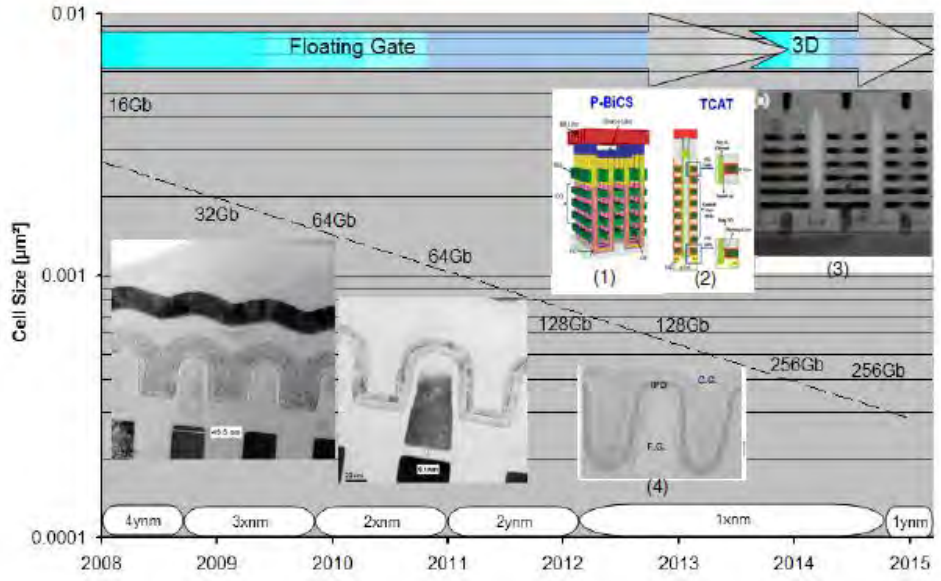
Exploiting Latency Variation for Access Conflict Reduction of NAND Flash Memory

Jinhua Cui, Weiguo Wu, Xingjun Zhang, Jianhang Huang, Yinfeng Wang*
Xi'an Jiaotong University, *ShenZhen Institute of Information Technology

