechanism



# Performance with failure cluster



# Performance with failure cluster



# Conclusions

- A cooperative hardware/software system with low complexity that mitigates failures in wearable memories
- Wear leveling can be detrimental when errors have to be exposed to software
- Failure-aware managed runtime can benefit from wear-unleveling

# Thank you