# A Coherent Hybrid SRAM and STT-RAM L1 Cache Architecture for Shared Memory Multicores

Jianxing Wang, Yenni Tim
Zhong-Liang Ong and Weng-Fai Wong
School of Computing
National University of Singapore

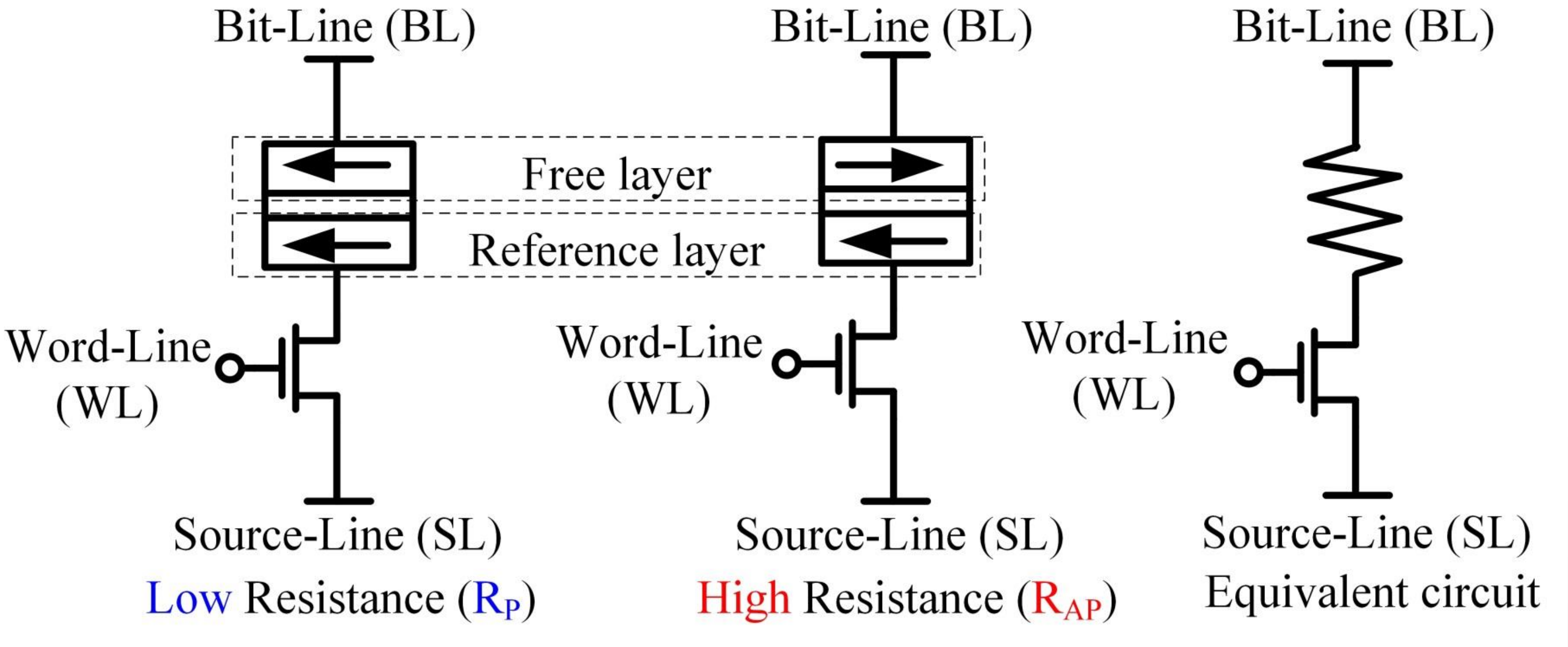
Zhenyu Sun and Hai (Helen) Li
Swanson School of Engineering
University of Pittsburgh

# Outline

- STT-RAM Basics
  - Cell structure and advantages
  - Challenges and motivations
- Hybrid L1 Cache Architecture
  - Naïve solution
  - The MESI protocol
  - Block transfer mechanisms
- Evaluation
  - Performance and energy
  - STT-RAM endurance
- Conclusion

# STT-RAM Basics

- Magnetic Tunnel Junction (MTJ)
  - Two ferromagnetic layers separated by a barrier