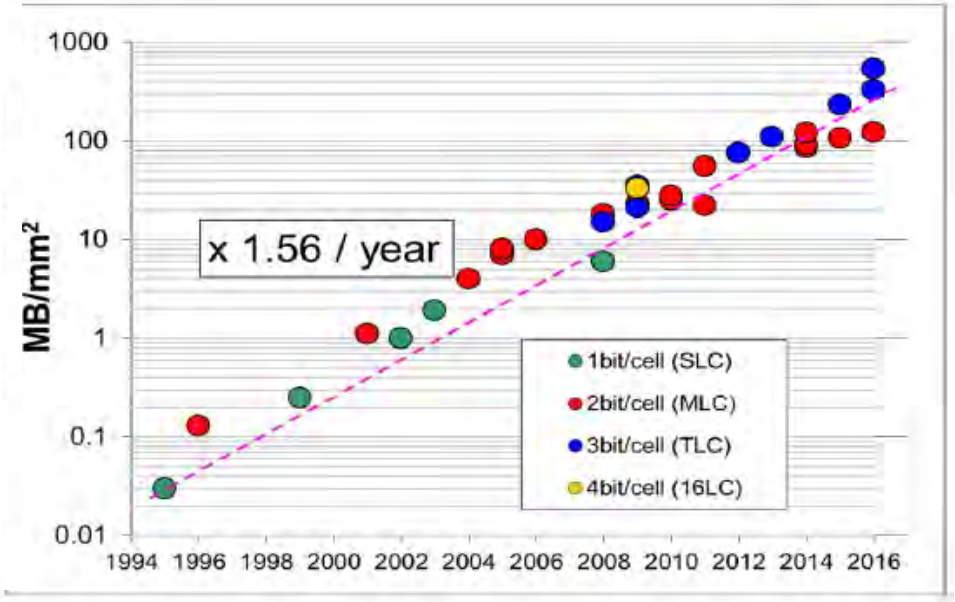
OUTLINE

1. Background and Motivation
2. Design of RHIO
3. Evaluations
4. Conclusions

NAND Flash Memory



Flash Cell Size Trends
Source: Flash Memory Summit



NAND Flash Memory Trends
Source: ISSCC'16 Tech. Trends

Cell size↓

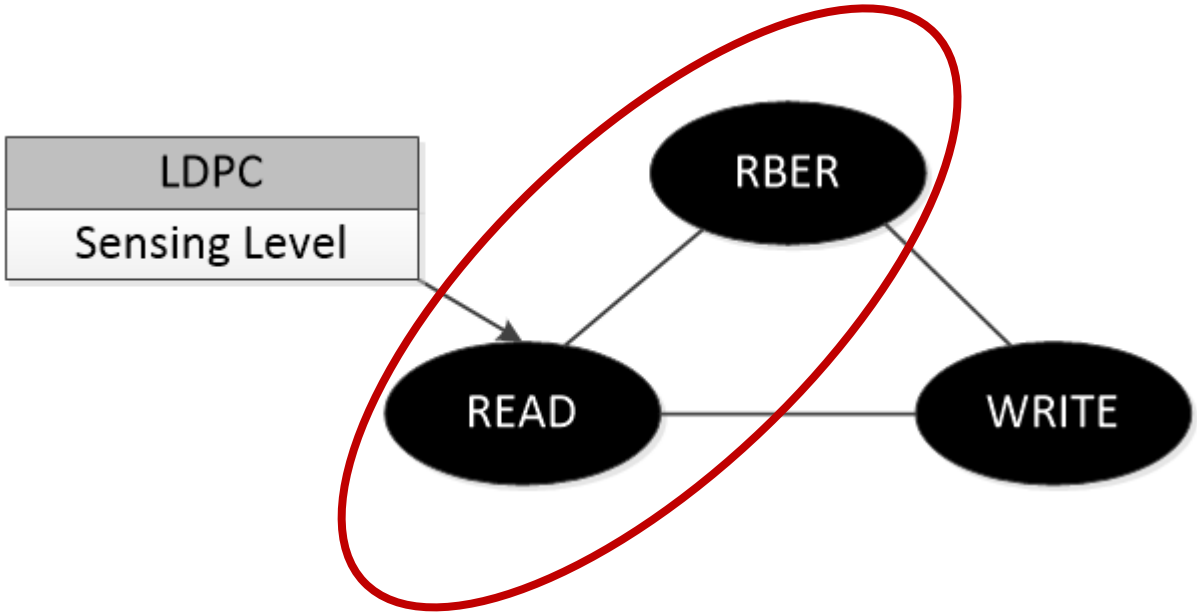
Bit/cell↑

→ RBER ↑ → Write ↓ | Read ↓

Tradeoff ☺

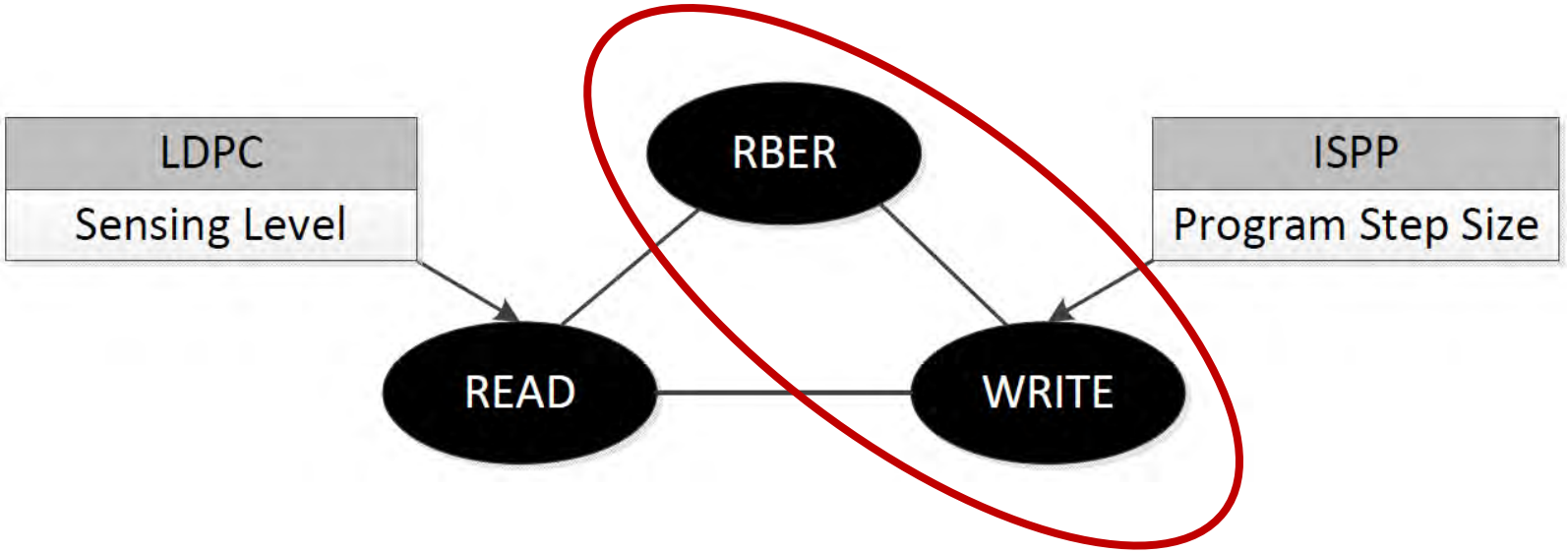
Tradeoff: RBER, Write, Read (1/4)

- ECC complexity, ECC capability and read speed
  - Soft-decision memory sensing
  - Sensing levels ↑ → preciser memory sensing (stronger ECC capability)
  - Sensing levels ↓ → less reference voltage (faster read)



