ROSS:
A Design of Read-Oriented STT-MRAM Storage for Energy-Efficient Non-Uniform Cache Architecture

Jie Zhang, Miryeong Kwon, Changyoung Park, Myoungsoo Jung, Songkuk Kim

Computer Architecture and Memory System Lab,

Yonsei University

Summary

Motivation:

There is a huge demand to increase the last-level cache size. However, SRAM suffers from the **high leakage power** and **low density**.

STT-MRAM can be a good candidate to replace SRAM, due to its small cell size and low leakage power.

Challenge:

Conventional STT-MRAM *cannot directly substitute* SRAM, because of

- *High write latency*
- *High write energy*

Solution:

We observe read-oriented data pattern in real applications;

We propose ***ROSS,*** a read-oriented STT-MRAM storage that can

- dynamically detect the read-oriented data pattern
- *reduce the write frequency in STT-MRAM*
- *accommodate as many read misses as possible in LLC*

Motivation

Data set size increases explosively.

Webs

1 Billion Web Pages

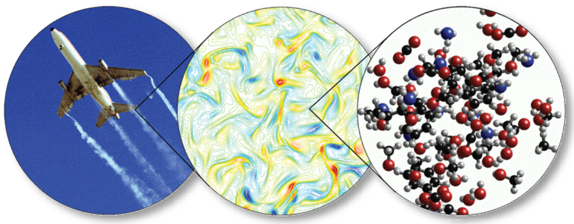
Videos

300-hour new videos/min

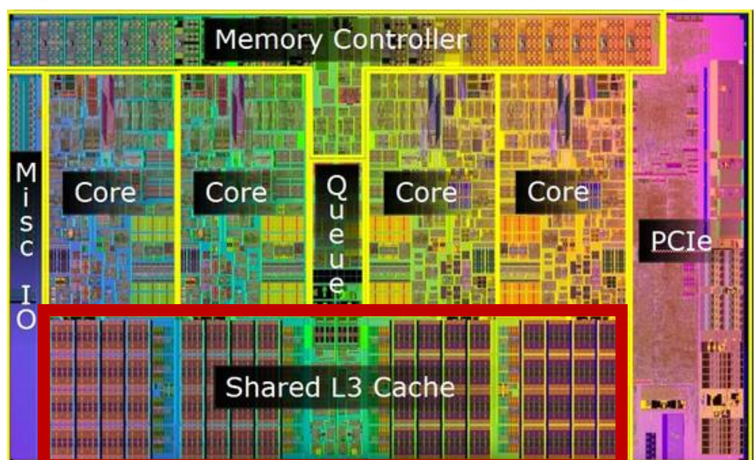
Graphs

30 Billion Photos

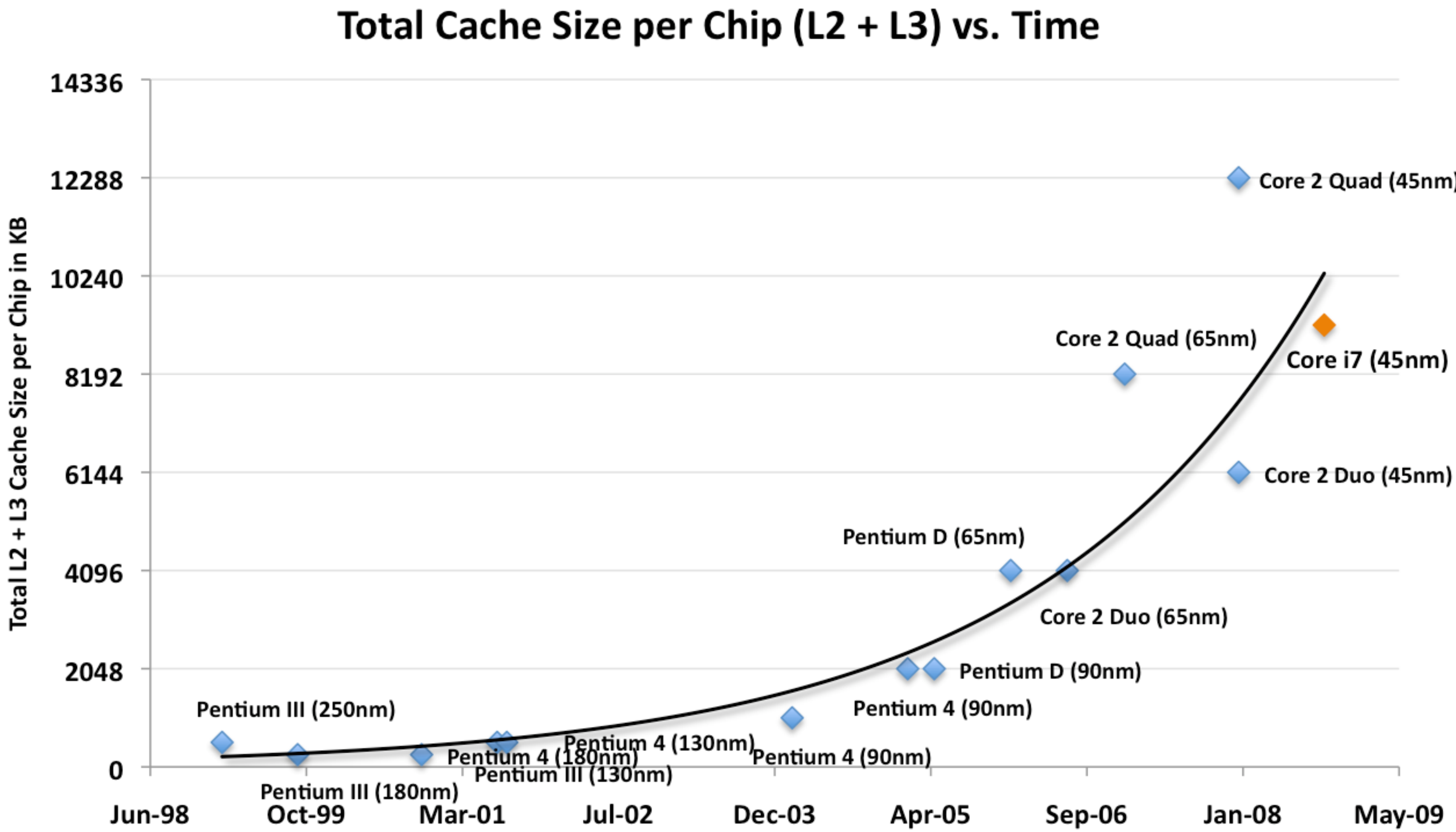
Scientific Computing



10-month Spacecraft Simulation



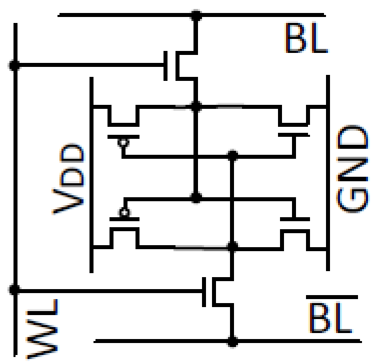
Motivation



Huge demand to increase on-chip last-level cache size in future

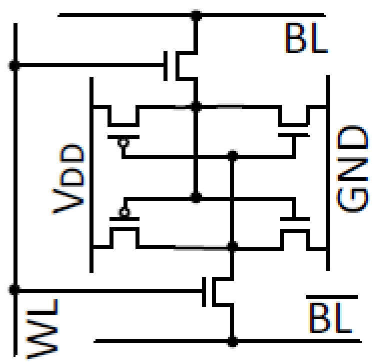
Challenges

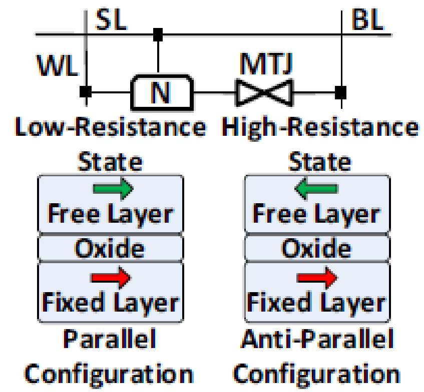
Traditional SRAM



Read Latency	397ps	Fast speed
Write Latency	397ps	
Read Energy	35pJ	Reasonable energy
Write Energy	35pJ	
Cell Area (F^2)	50~120	Low density
Leakage	75.7mW	High Leakage

Challenges





	Traditional SRAM	STT-MRAM
Read Latency	**397ps**	**238ps**
Write Latency	**397ps**	**6932ps**
Read Energy	**35pJ**	**13pJ**
Write Energy	**35pJ**	**90pJ**
Cell Area (F^2)	**50~120**	**6~40**
Leakage	**75.7mW**	**6.6mW**