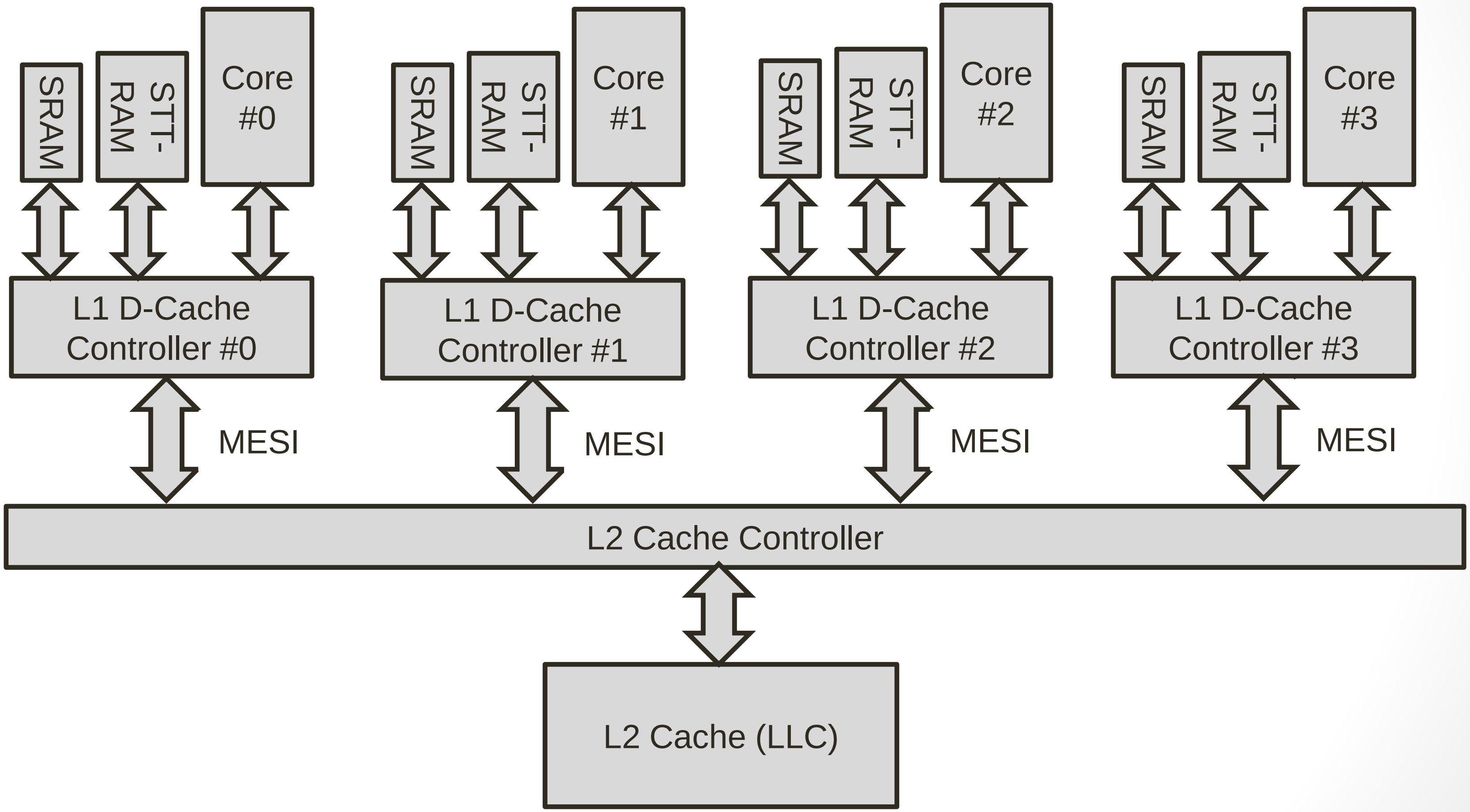
# STT-RAM Basics (cont.)