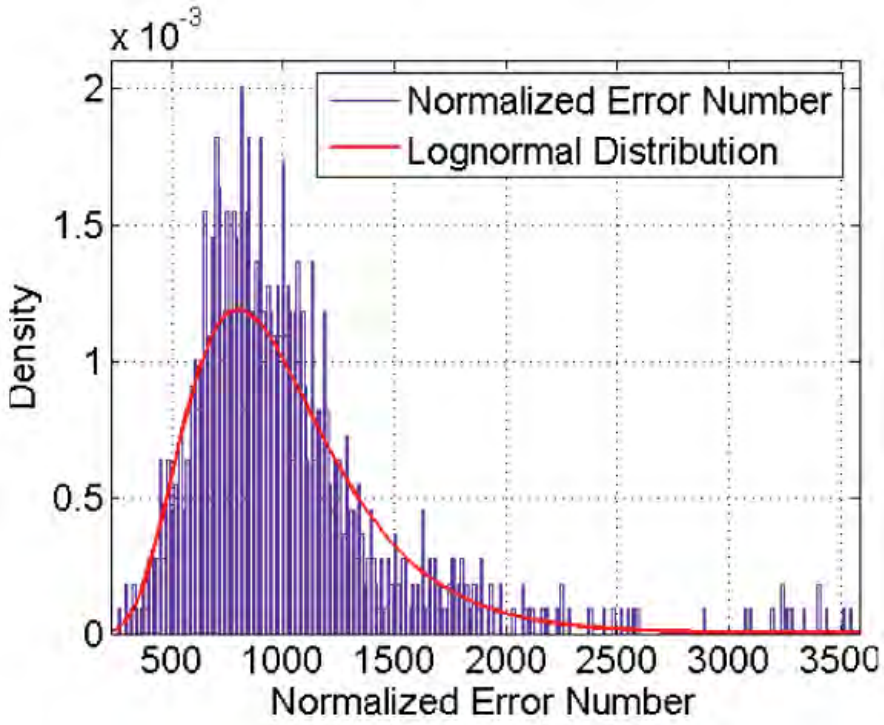
Tradeoff: RBER, Write, Read (2/4)

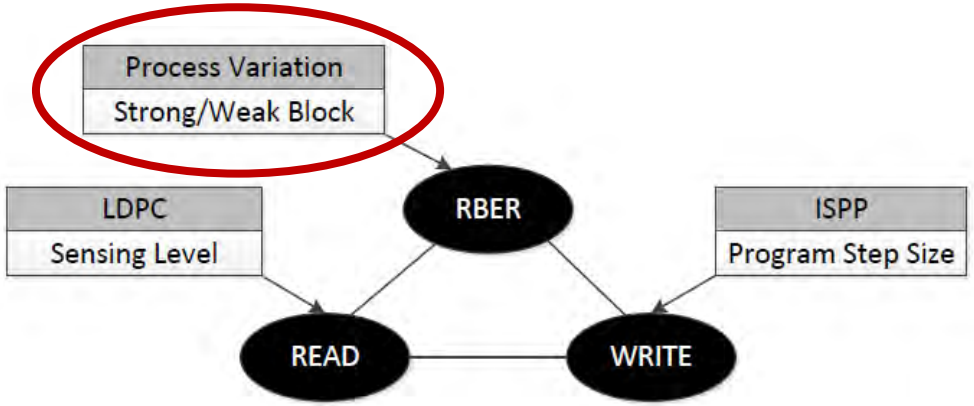
- RBER, program step size and write speed
  - Incremental step pulse programming (ISPP)
  - $\Delta V_p \uparrow \rightarrow$ fewer steps (faster write)
  - $\Delta V_p \downarrow \rightarrow$ preciser control on V_{th} (lower RBER)



Tradeoff: RBER, Write, Read (3/4)

- Process Variation (PV)
  - Different worst-case RBER under the same P/E cycling
  - Strong block ↑ → lower RBER ↓
  - Weak block ↓ → higher RBER ↑

