

Lee TACO 2015

A New Memory-Disk Integrated System with Hardware Optimizer

NVRAM Opens Doors

- ✦ Emerging RAM replacement technologies
 - ✦ PCM, FeRAM, MRAM, Memristors, STTRAM
- ✦ More dense than RAM, not quite as fast, may wear out
- ✦ Nonvolatile (because charge-based devices become unstable when geometry is small enough for tunneling)
- ✦ How does that change our perspective?

Old Familiar

- ✦ Volatile memory is fast, small, fine-grained access
- ✦ Nonvolatile storage is slow, large, large granularity
- ✦ Desktop/filing cabinet analogy
 - ✦ But where the cabinet always holds a copy
 - ✦ Files are opened, read in, written out, closed
- ✦ Paging to disk is like having a reserved space in the cabinet for current work that doesn't fit the desk

What it Means

Much of the activity and content of RAM is copying data from/to nonvolatile memory

What it Means

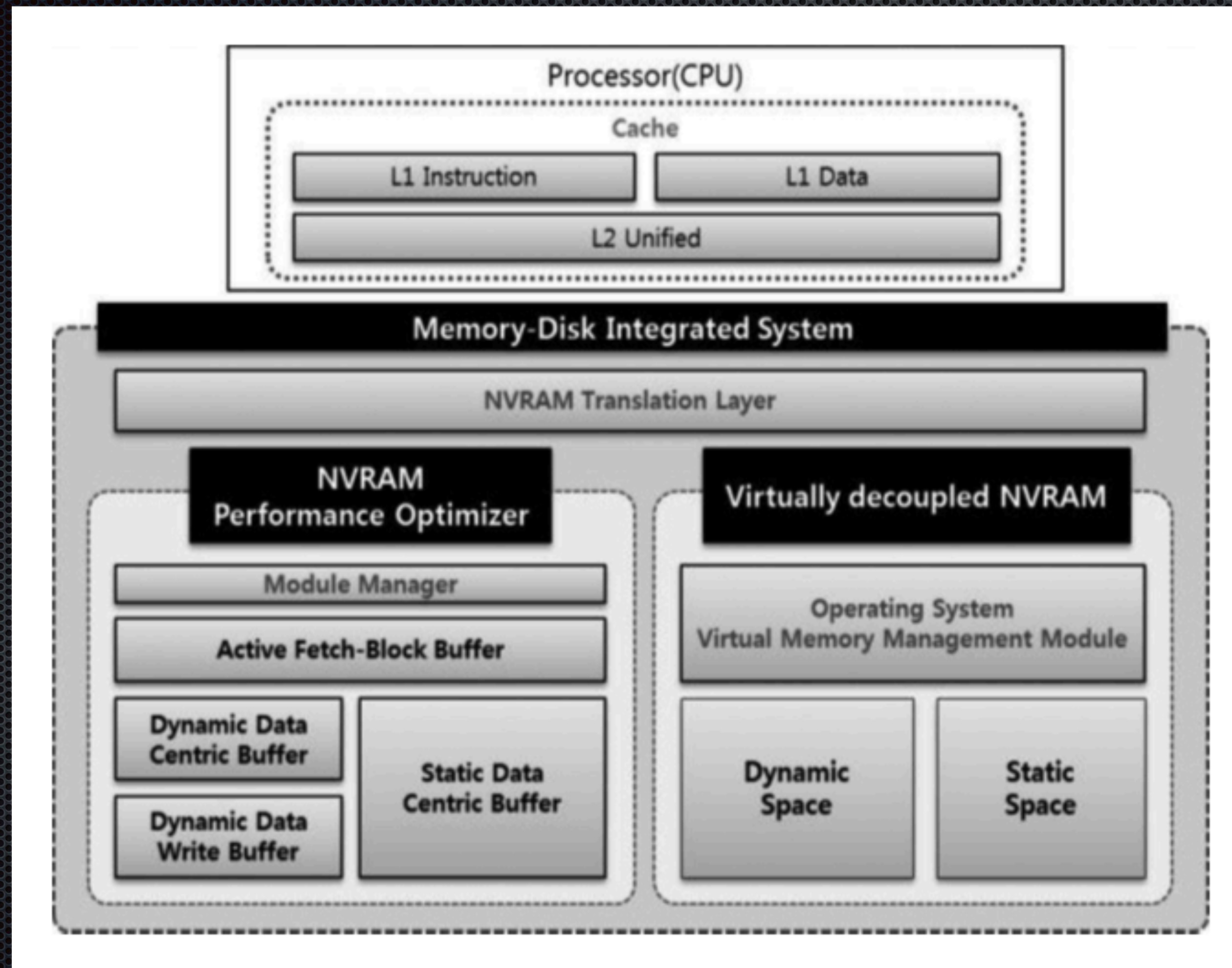
Much of the activity and content of RAM is copying data from/to nonvolatile memory

But what if RAM itself is no longer volatile?

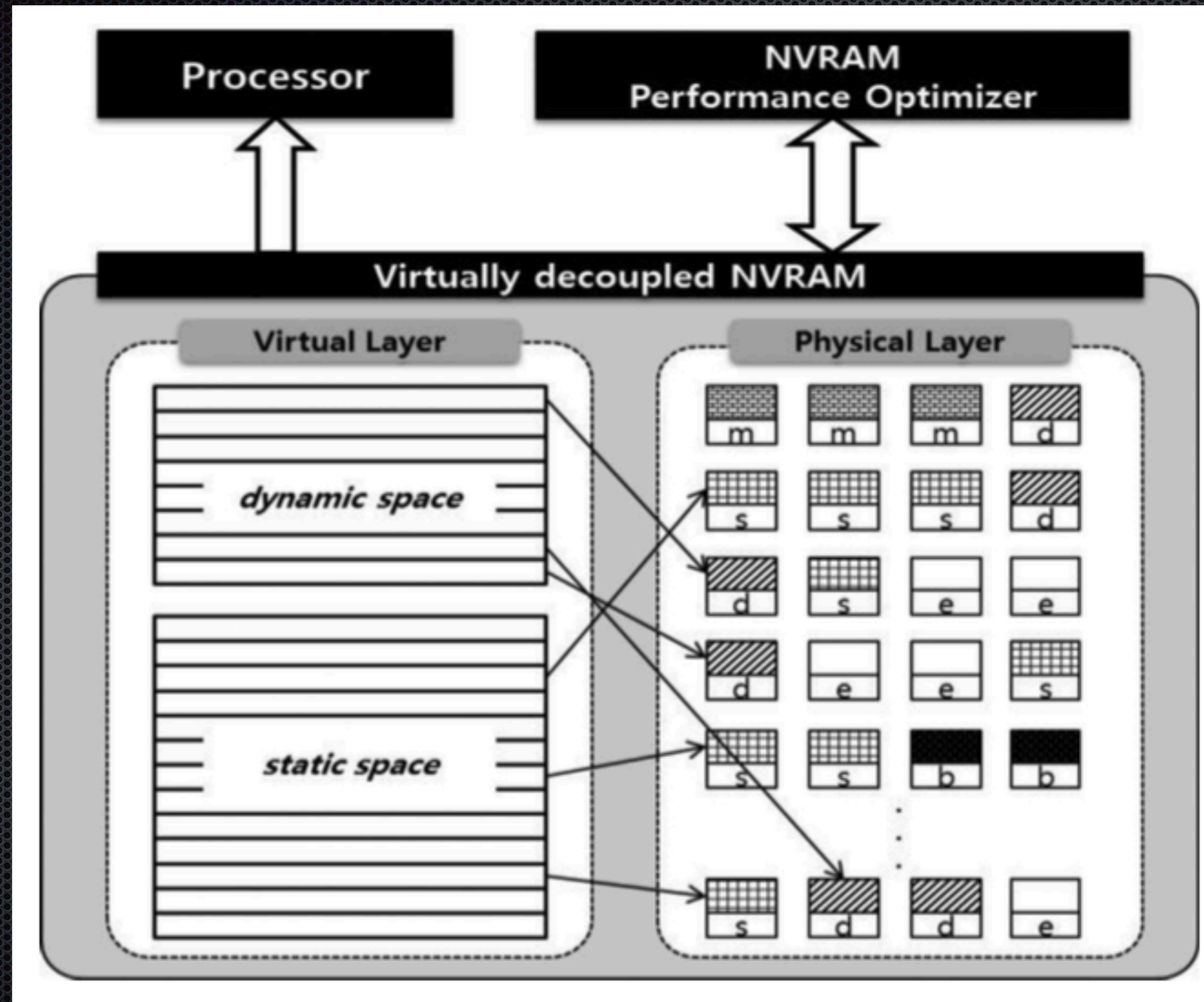
Static and Dynamic Spaces

- ✦ Static holds program files and static data
 - ✦ Accessed and linked directly, without copying
- ✦ Memory allocation and deallocation are in dynamic space
- ✦ Requires OS modification
- ✦ No paging