- Advantages
  - Non-volatile, near zero leakage energy
  - As fast as SRAM (read)
  - As dense as DRAM
  - Multi-level cell capability (stacking MTJs)
  - CMOS-compatible
  - Universal memory

# Motivations of Hybrid Cache

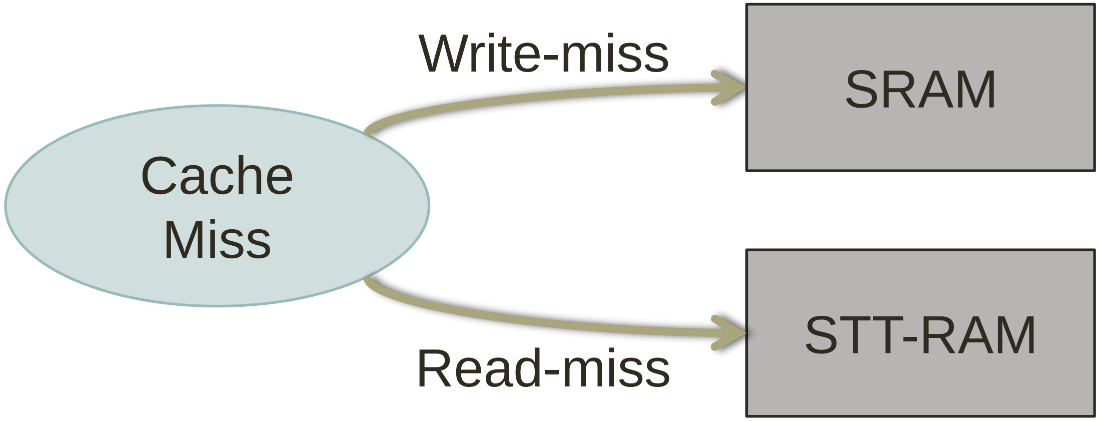
- Expensive write operation of STT-RAM
  - High latency (10ns+)
  - High energy
  - Compensated by relaxed non-volatility [*Smullen et al. 11*]
    - Refresh
  - Endurance
- Intense writes in L1
  - *bodytrack*: L1(s) / L2 = ~29!
  - Additional synchronous operations under multi-core environment