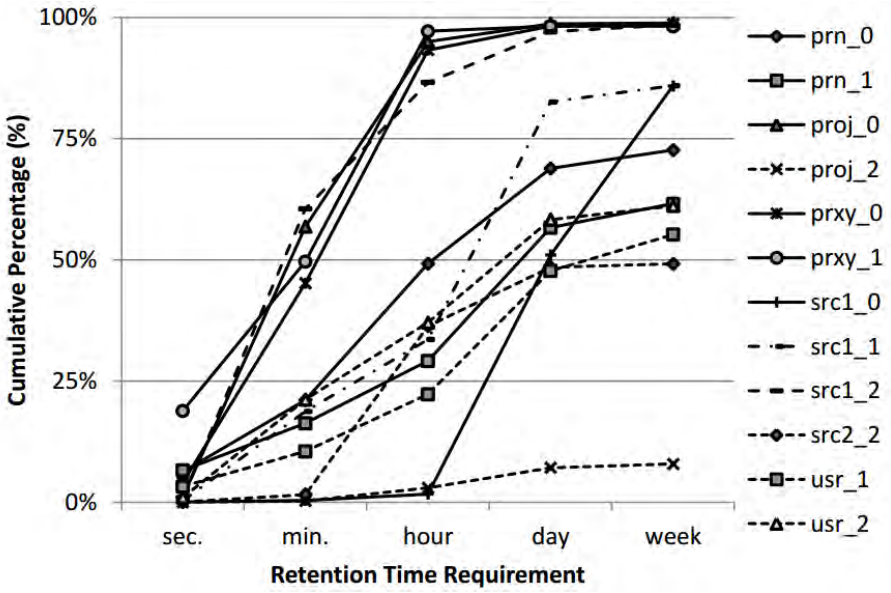


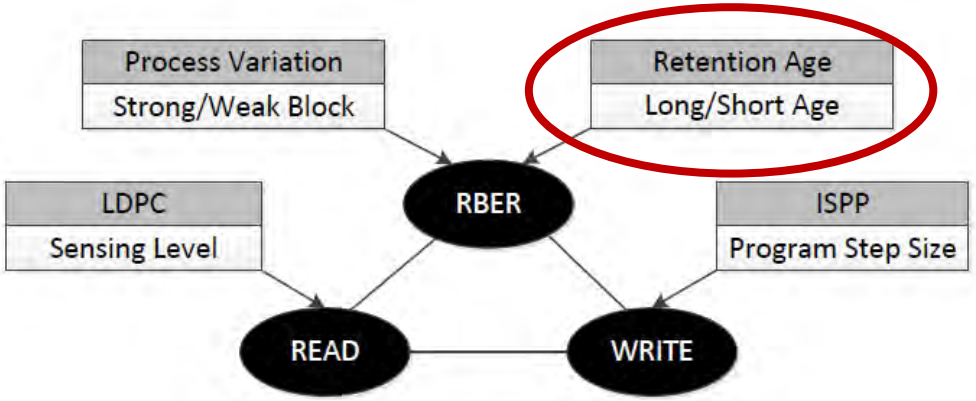


Source: Pan et al, "Error Rate-BasedWear-Leveling for NAND Flash Memory at Highly Scaled Technology Nodes"

Tradeoff: RBER, Write, Read (4/4)

- Retention Age Variation
  - The length of time since a flash cell was programmed
  - Short age↑ → lower RBER ↓
  - Long age↓ → higher RBER ↑