Overall Architecture

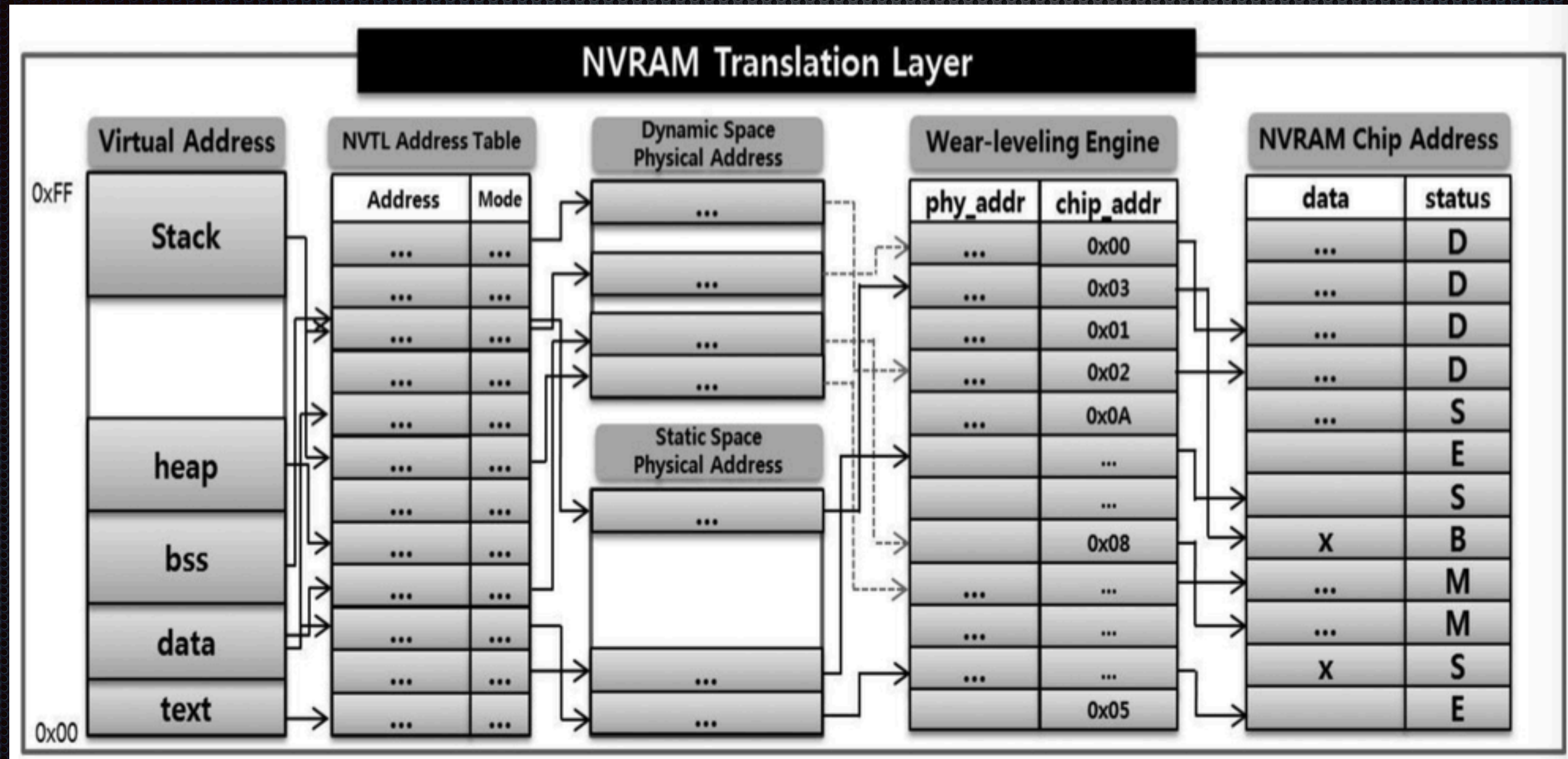


Virtually Decoupled NVRAM



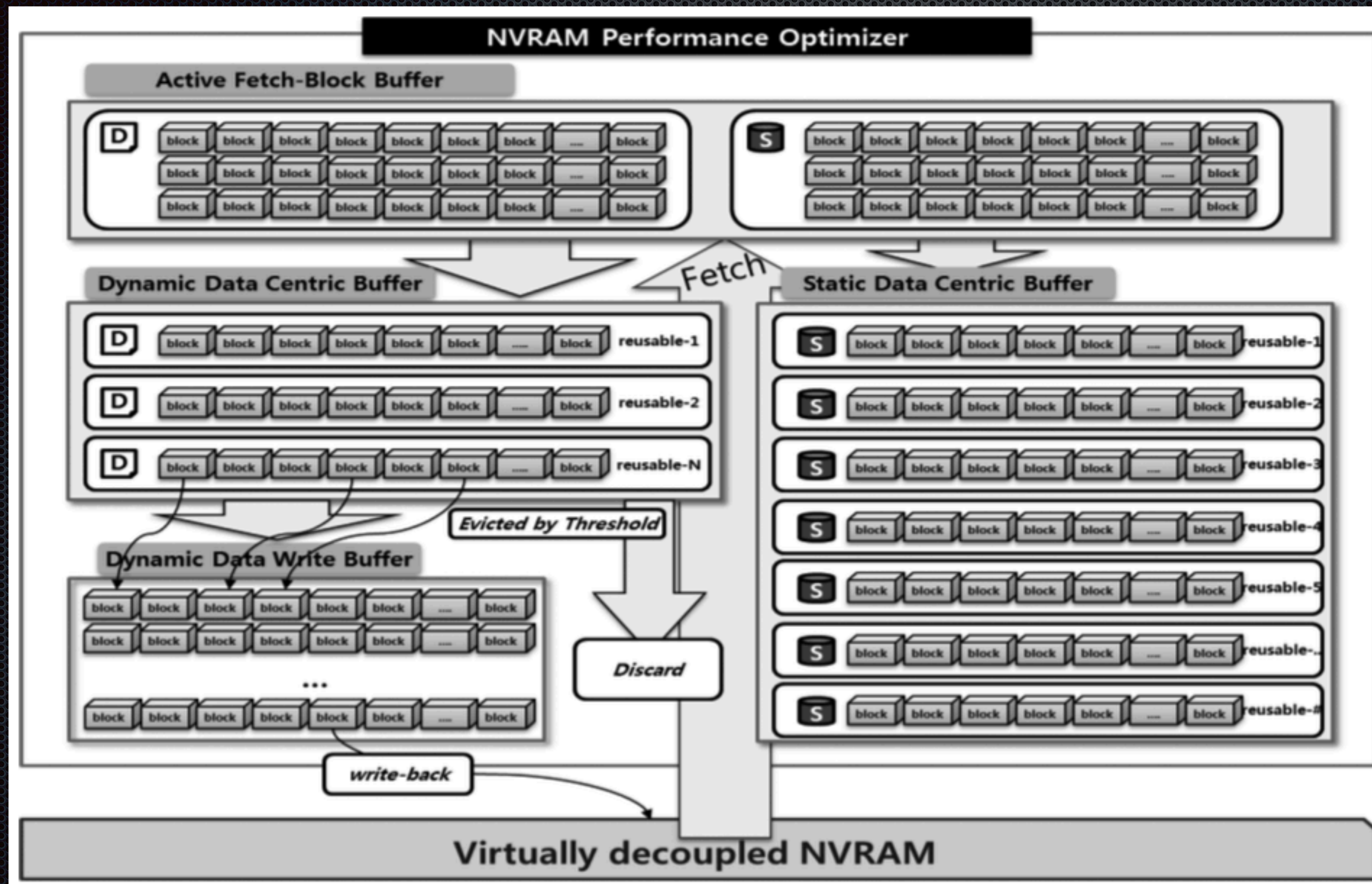
Dynamic
Static
Empty
Bad

NVRAM Translation Layer

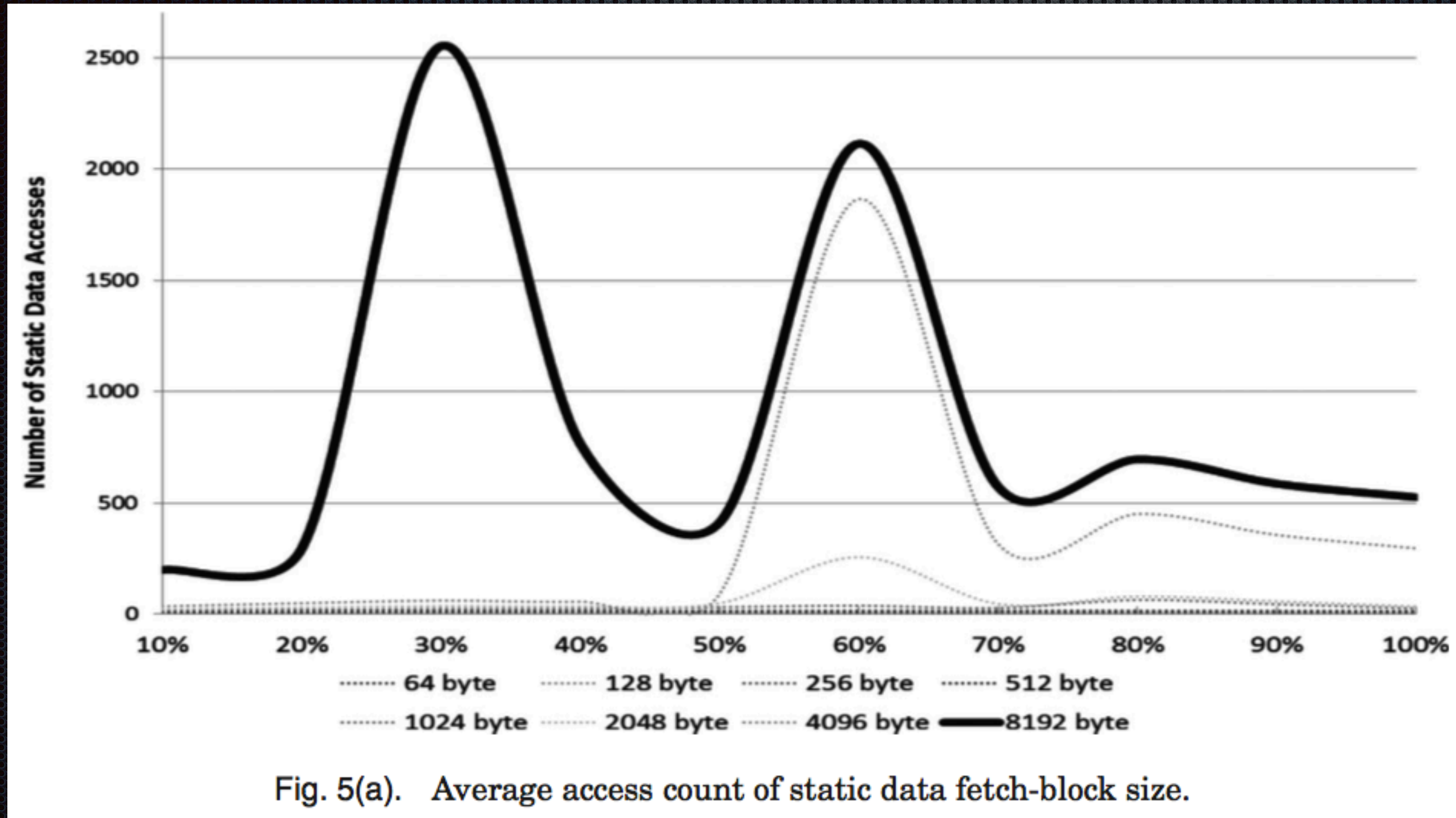


With NVRAM, Physical is the New Virtual

Performance Optimizer

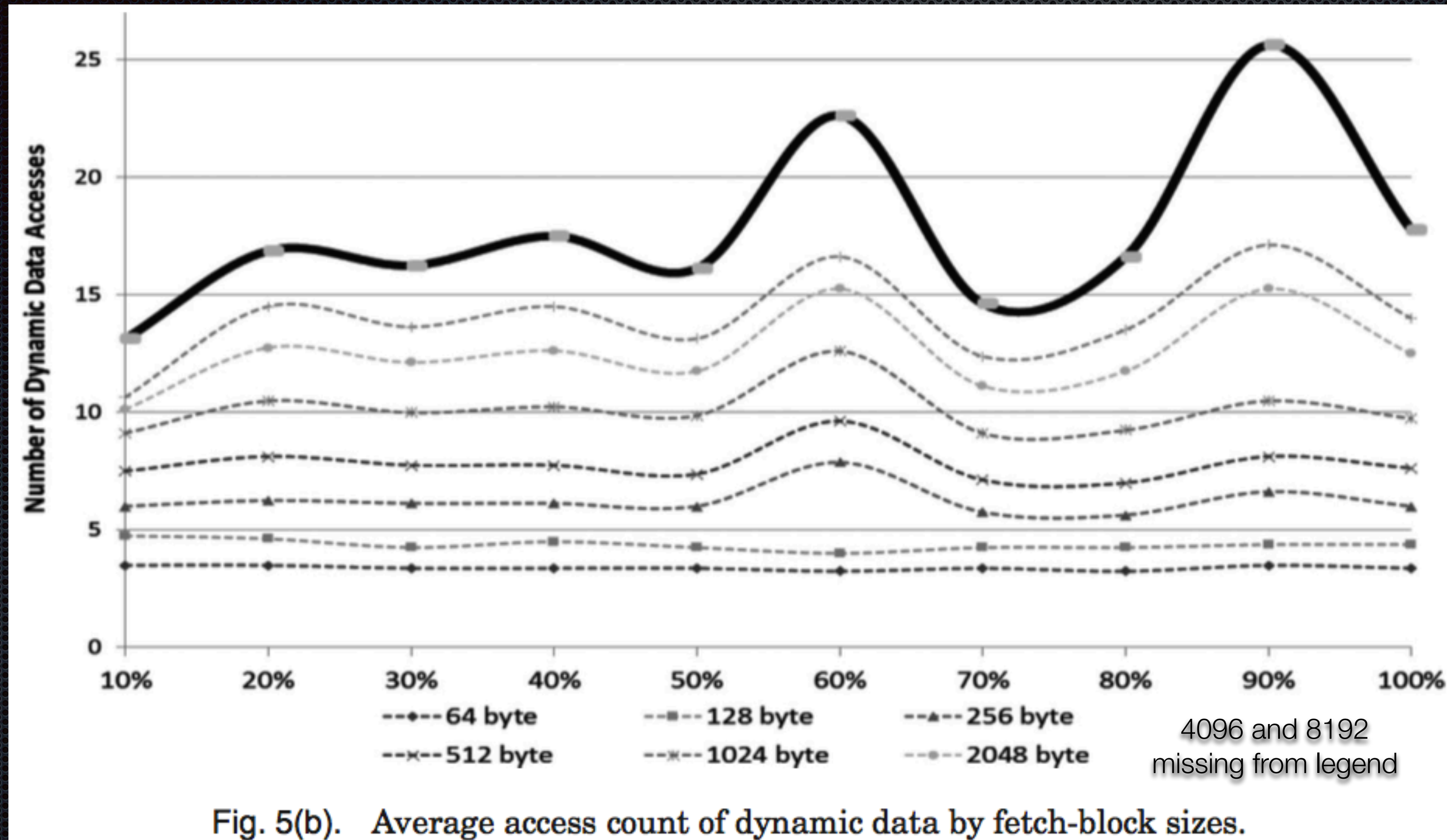


Static Data Fetch Size



No surprise -- page-like fetches are most common

Dynamic Access



Allocations also tend to be page sized

Active Fetch Block Buffer