# Proposed Hybrid Cache Hierarchy



# Cache Block Management

- Naïve solution
  - Based on temporal locality



  - Simple but not good enough
    - > 3% IPC degradation

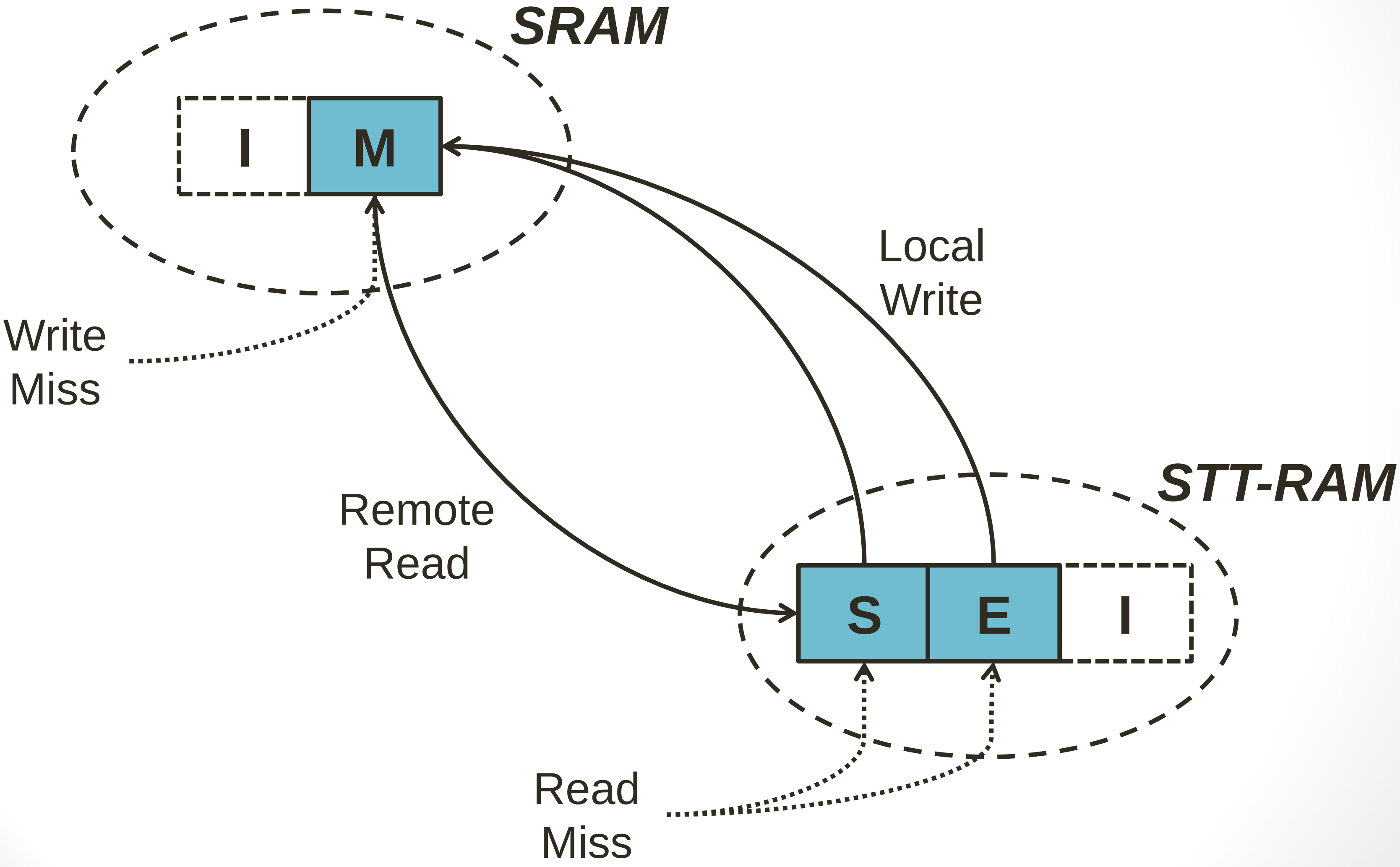
# The MESI Coherent Protocol

- Developed by University of Illinois
  - Illinois MESI
- For each cache block
  - **M** (modified) state – data dirty, exclusive copy
  - **E** (exclusive) state – data clean, exclusive copy
  - **S** (shared) state – data clean, multiple copies
  - **I** (invalid) state
- Common event bus
  - Local (processor) read/write
  - Remote (snoop / bus) read/write

# Cache Block Management (cont.)

- Immediate transfer policy (IT)
  - Place dirty data (**M** state) block in SRAM
  - Place clean data (**E/S** state) block in STT-RAM
  - Transfer cache block when coherent state changes
  - DO NOT need extra information (built-in by MESI)