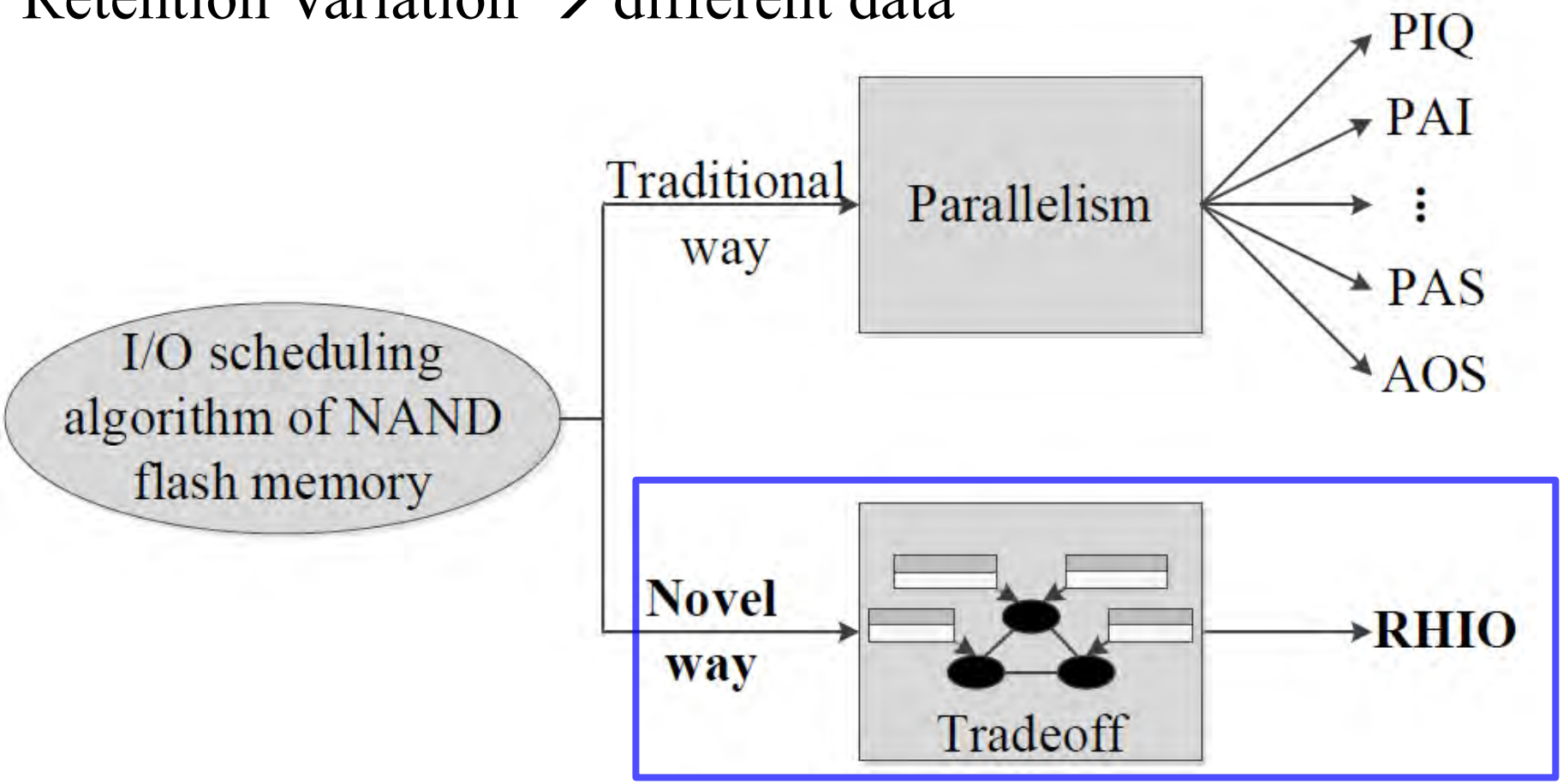


Source: Liu et al, "Optimizing NAND Flash-Based SSDs via Retention Relaxation", Fast 2012

Motivation

- Process Variation → different blocks
- Retention Variation → different data

Speed variation



Our work is focused on here

西安交通大学
XI'AN JIAOTONG UNIVERSITY

Design of RHIO

02

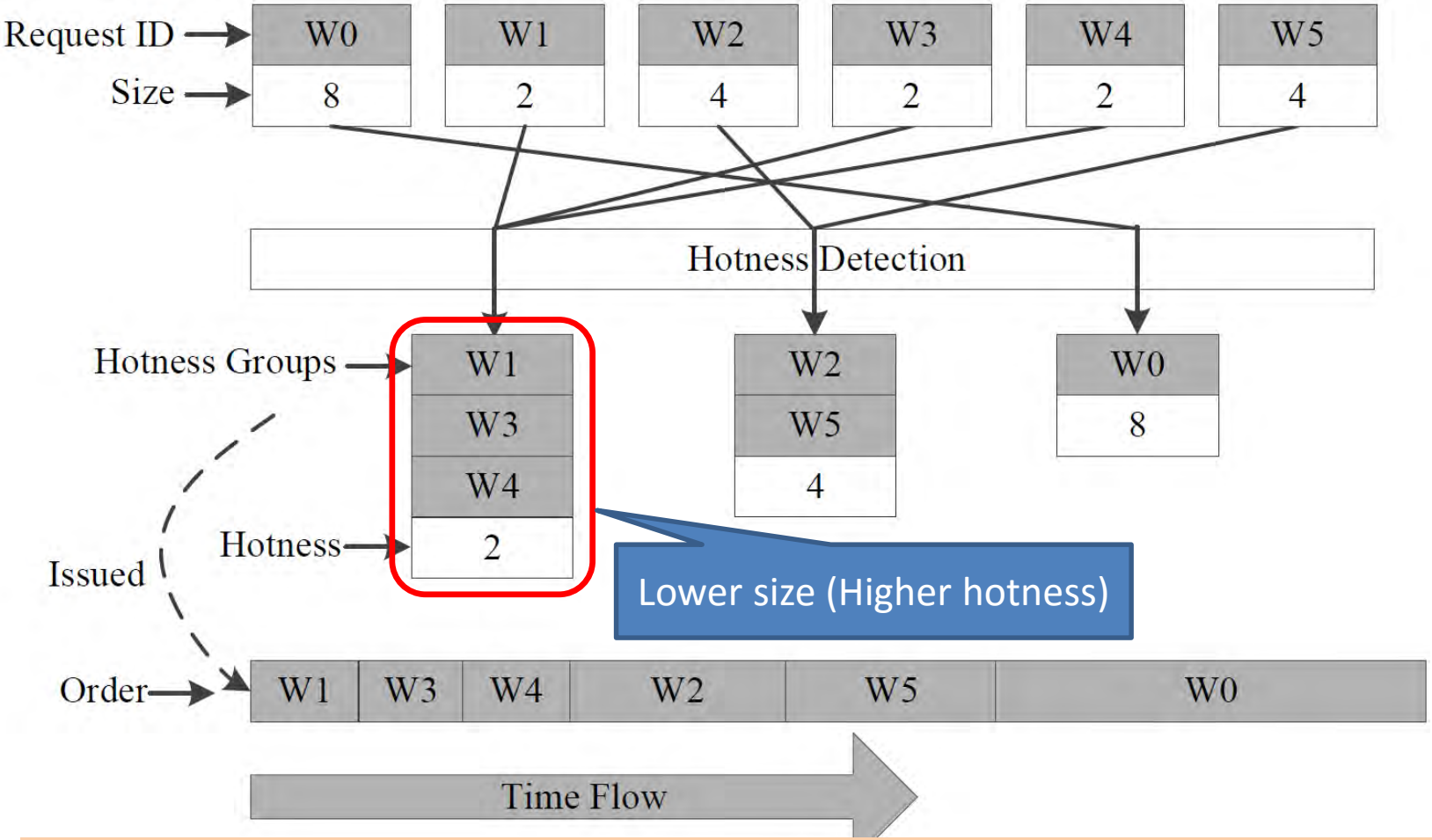
Main idea of RHIO

- Observation
  - If a tradeoff-aware technique improves I/O performance based on the variation characteristic of an attribute, the detection of the attribute can be implemented in **I/O scheduling** and thus the tradeoff induced **speed variation** can be exploited for maximal benefit by **giving scheduling priority to fast writes and fast reads.**
- Techniques
  - Process variation based fast write
  - Retention age based fast read
  - Shortest-job-first scheduling