- ✦ 1MB RAM space for 8K blocks, FIFO replacement
- ✦ Separate dynamic and active buffer space
- ✦ Blocks can be split into 4K units
- ✦ Holds data that is recently fetched
 - ✦ Tenure in AFBB enables analysis of behavior
 - ✦ Subsequent eviction to corresponding buffers

8K Sweet Spot

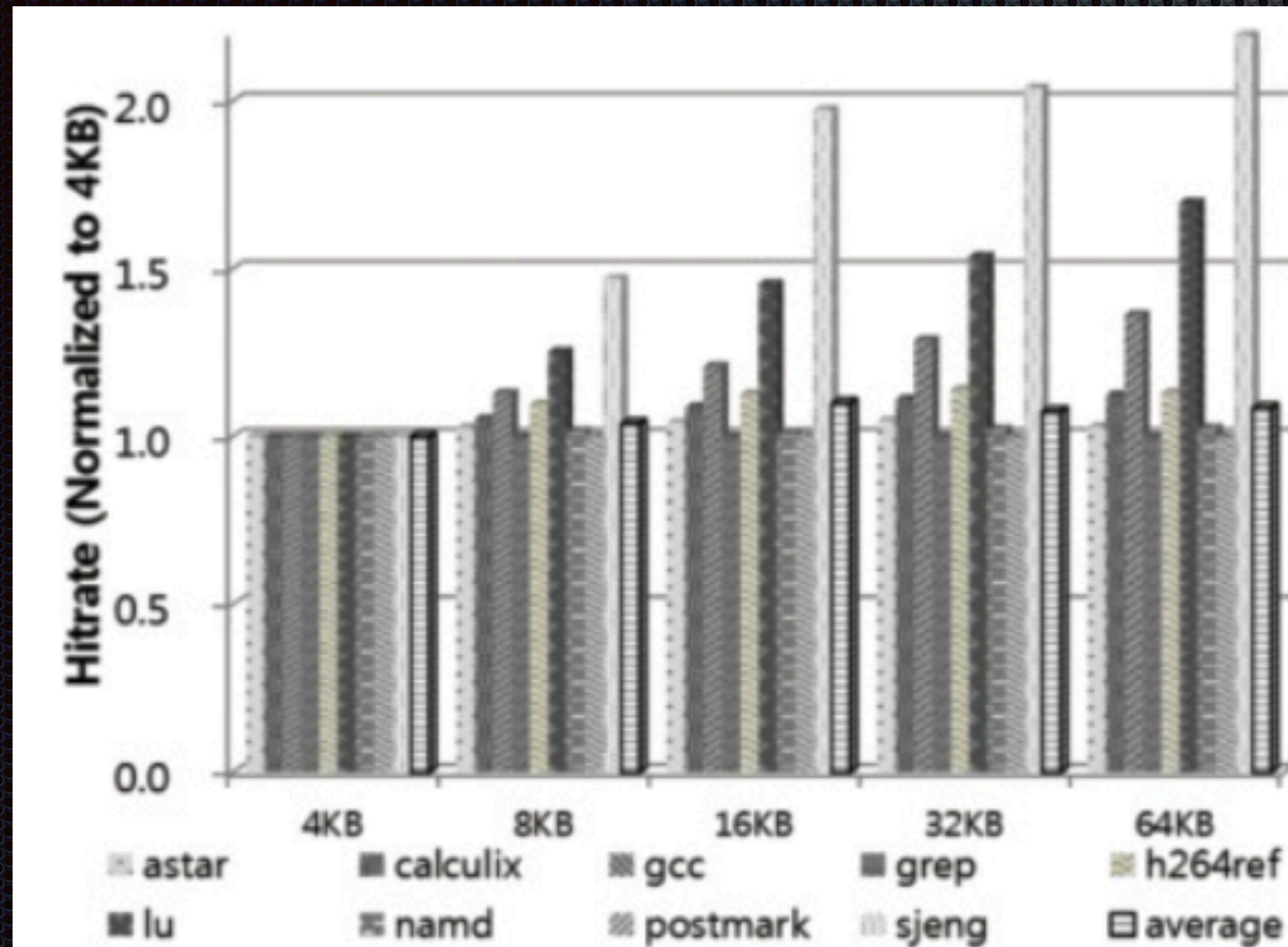


Fig. 6(a). Hit ratio of AFBB by fetch-block size.

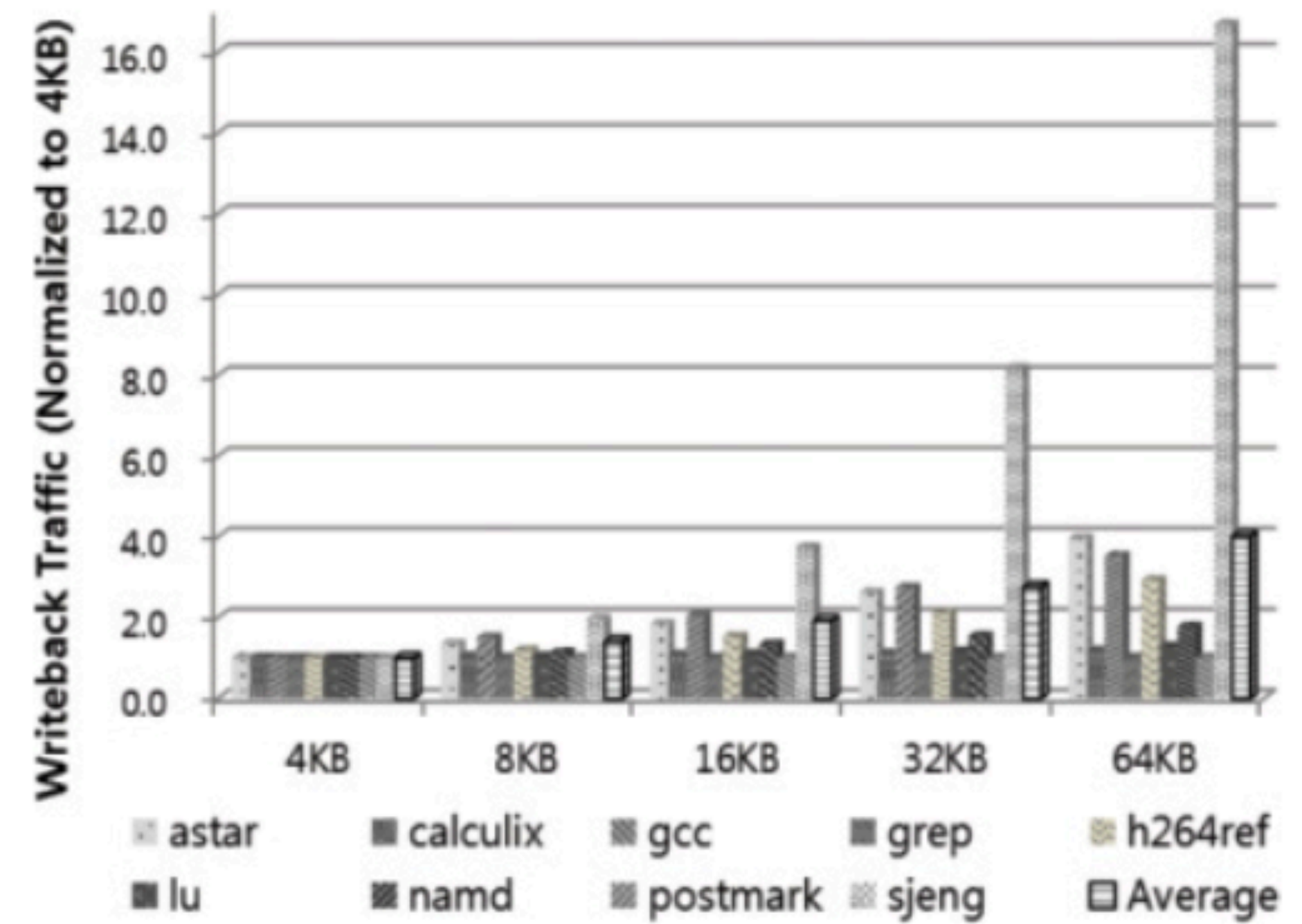


Fig. 6(b). Write-back traffic of AFBB by fetch-block size.