High Write Latency/Energy

High density

Low leakage

Prior work

Read Preemption: suspend write operation and give priority to read operation

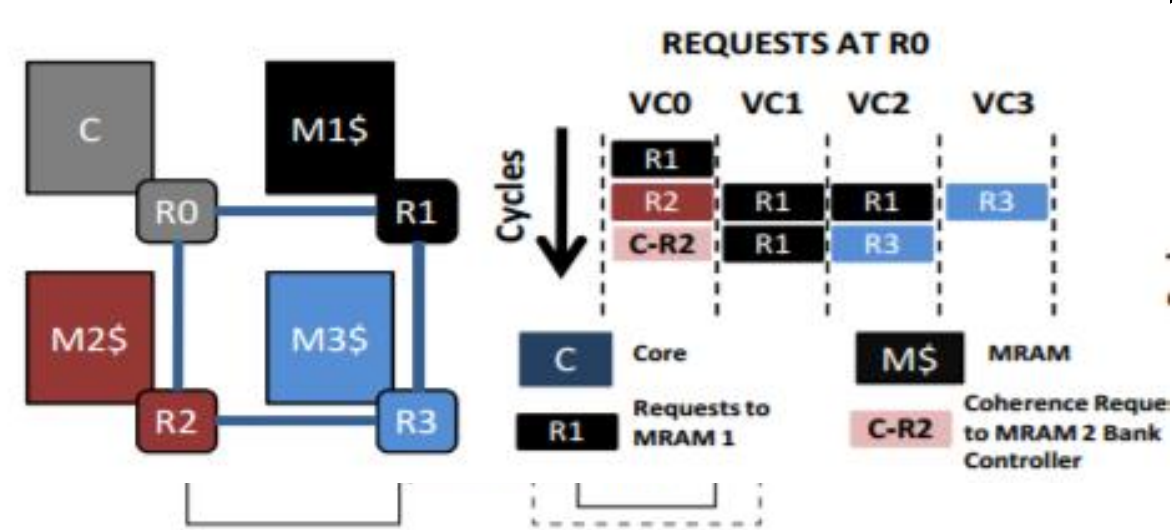
- Mitigate the impact of long write latency to read access;
- Cannot reduce the write latency

Write Request Rerouting: accommodate write requests with idle blocks

- Parallelize write requests to hide long write latency;
- Cannot reduce the write energy;

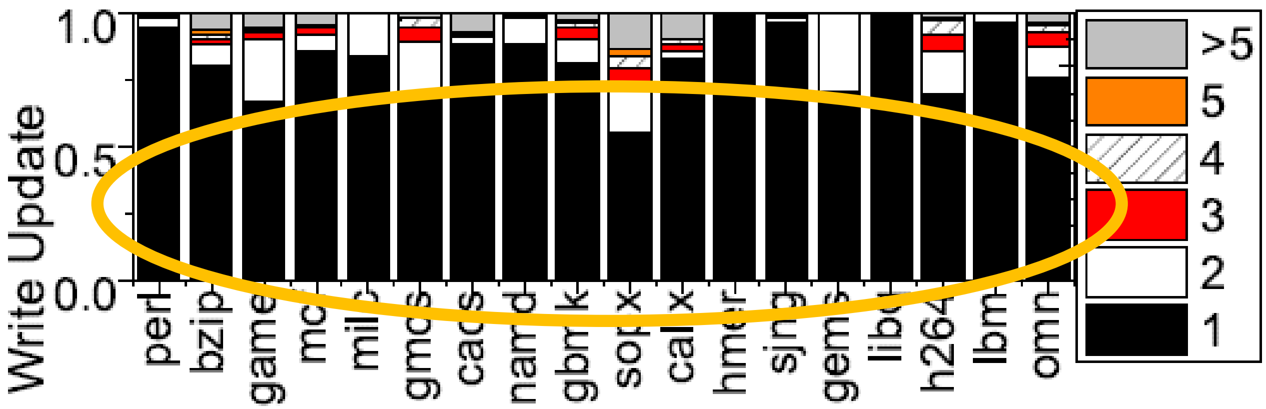
Deadwrite bypass: predict and bypass the “deadwrite” (single write no read).

- Effective
- Bypassir on STT-MRAM



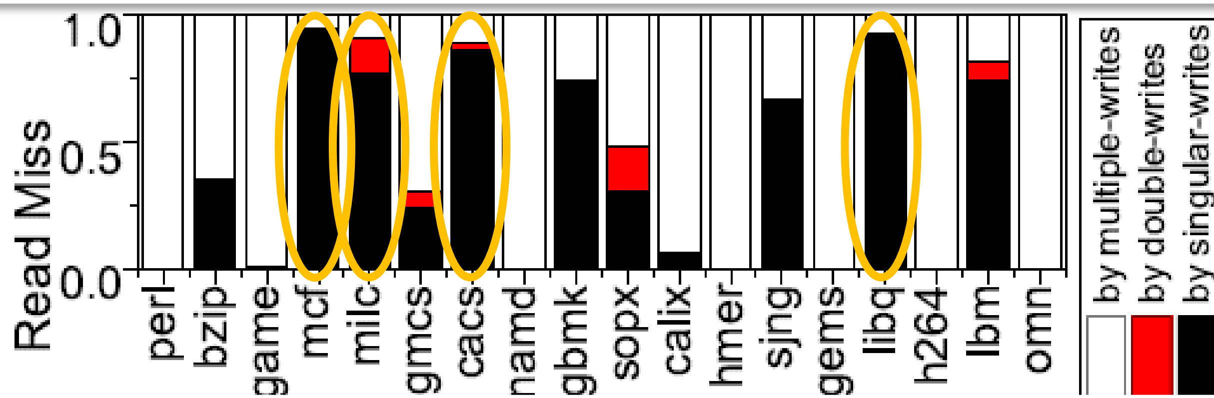
ROSS-key observation

How the application behavior can impact LLC data access?