Hotness-aware Write Scheduling

- Put **hot data** in strong blocks using **fast write**, and **non-hot** data into normal blocks with **normal** writes
- Give scheduling priority to **hot write requests** to reduce the conflict latency of next few requests in the queue
- Use the size of IO requests to identify hotness

Hotness-aware Write Scheduling

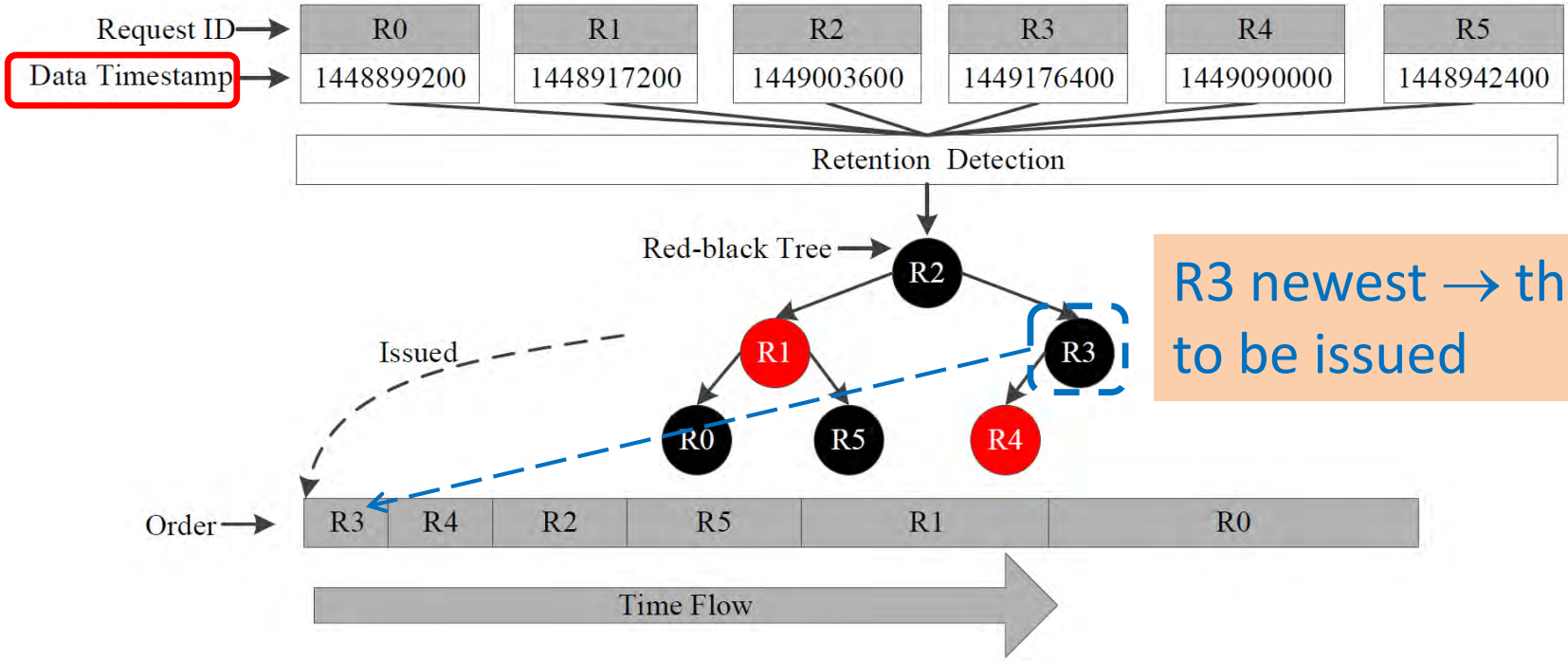


- Read-write separation
- Hotness Groups are issued in the order of hotness

Retention-aware Read Scheduling

- Perform **fast read** (less sensing levels) on the data with **low retention ages**
- Give scheduling priority to **reads accessing data with low retention ages** to reduce the conflict latency of next few requests in the queue
- **Retention age identification** by extending each mapping entry in the FTL with a timestamp field and recording the timestamp when data is programmed

Retention-aware Read Scheduling



R3 newest → the first to be issued

- Write: size-based predicted hotness
  Read: retention-based actual hotness
- Write: discrete size → hotness groups
  Read: consecutive retention age → red-black tree
- Deadline → FIFO queue
  SATA interface → PRIO: 01b, ICC: deadline value