Larger AFBB blocks have better hit rate but also induce more write back traffic

Static Data Centric Buffer

- ✦ Holds static data from AFBB in 4K blocks
 - ✦ 8K blocks are divided if access is only to one half
 - ✦ Other 4K is discarded (static blocks are never dirty)
- ✦ Hit rate threshold replacement, no write back
- ✦ 512 KB
- ✦ Management table keeps hit statistics

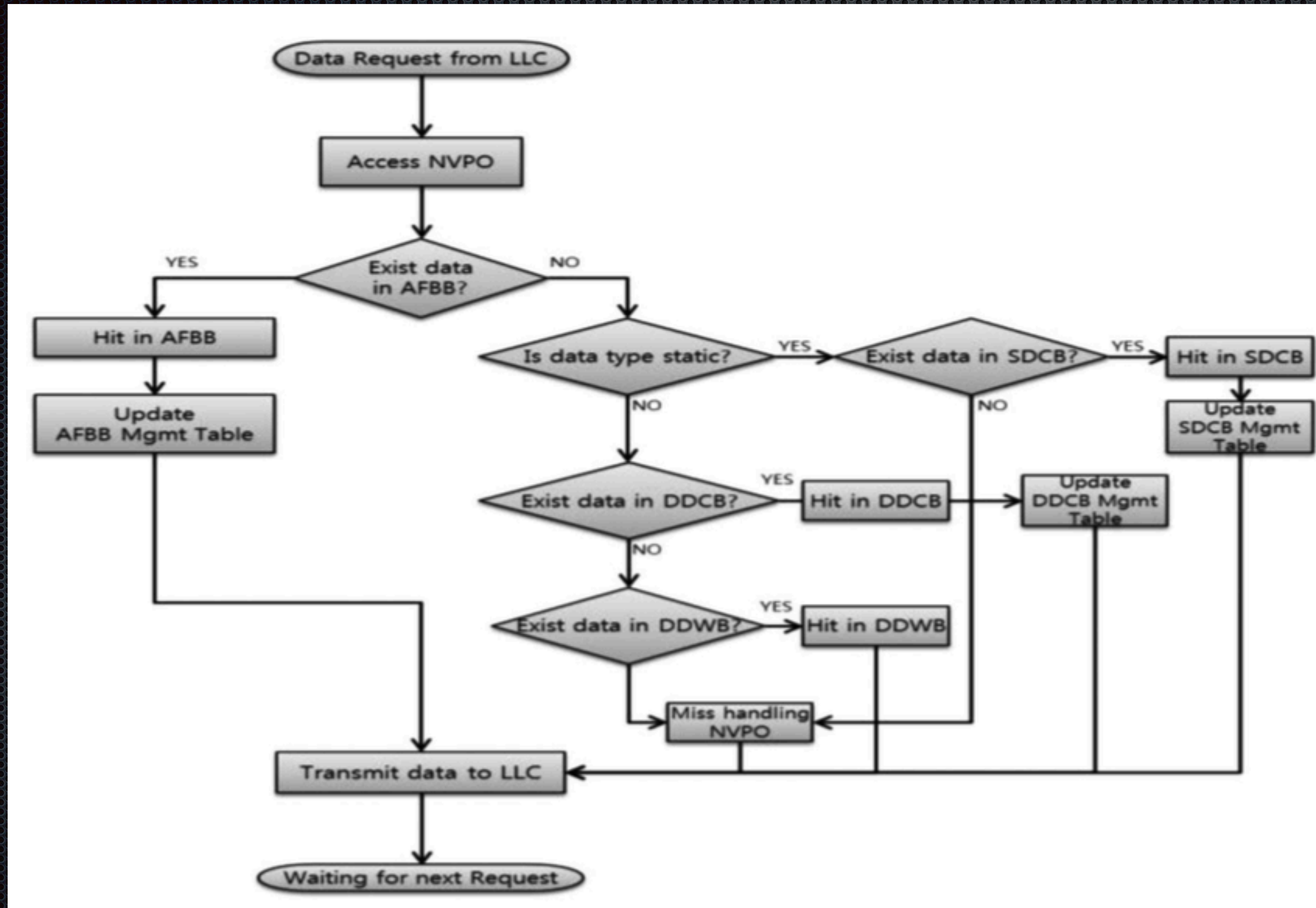
Dynamic Data Centric Buffer

- ✦ Holds dynamic data from AFBB in 4K blocks
 - ✦ 8K blocks are divided if access is only to one half
 - ✦ Other 4K is written back if dirty
- ✦ Hit rate threshold replacement, write back if dirty
- ✦ 265 KB
- ✦ Management table keeps hit statistics

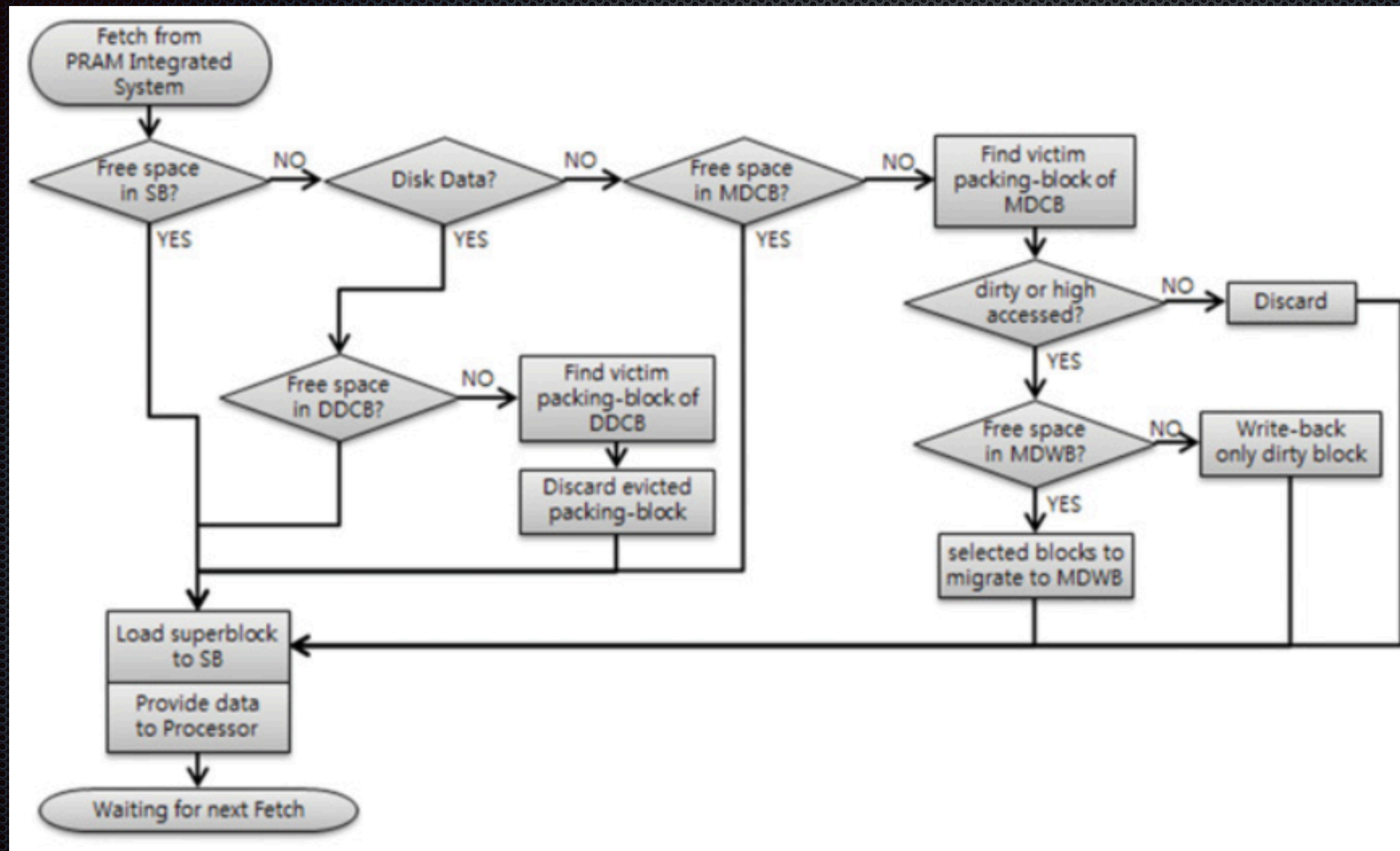
Dynamic Data Write Buffer

- ✦ 256 KB FIFO write buffer for DDCB evictions
- ✦ Data can be read back from queue
- ✦ Hides NVRAM write latency
- ✦ Third level of write path
 - ✦ Increases time before NVRAM write
 - ✦ More chance for recall in cases of lower frequency access (reducing unnecessary writes = less wear)