Last Level Cache eviction trend

Large working sets result in frequent cache eviction



What if we can detect the singular write blocks and put in STT-MRAM?

ROSS

Target: build a hybrid last-level cache to

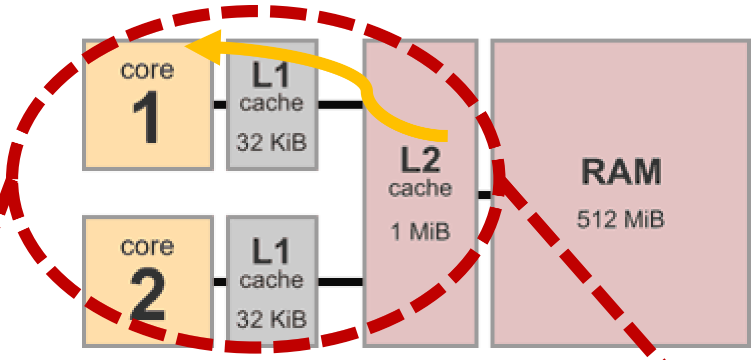
i. avoid many cache read misses with larger STT-MRAM;

ii. accommodate write updates in SRAM;

iii. intelligent data migration algorithm;

Hybrid-NUCA

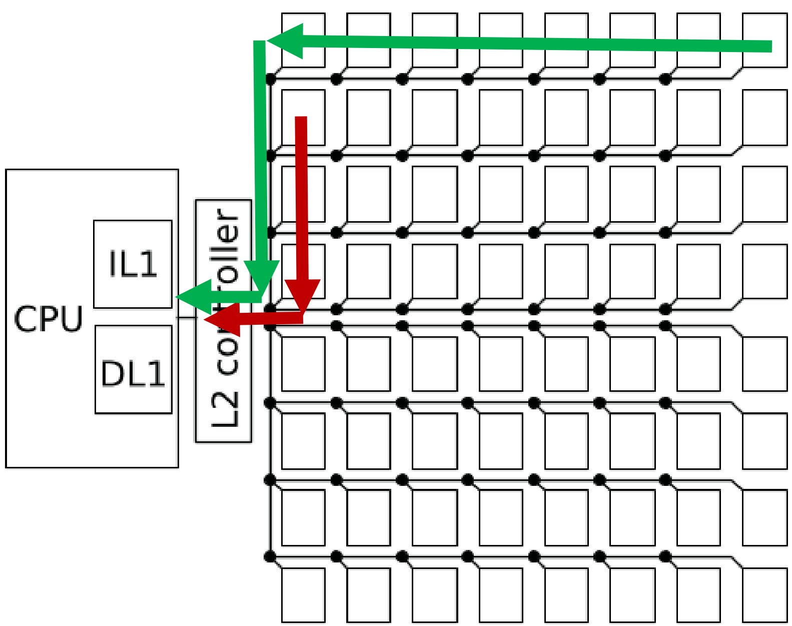
Conventional cache hierarchy



Uniform latency

Severe wire delay

Non-uniform cache architecture

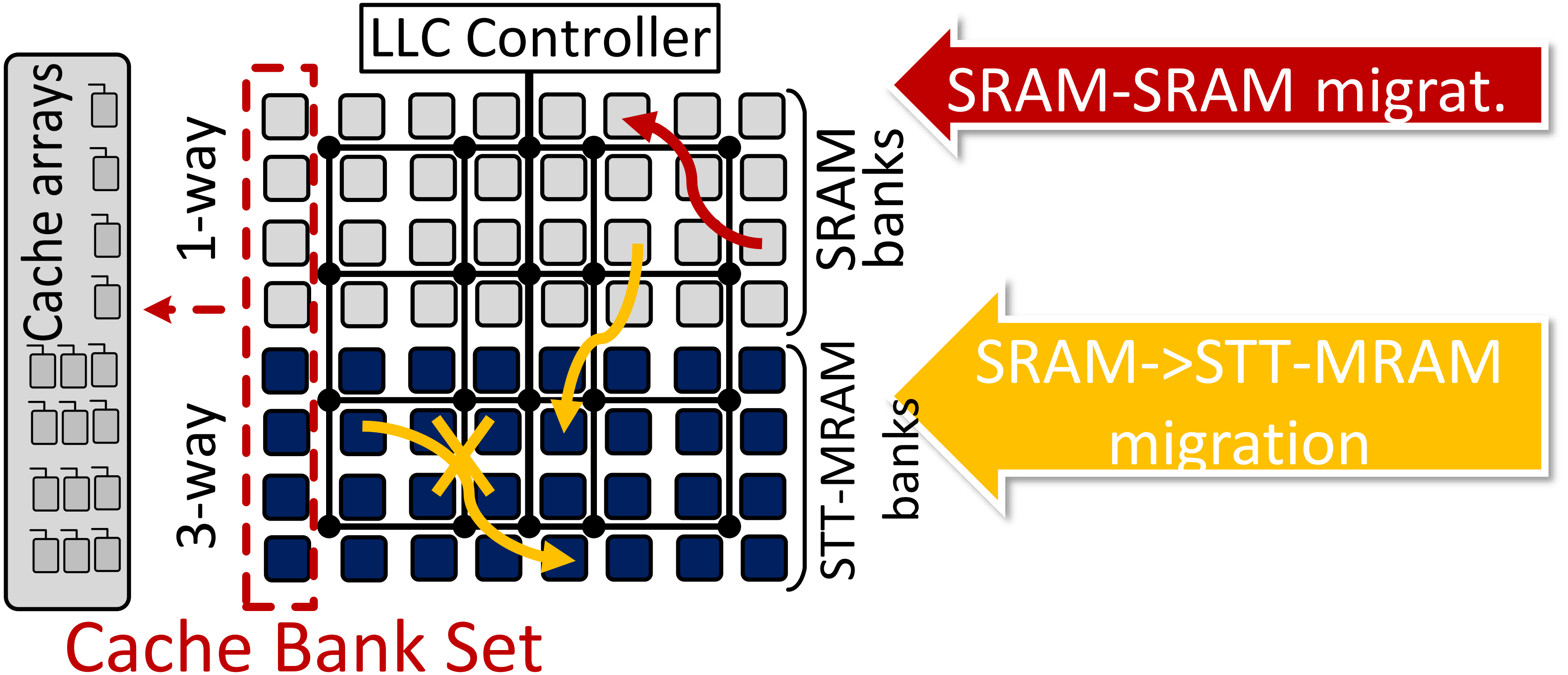


Different wire latency

Hybrid-NUCA

Migration policy

Address map policy