西安交通大学
XI'AN JIAOTONG UNIVERSITY

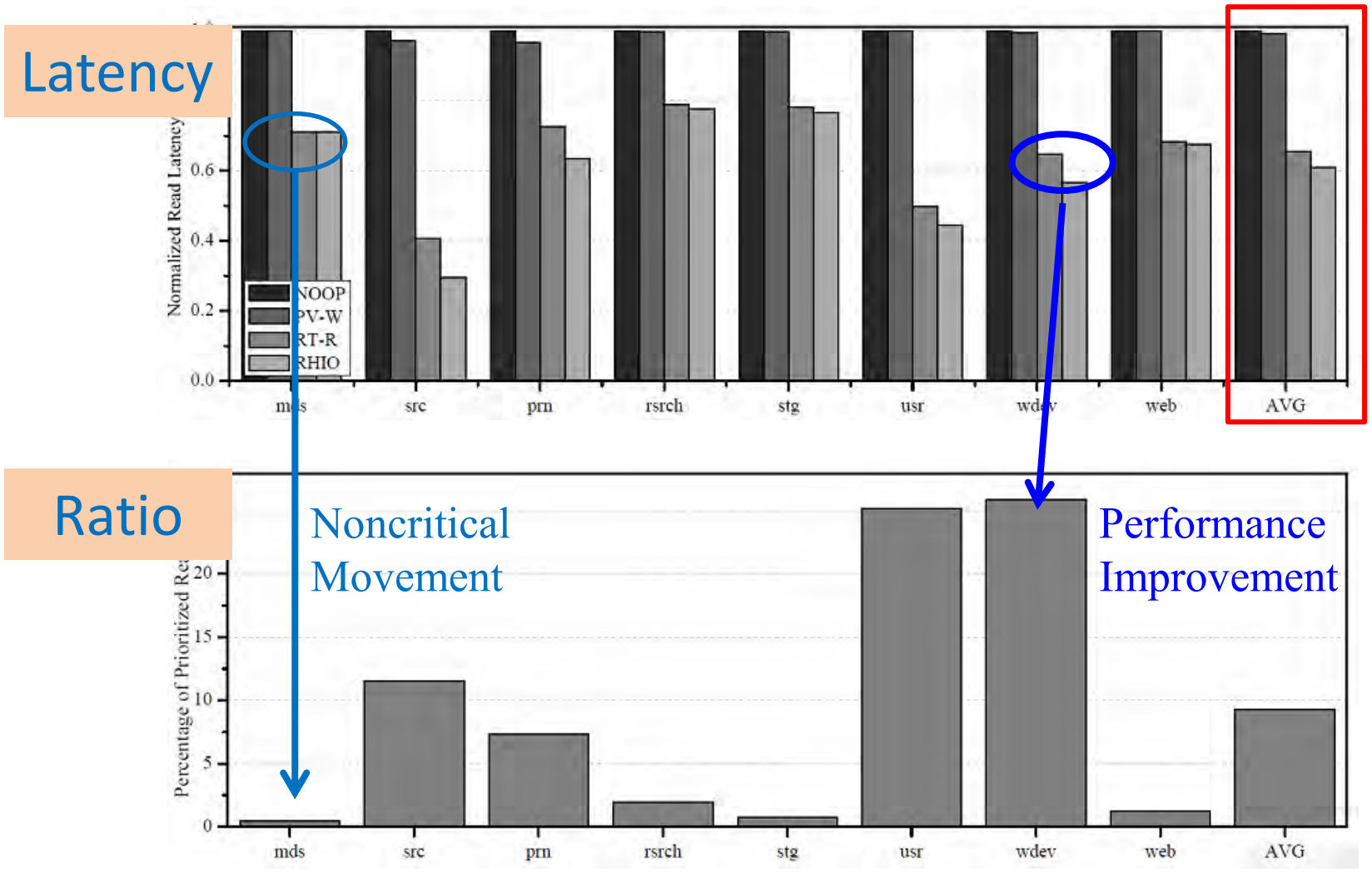
Evaluations

03

Evaluation and Discussions

- A trace-driven simulator is used to verify the proposed algorithm.
- Traces include a set of selected MSR Cambridge traces from SNIA.
- Comparison among: NOOP, PV-W, RH-R, RHIO.
  - NOOP: Traditional I/O Scheduler
  - PV-W: PV-aware write performance improvement without conflict-aware reordering.
  - RH-R: Retention-aware read performance improvement without reordering I/O requests sequence.
  - RHIO: Our proposed I/O scheduler.