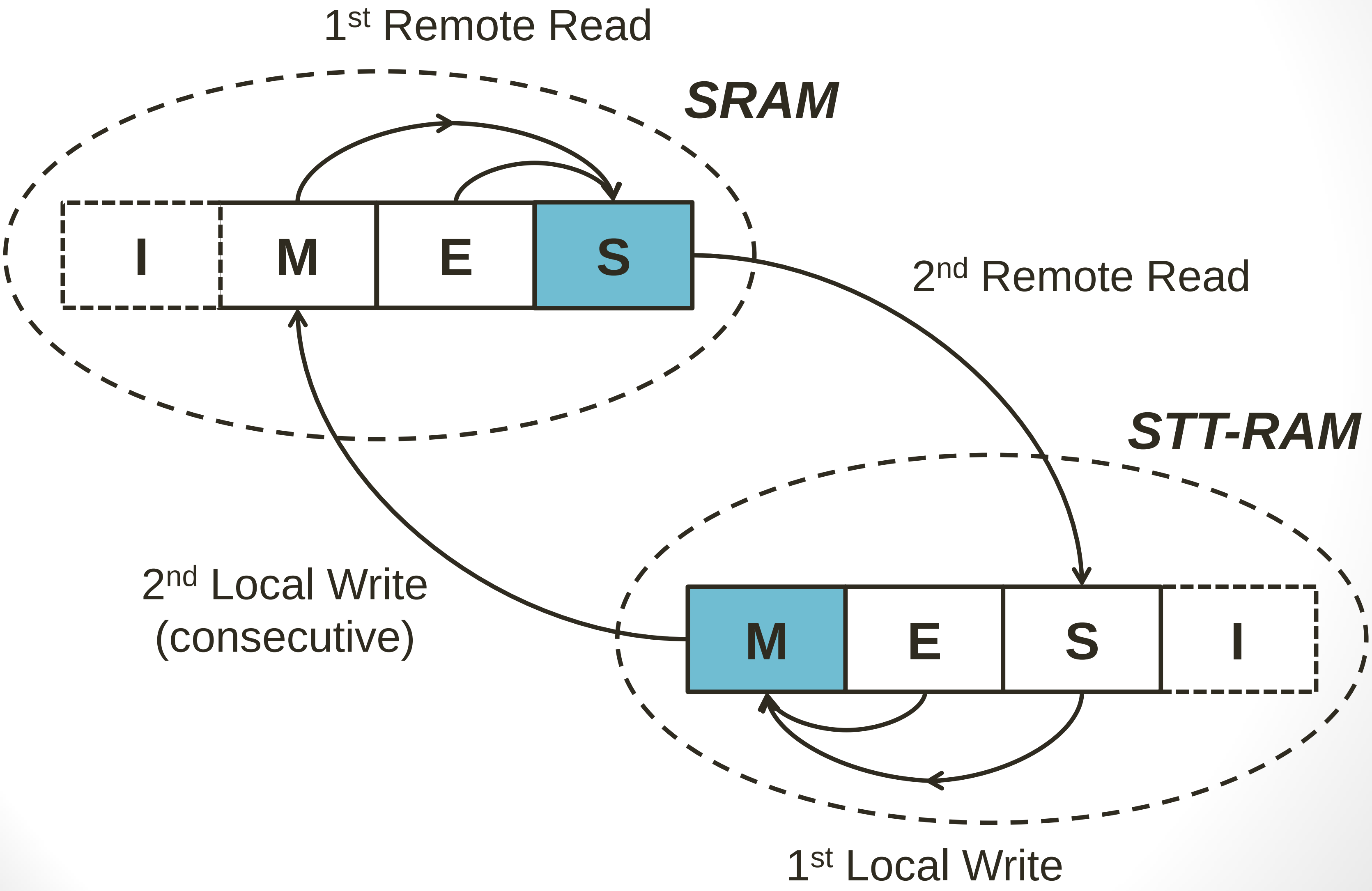
# Immediate Transfer Policy (IT)



# Cache Block Management (cont.)

- Delayed transfer policy (DT)
  - IT could be too aggressive
    - Coherent state “ping-pong” between M and S
  - Relax state restriction
  - Consider request history in prediction
  - Extra information required