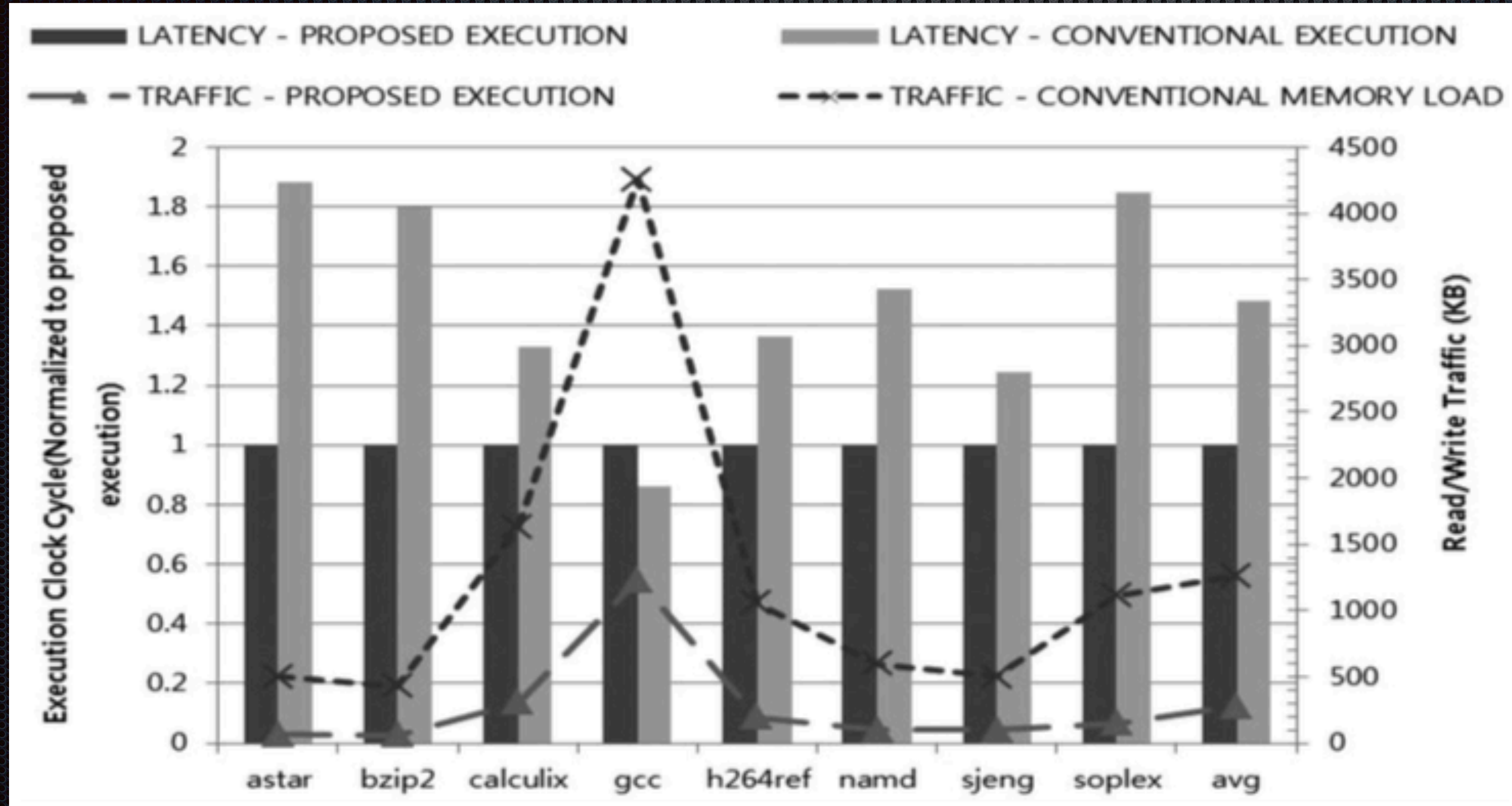
Operation



Miss Handling



Load Latency and Traffic



AFBB Optimal Capacity

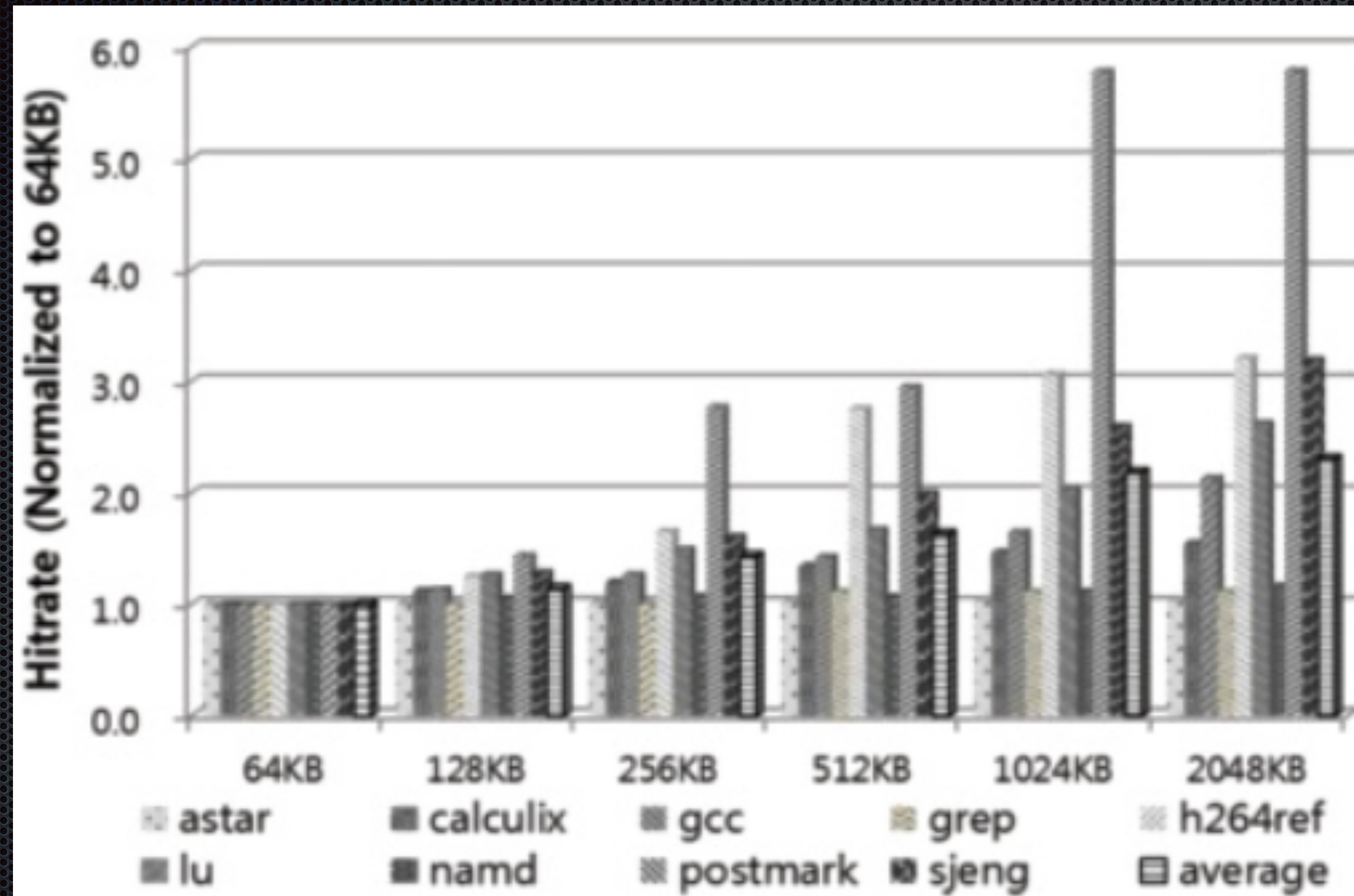


Fig. 11. Hit rate of AFBB by capacity.

SDCB Optimal Capacity

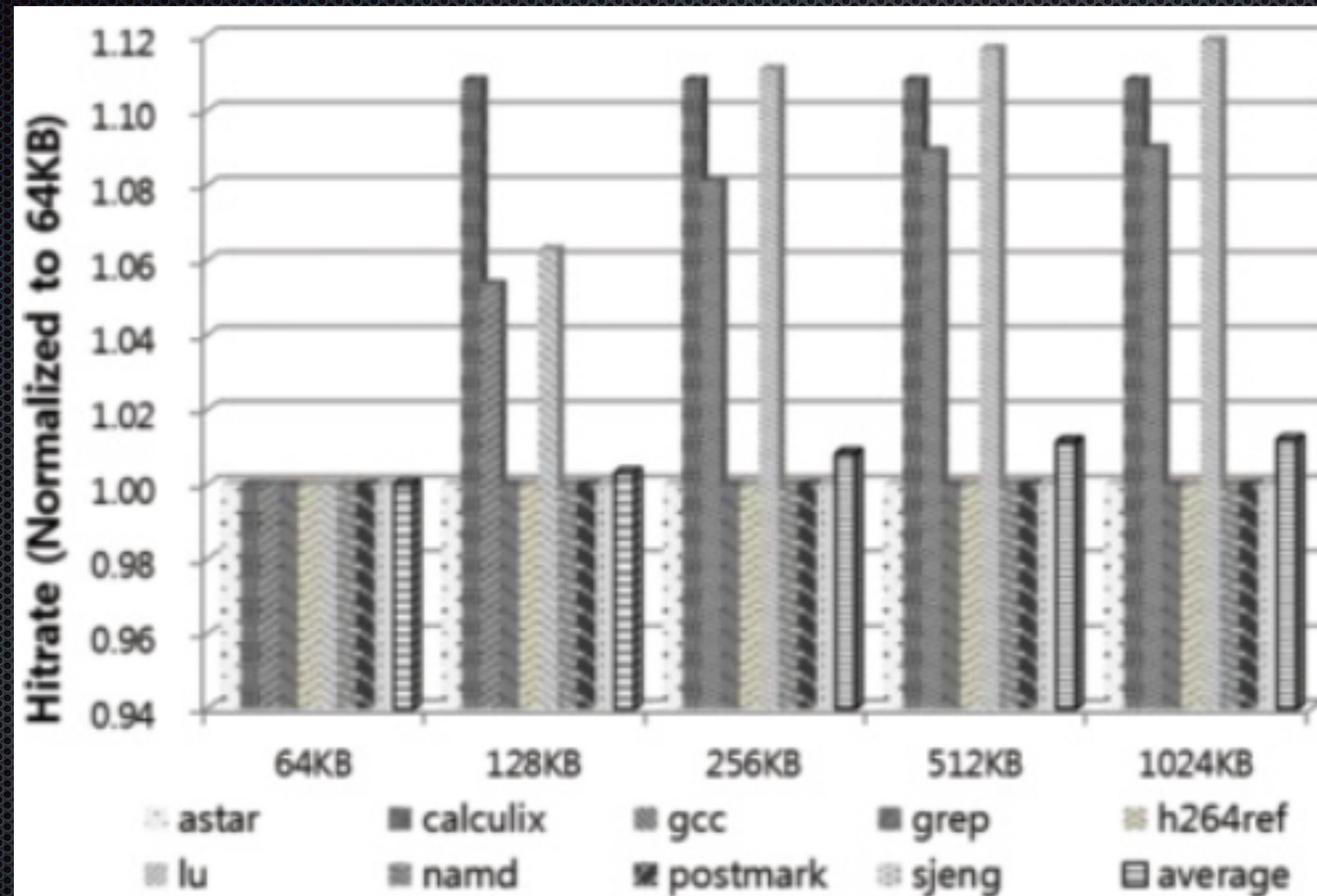


Fig. 12. Hit rate of SDCB by capacity.