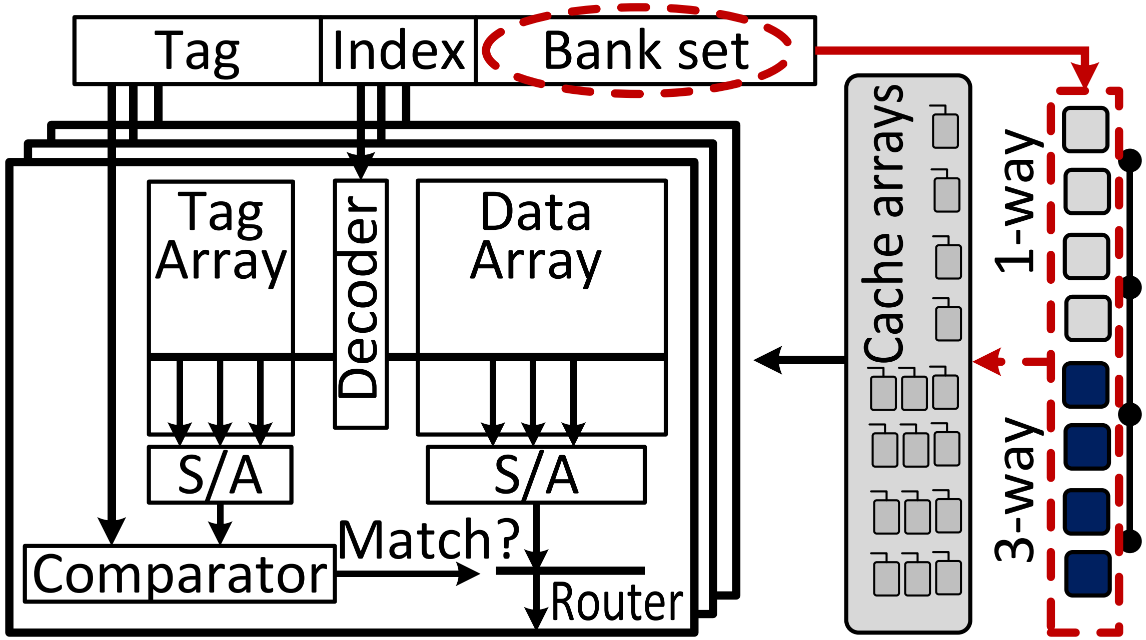
Less flexibility, but simplified design

Hybrid-NUCA

Migration policy

Address map policy

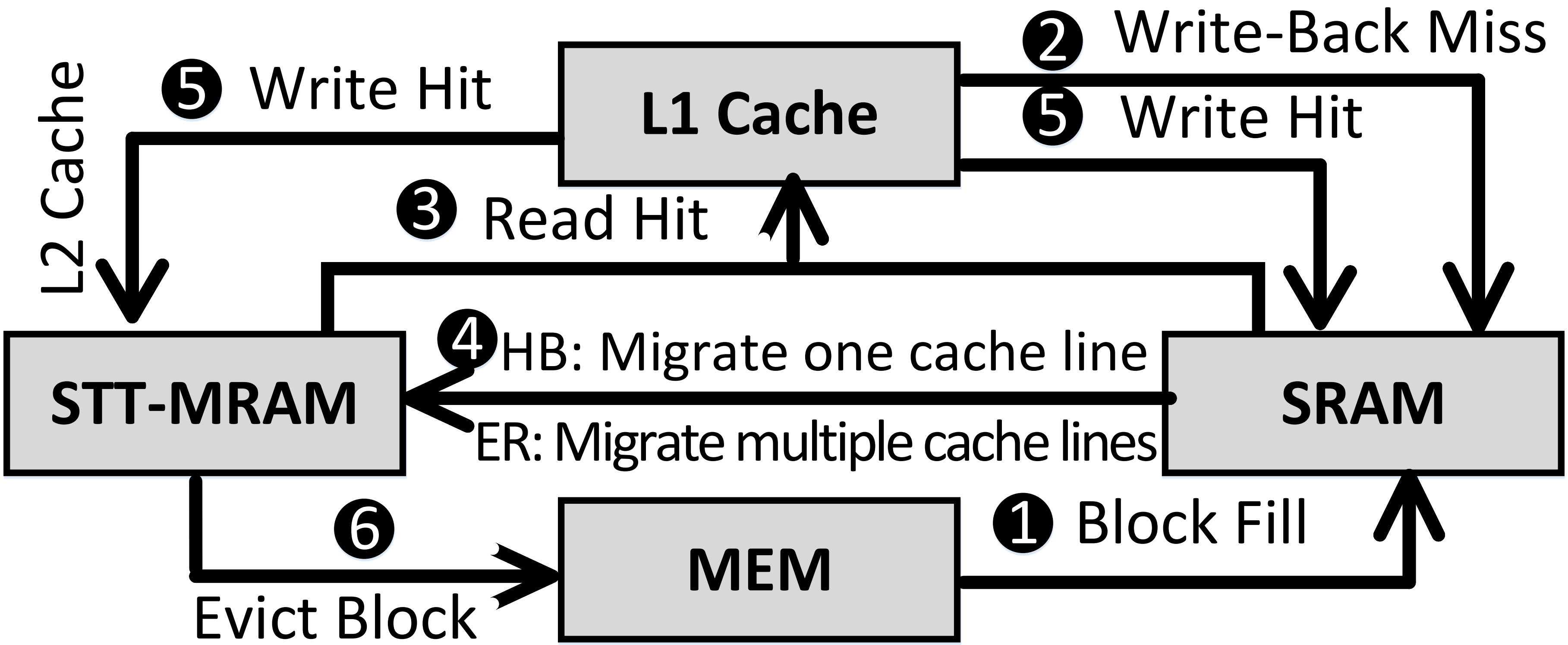
Data search policy



Request routes to specific cache bank set

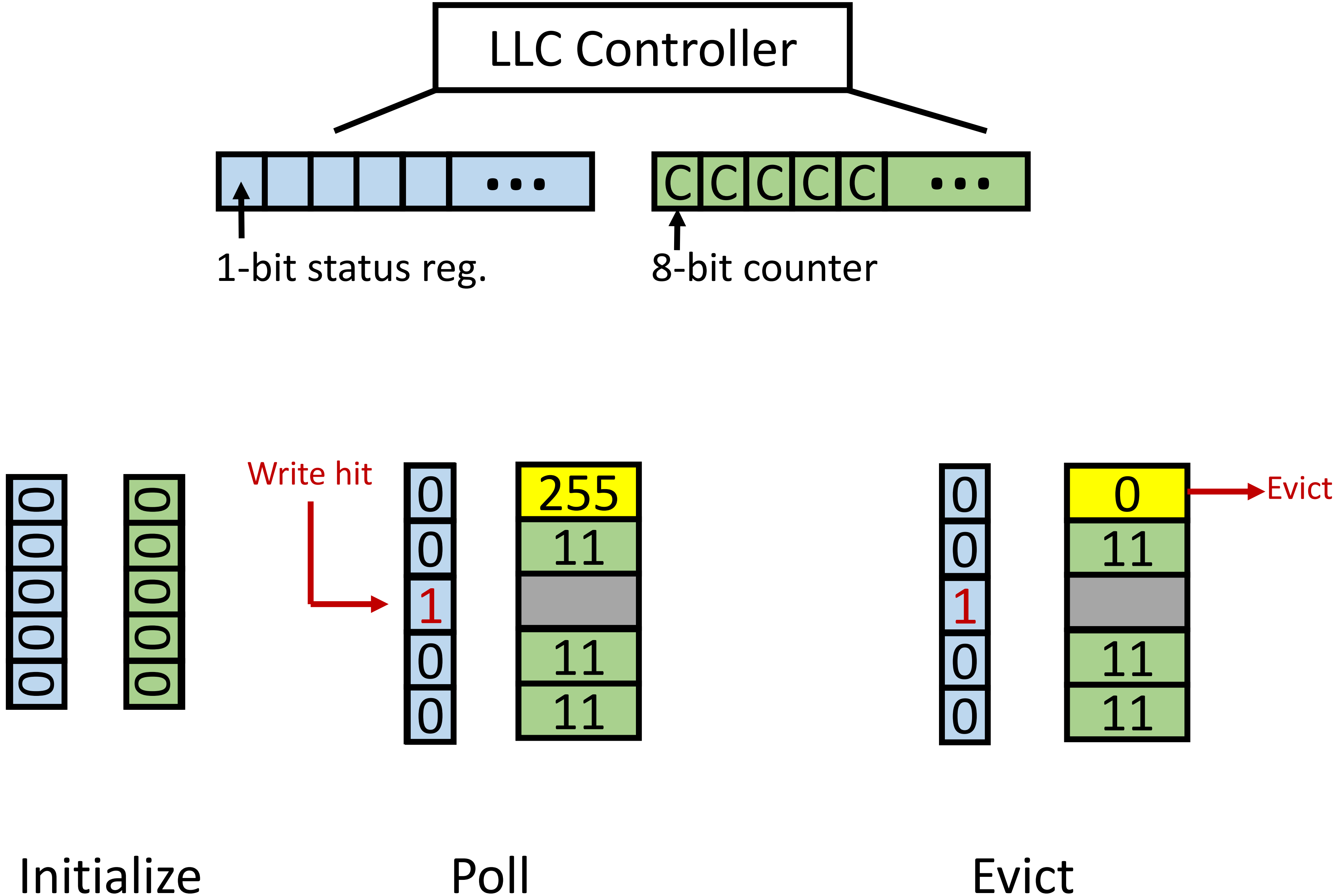
Tag broadcast to each bank for searching