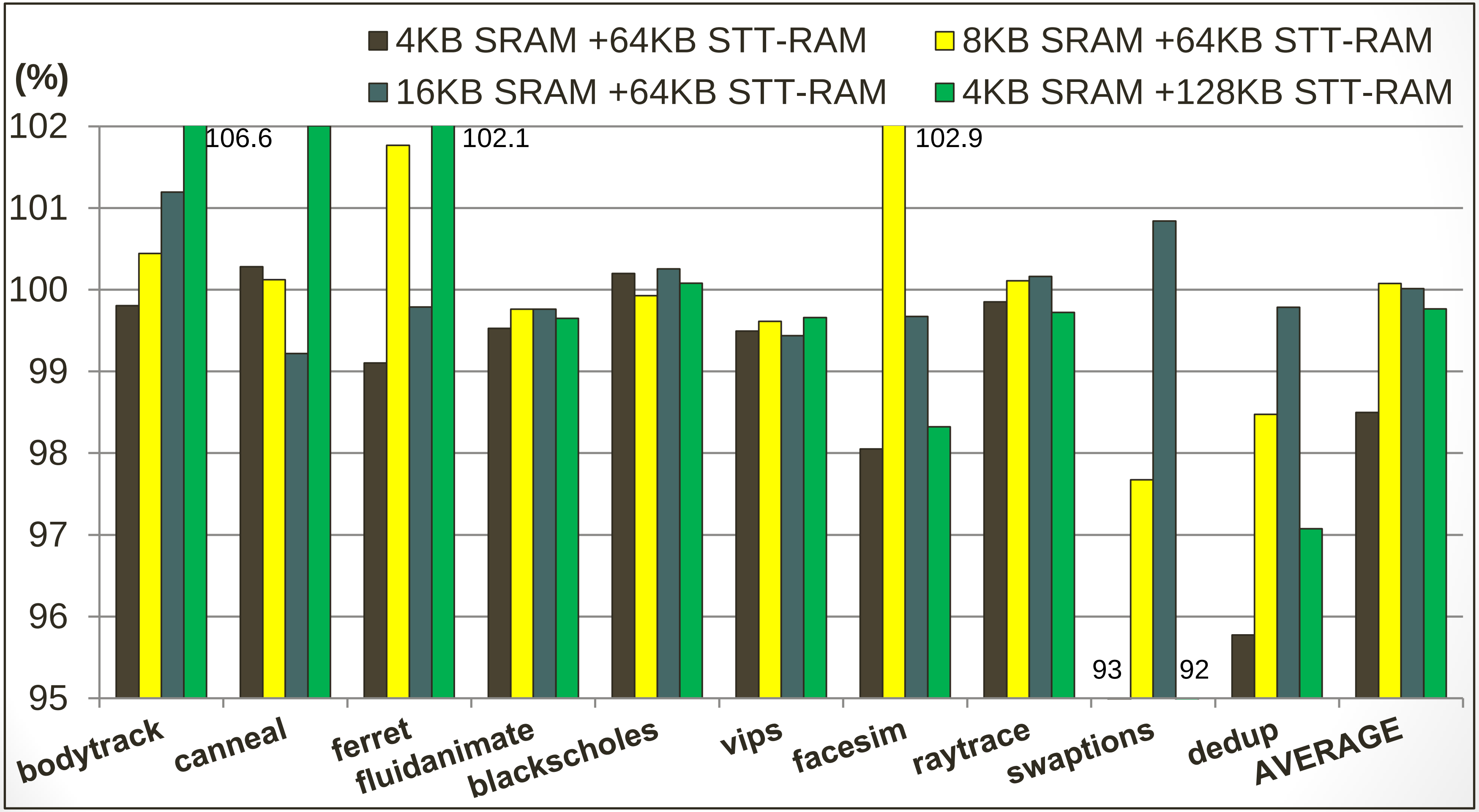
# Delayed Transfer Policy (DT)