Write Traffic

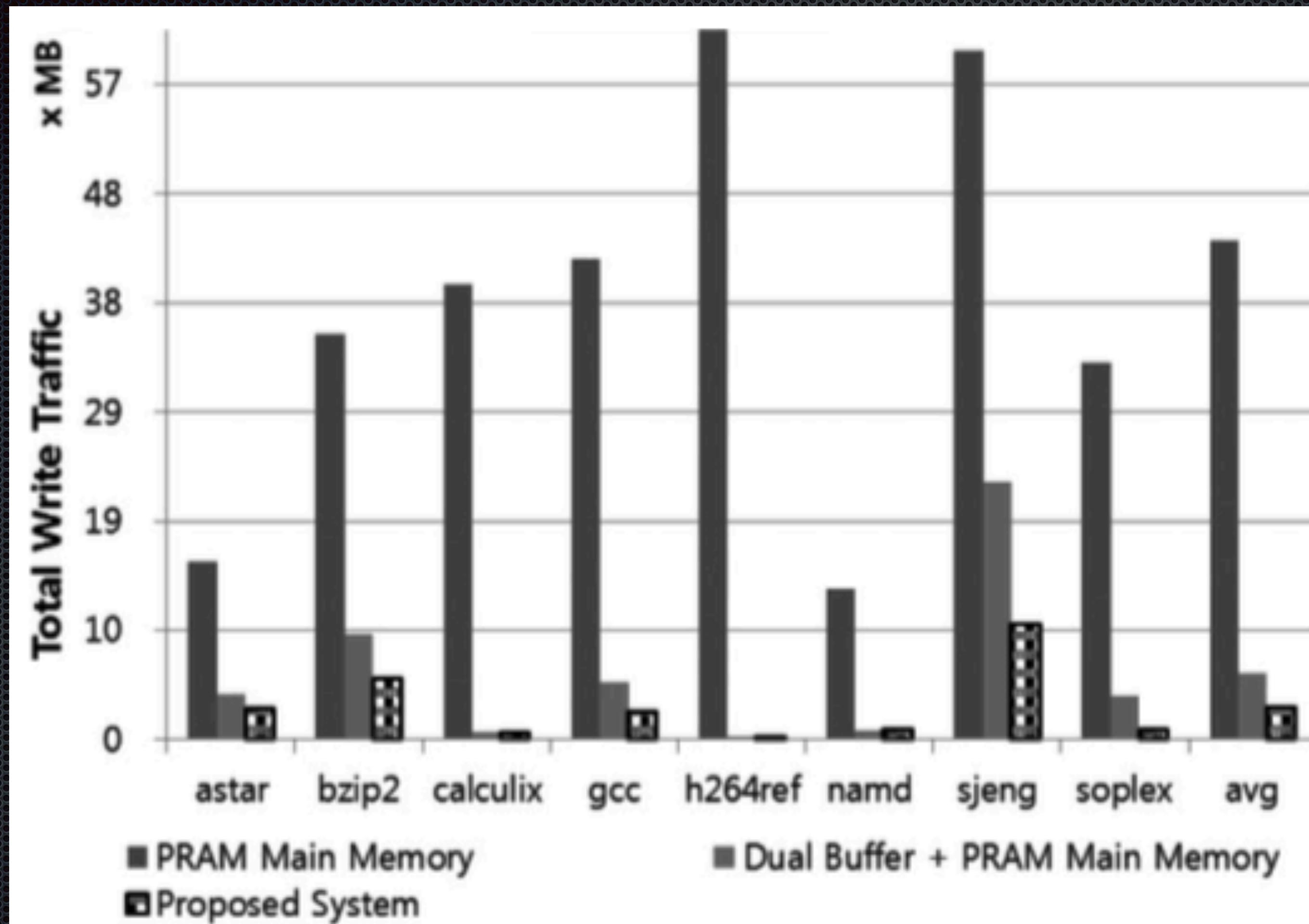
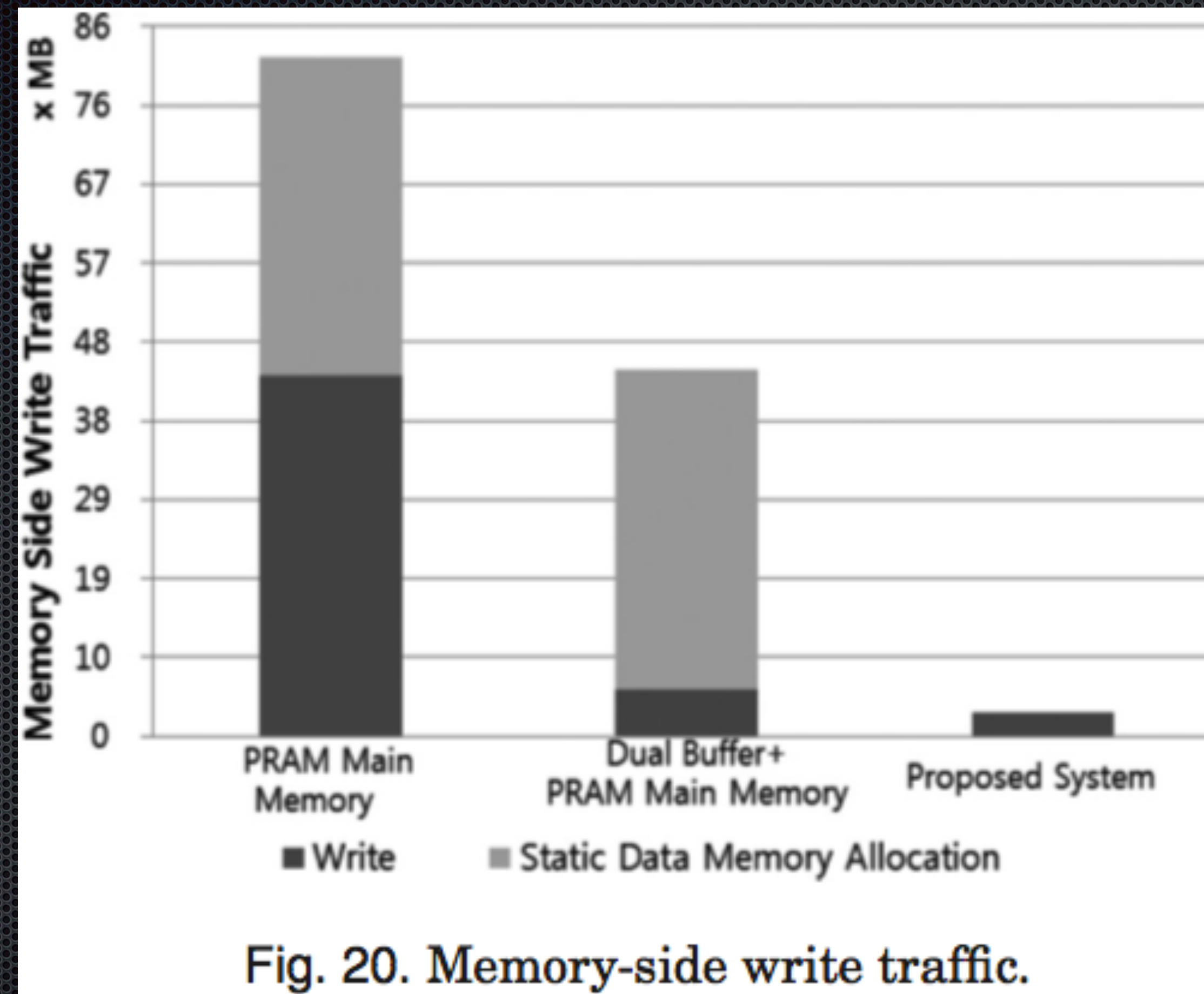


Fig. 19. Total write traffic.