Data Flow



L2 Cache
❺ Write Hit
L1 Cache
❷ Write-Back Miss
❺ Write Hit
❸ Read Hit
❹ HB: Migrate one cache line
STT-MRAM
SRAM
ER: Migrate multiple cache lines
❻
MEM
❶ Block Fill
Evict Block

Singular write detection



Experiment Setup

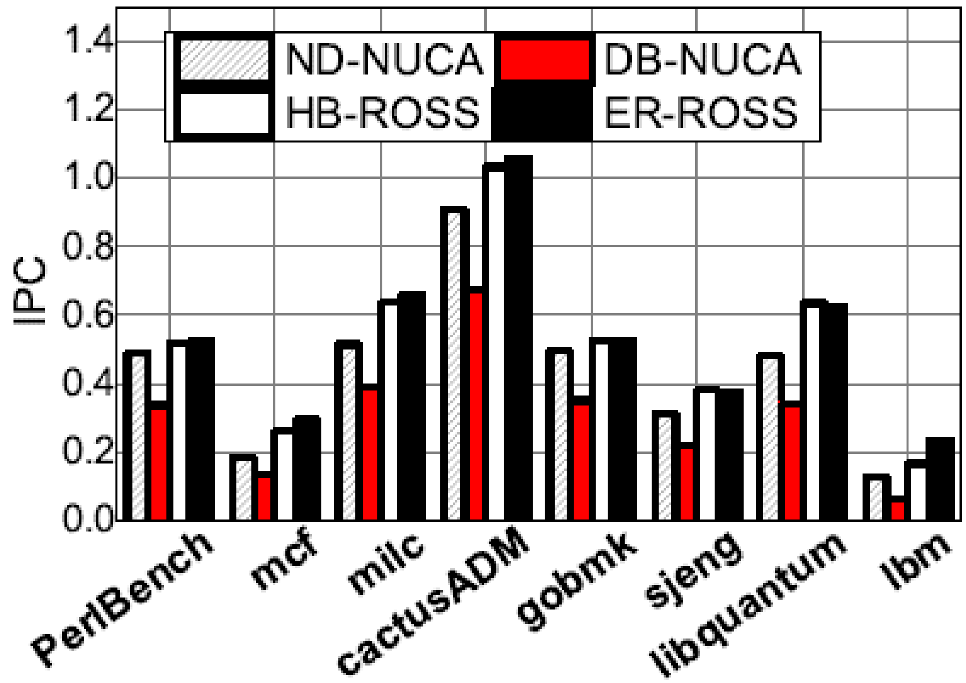
Evaluation model:

- ND-NUCA: traditional SRAM NUCA model;
- DB-NUCA: STT-MRAM NUCA model aware of deadblock;
- HB-ROSS: Hybrid Read-Oriented STT-MRAM Storage;
- ER-ROSS: Early Retirement aware Raed-Oriented STT-MRAM Storage;

Gem5 simulation parameters and workloads:

Processor	1 core, OoO execution, SE mode
Frequency	3.2GHz
L1 Cache	16KB/core, 2-way, 2 cycles
SRAM L2	32KB/bank, R/W : 20 cycles
STT-MRAM L2	96KB/bank, R/W : 20/60 cycles
Network	Wormhole Switching, 2 cycles
Off-chip Mem	4GB DDR3 DRAM FR-FCFS
Benchmark	SPEC2006
Workloads	PerlBench, mcf, milc, libquantum, lbm, cactusADM, sjeng, and gobmk