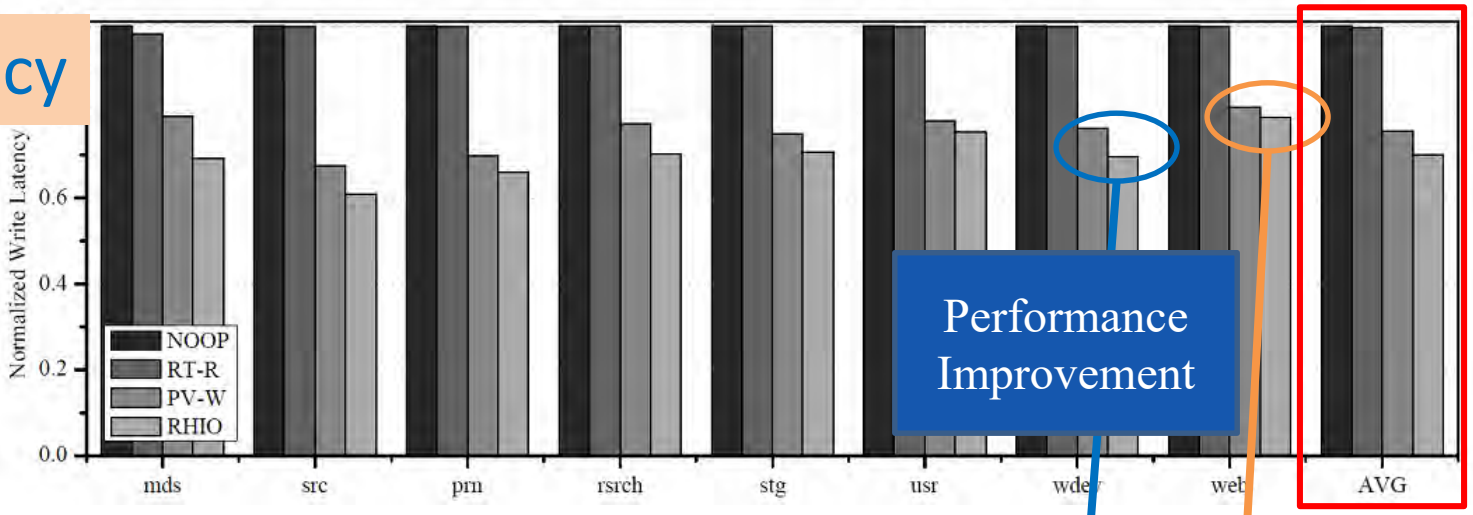
Read Performance



- RHIO vs. NOOP: 39.11% ↑
- RHIO vs. RT-R: 7.04% ↑

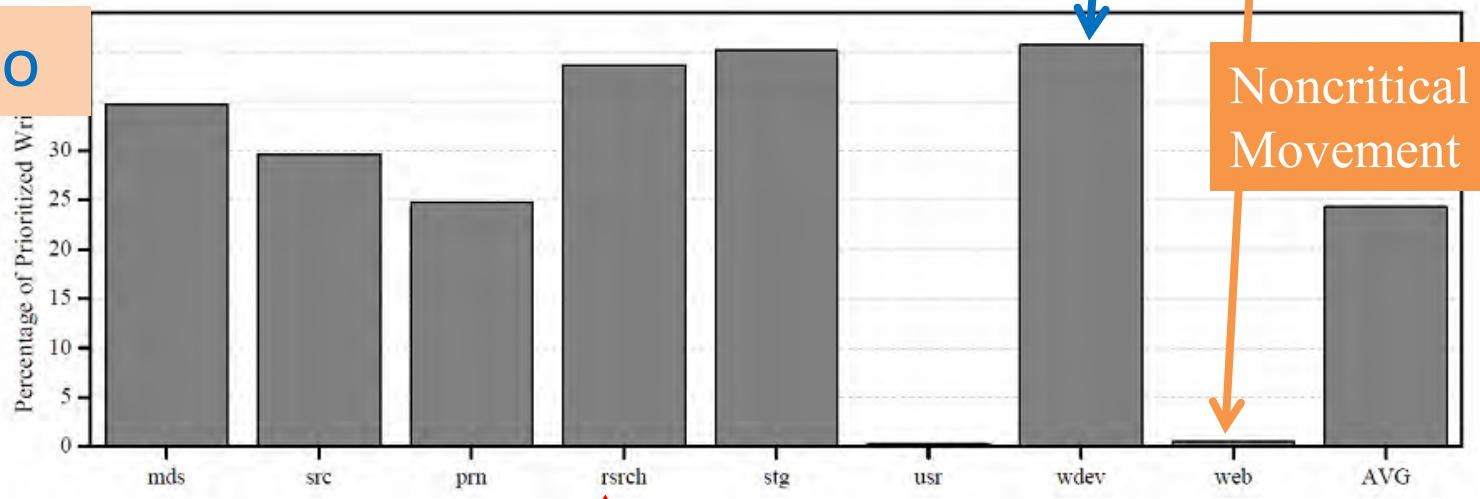
Write Performance

Latency



Performance Improvement

Ratio



Noncritical Movement

- RHIO vs. NOOP: 29.92% ↑
- RHIO vs. PV-W: 7.12% ↑

西安交通大学
XI'AN JIAOTONG UNIVERSITY

04

Conclusions

Conclusions

- Proposed an I/O scheduler (RHIO) to exploit latency variation for access conflict reduction of NAND flash memory.
  - *Hotness-aware write scheduling: give scheduling priority to hot write requests and allocate their data to strong blocks with fast write.*
  - *Retention-aware read scheduling: give scheduling priority to read requests which access data with low retention ages using fast read.*

- Experimental results show that the proposed approach is very efficient in performance improvement.

THANKS FOR YOUR ATTENTIONS !

QUESTIONS?

cjhnicole@gmail.com