Memory-Side Writes



Average by Type

Estimate Lifetime Effect

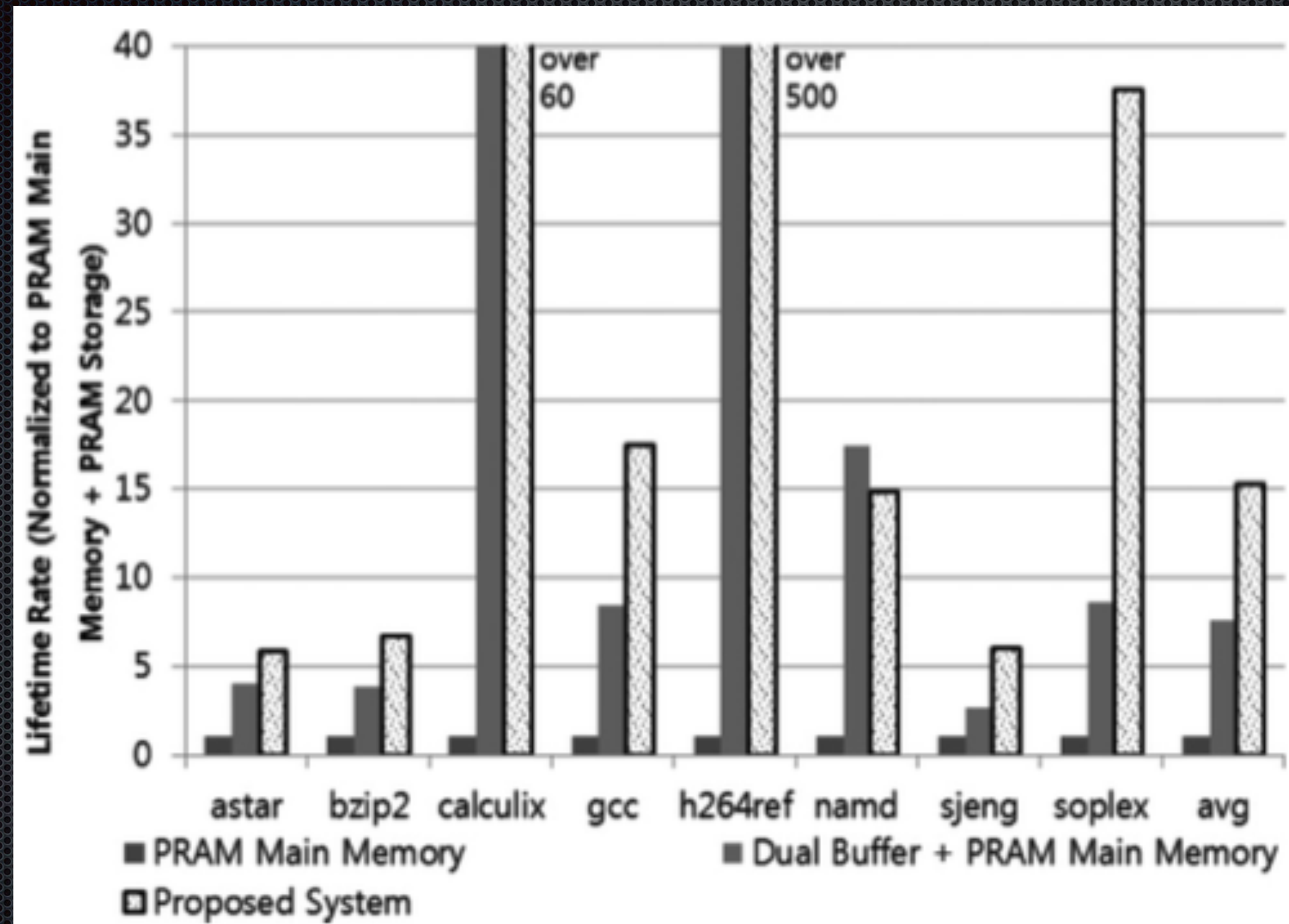
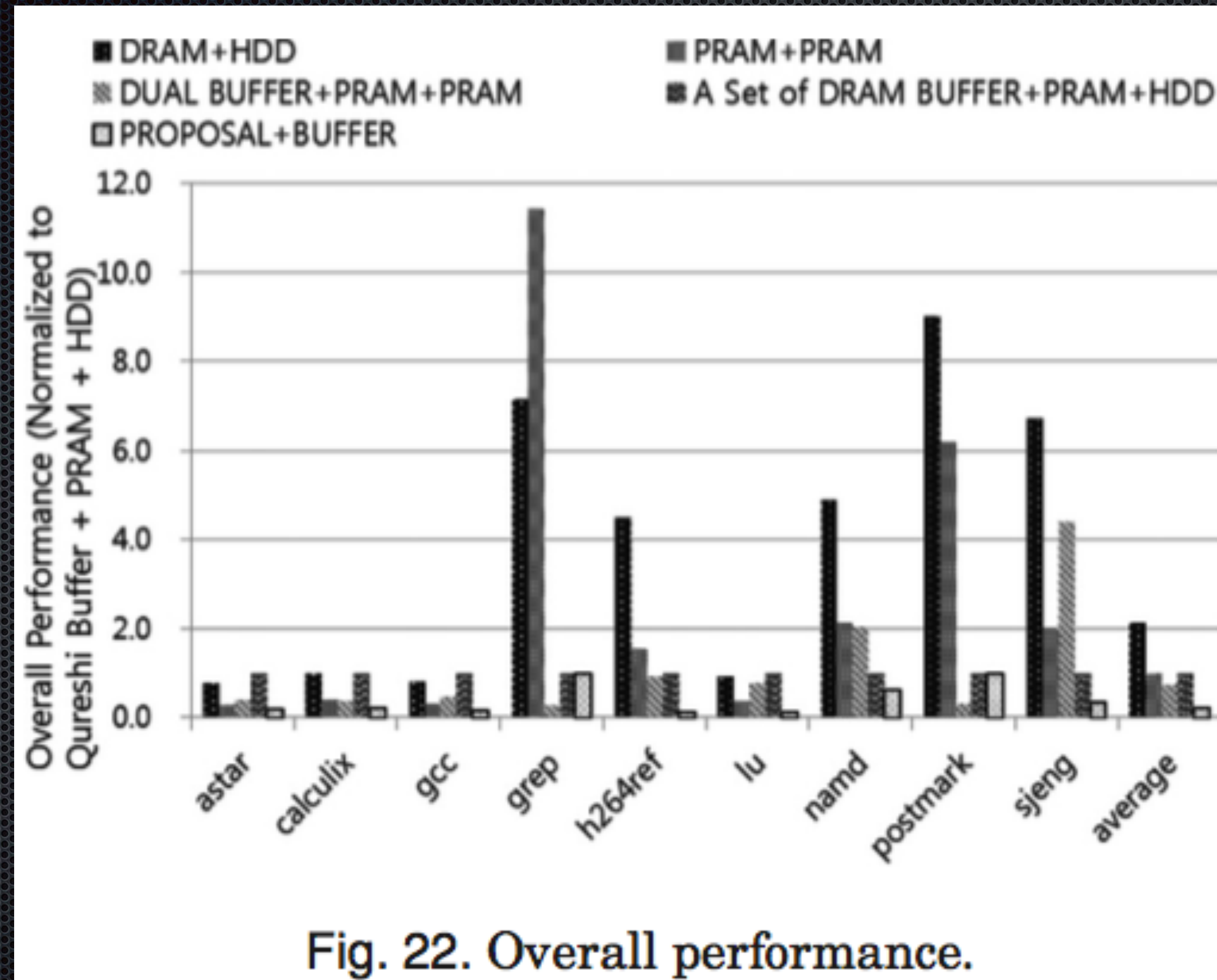


Fig. 21. PRAM lifetime.

Performance



Power

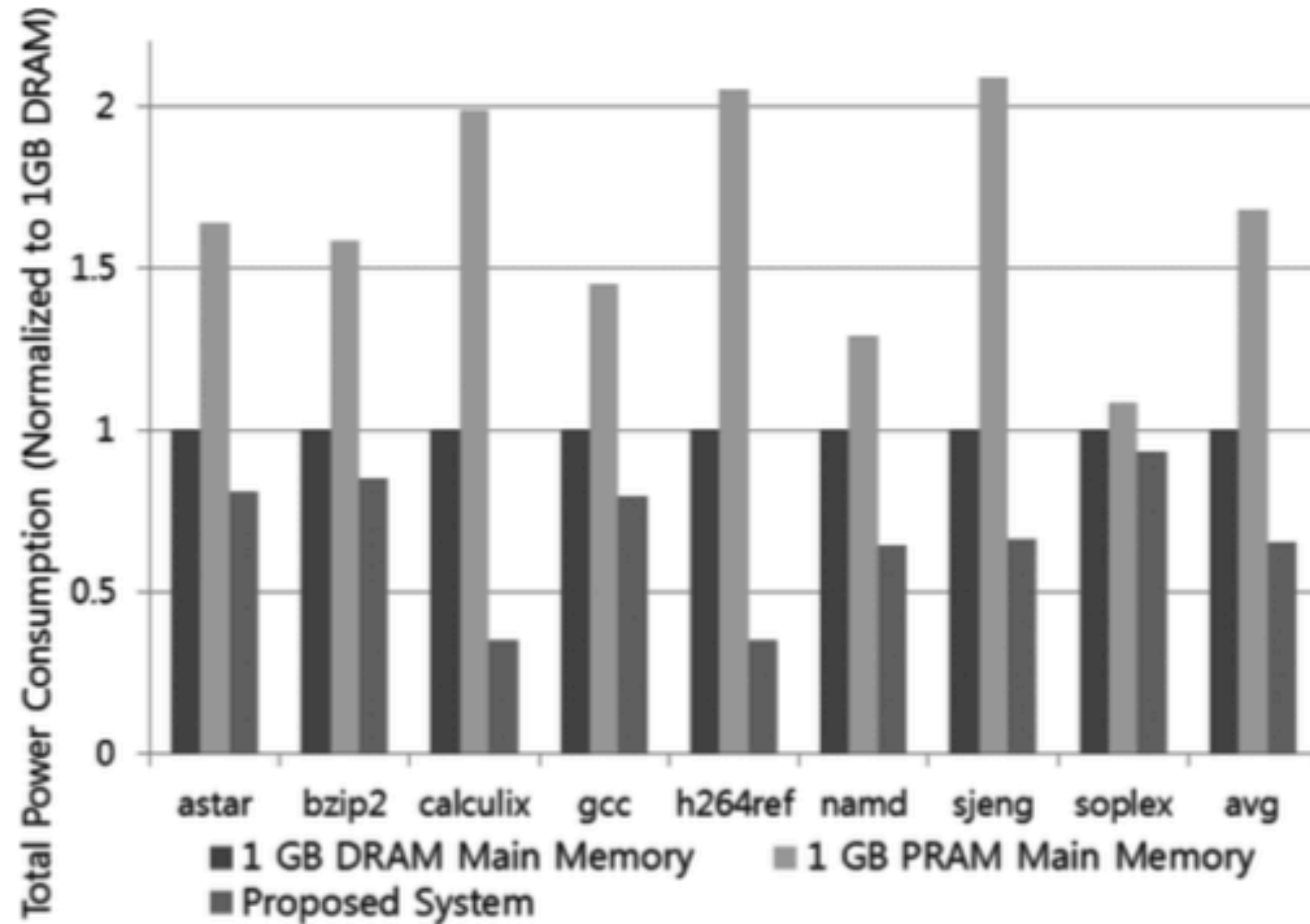


Fig. 23. Total power consumption.