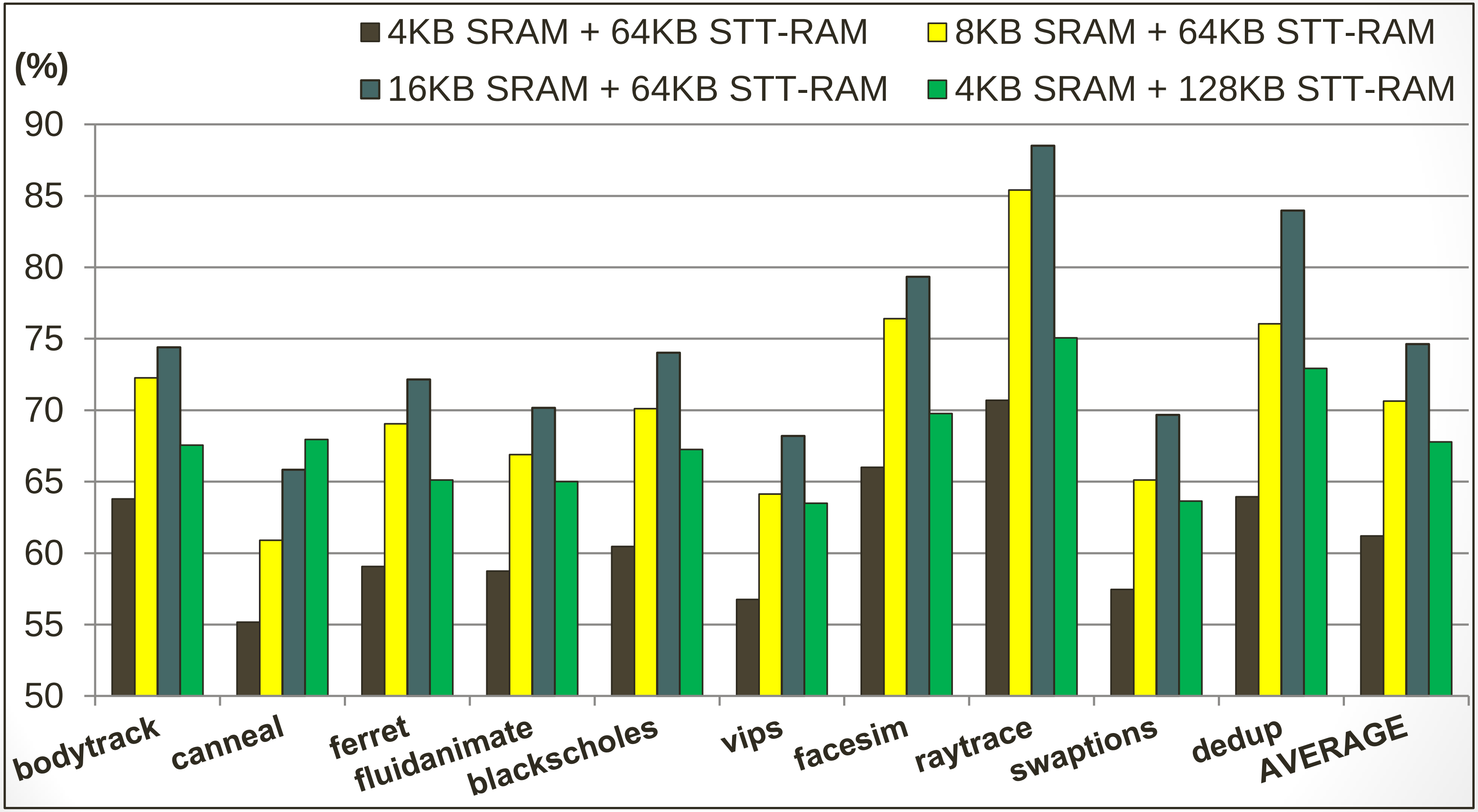
# Evaluation

- PARSEC on MARSSx86 [*Patel et al.11*]
  - IPC (*Instruction Per Cycle*)
- NVSim [*Dong et al. 12*]
  - Latency, area and energy numbers (32nm)
- Configuration
  - Quadcore machine with two-level cache hierarchy
  - Relaxed STT-RAM's non-volatility with a ***26.5µs*** retention period [*Sun et al. 11*]
  - Various cache size combinations within the baseline area budget (*64KB SRAM*)