Evaluation

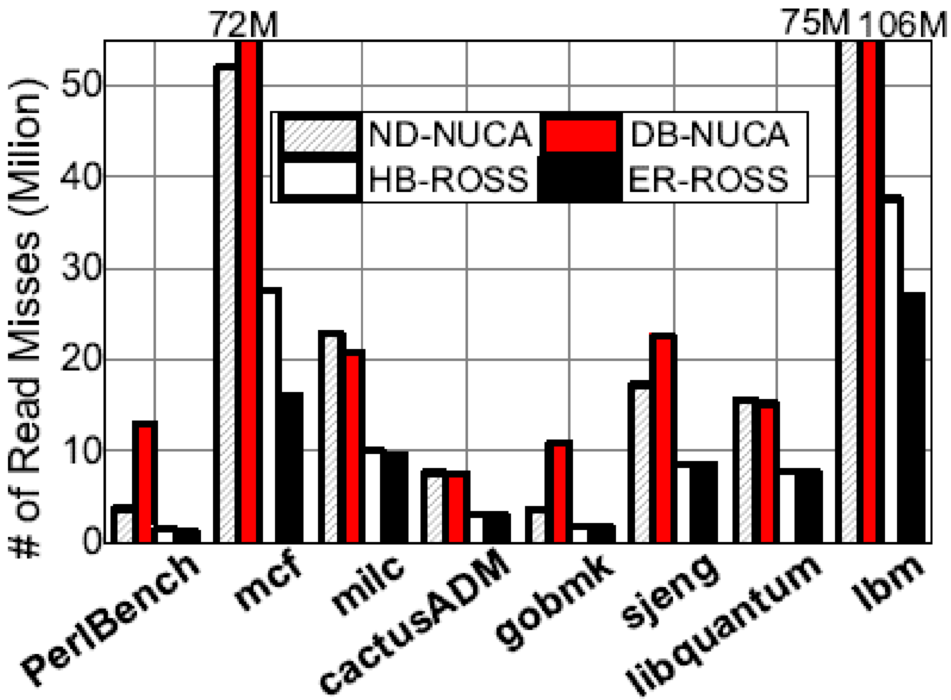


DB-NUCA: suffer from long write latency

HB-ROSS: larger capacity and less STT-MRAM write

ER-ROSS: benefits from early retirement

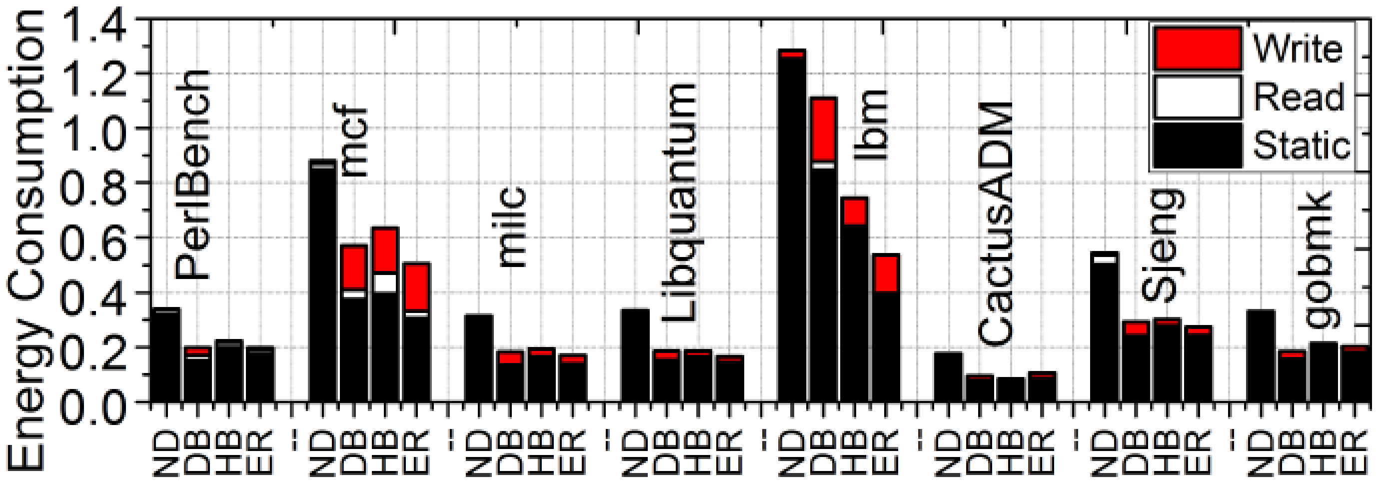
Evaluation



DB-NUCA: suffer from evicting singular write blocks

ROSS: STT-MRAM accommodates read-intensive data

Evaluation



ND-NUCA: consume more leakage energy

ROSS: reduce the write energy overhead in STT-MRAM

Thank You