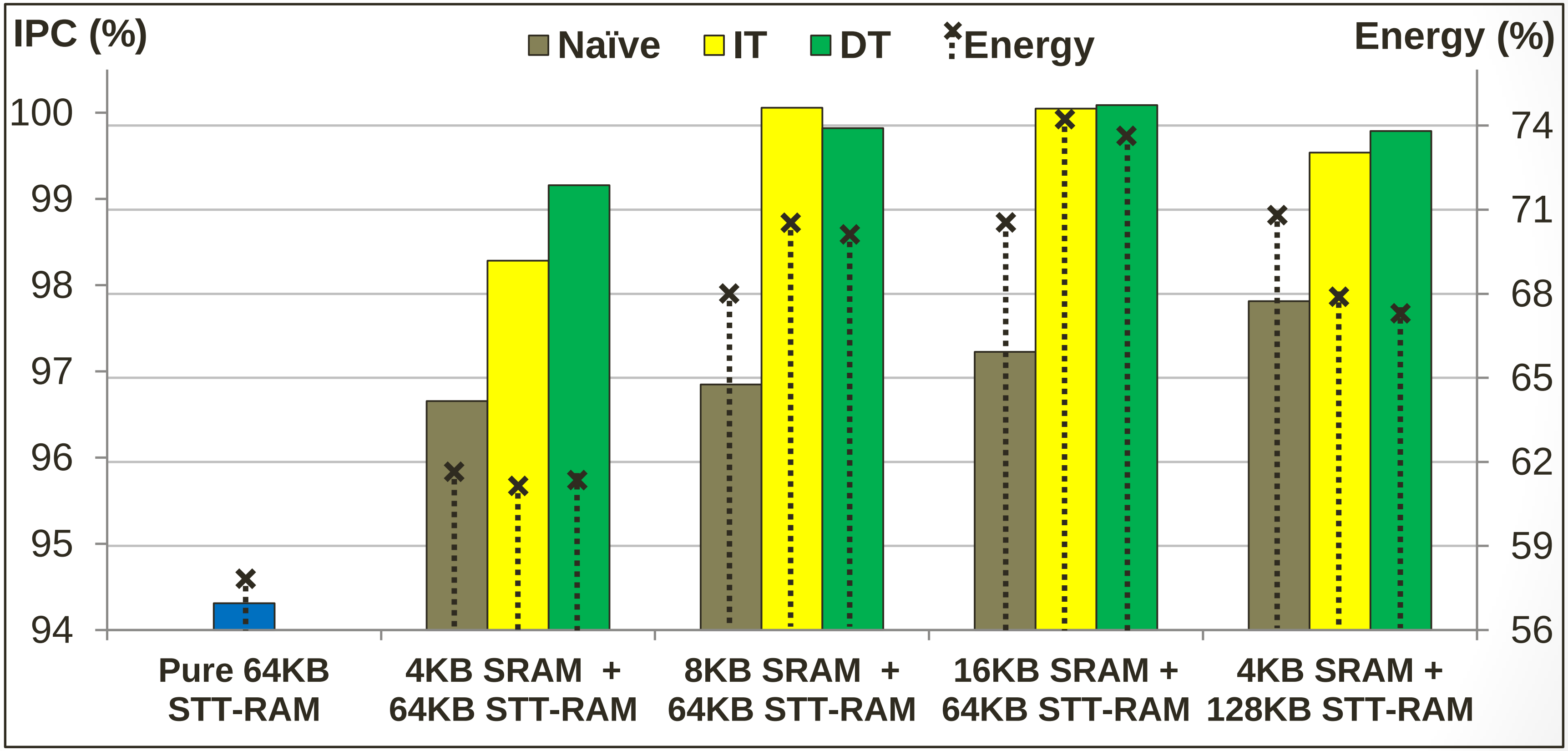
# Normalized IPC (IT policy)



# Normalized Energy (IT policy)



# Comparison of Transfer Policies



# STT-RAM Endurance

- Lifespan programming cycles
  - SRAM and DRAM: **10^16**
  - STT-RAM prediction [*Tabrizi 07*] : **10^15**
  - STT-RAM reported [*Diao et al. 07*] : **10^13**
  - SLC NAND flash: **10^5**
- Writes in L1 cache
  - High intensity
  - Non-even distributed
    - *bodytrack*: ~35% writes on one cache partition
    - *facesim*: ~50% writes on the same cache partition, ~15% on the same block!

# STT-RAM Endurance (cont.)

- *facesim*

| | Perfect distributed | Worst Partition | Worst Block |
|---|---|---|---|
| Baseline SRAM | 1,300+ years | 300+ years | < 360 hrs |
| Baseline STT-RAM | 1.3 years | 0.3 years | < 22 mins |
| Hybrid Naïve | 3.5 years | 1.0 year | 0.9 hr |
| Hybrid IT | 41.2 years | 6.9 years | 51.6 hrs |
| Hybrid DT | 32.9 years | 7.0 years | 54.3 hrs |

150x lifespan increases for the worst block!

# Conclusion

- Deploy STT-RAM as L1 cache
    - Expensive write (latency, energy and endurance)
- Architecture solution: hybrid cache
    - “big.LITTLE” model
- MESI-based Hybrid L1 Cache Architecture
    - Small SRAM partition + large STT-RAM partition
    - Using built-in information from coherent protocol
    - Performance maintained with less energy, and extended lifespan

# THANK YOU !

Q & A