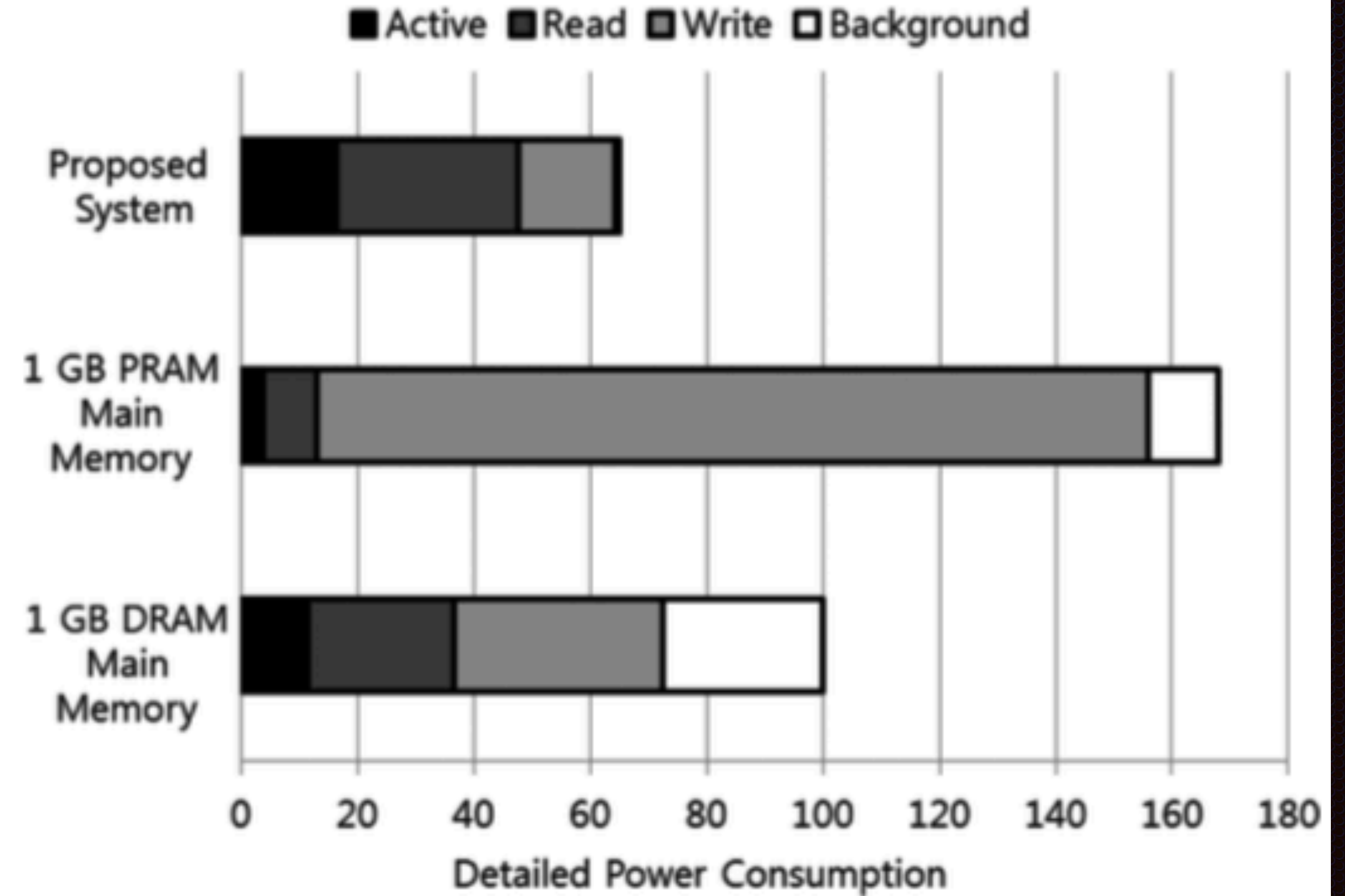


Fig. 24. Detailed power consumption.

Discussion

Liu ASPLOS 2014

NVM Duet: Unified Working Memory and Persistent Store Architecture

Persistent Store

- ✦ Nonvolatile stores are expected to be consistent and durable
- ✦ Consistent means a group of related state changes commit together, regardless of failure
- ✦ Durable means retention with power off

Memory Model

- ✦ Working memory in SRAM or DRAM
- ✦ Persistent store in PCM
- ✦ Oblivious store scheduling wastes cycles
- ✦ Need to be aware of which is being used

Example

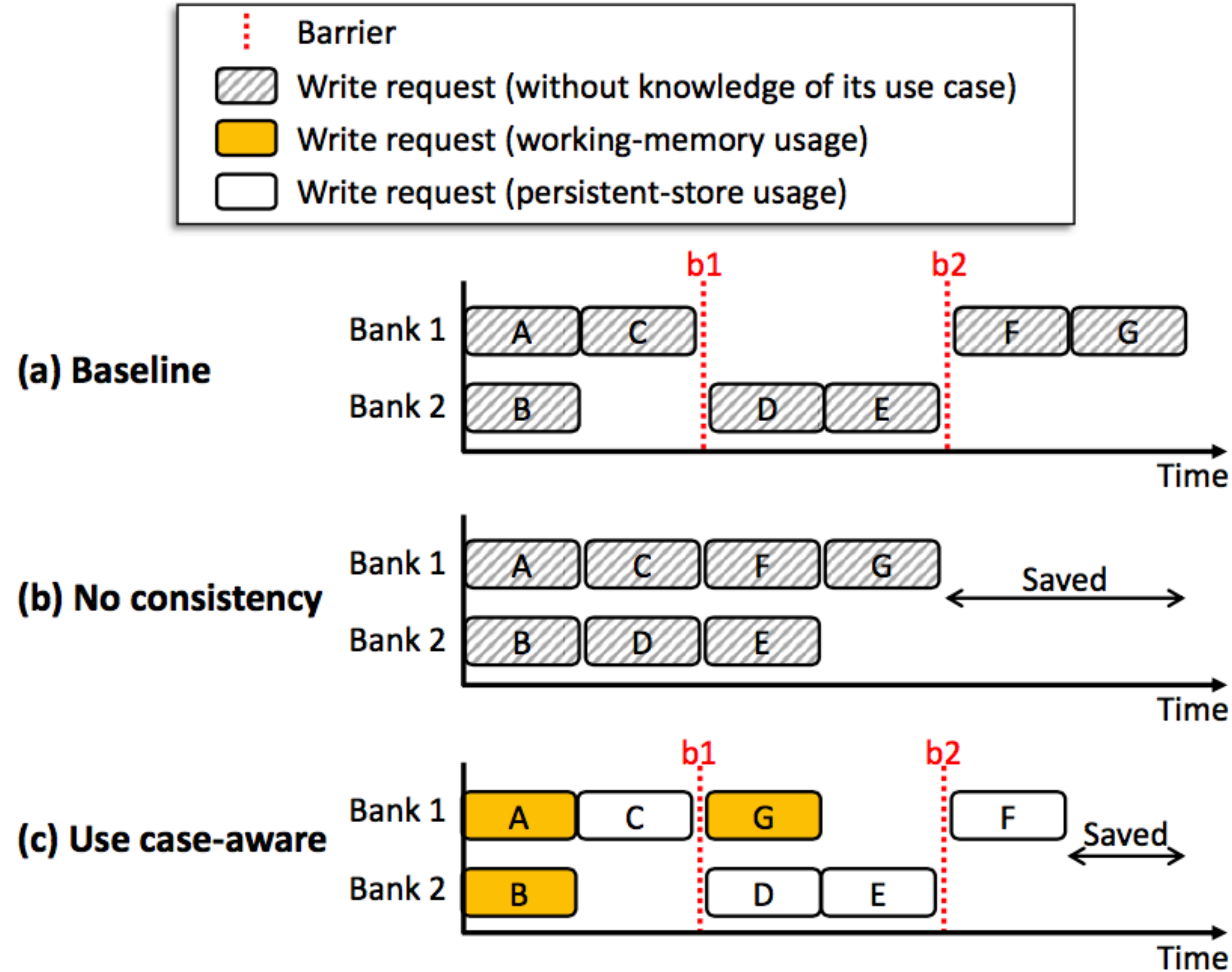
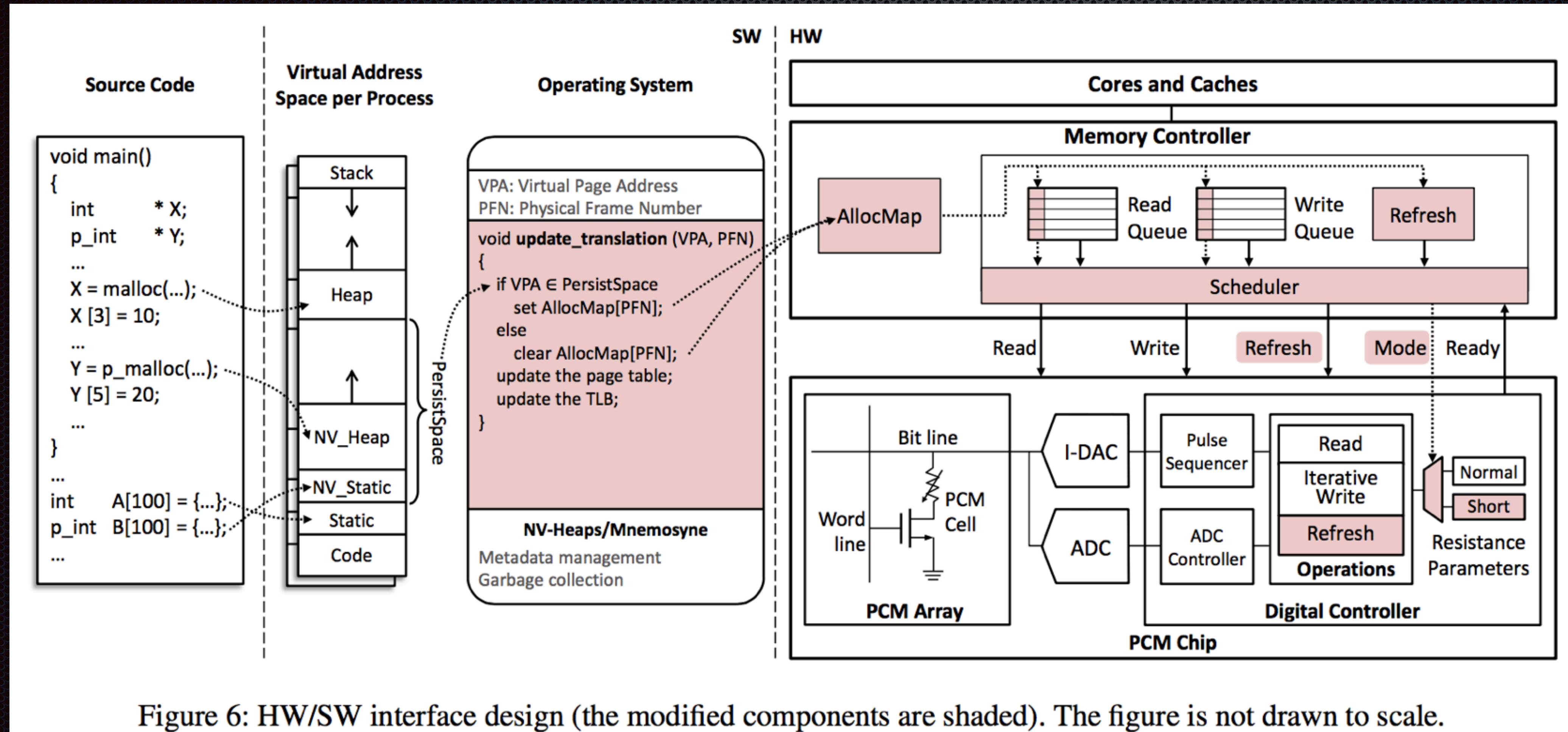


Figure 3: Schedule with mixing requests for persistent store and working memory

System Overview



Programmer specifies persistence

PCM Retention

- ✦ Resistance drifts over time
- ✦ NV writes maximize retention time, but are slow
- ✦ Can exploit for working memory by writing faster
 - ✦ Retains for less time
 - ✦ Needs periodic refresh

Two Scheduling Rules

- ✦ Writes to working memory can ignore barriers
 - ✦ Barriers are for persistent writes
- ✦ Writes to persistent store take priority if there is a barrier pending in the memory controller

Results

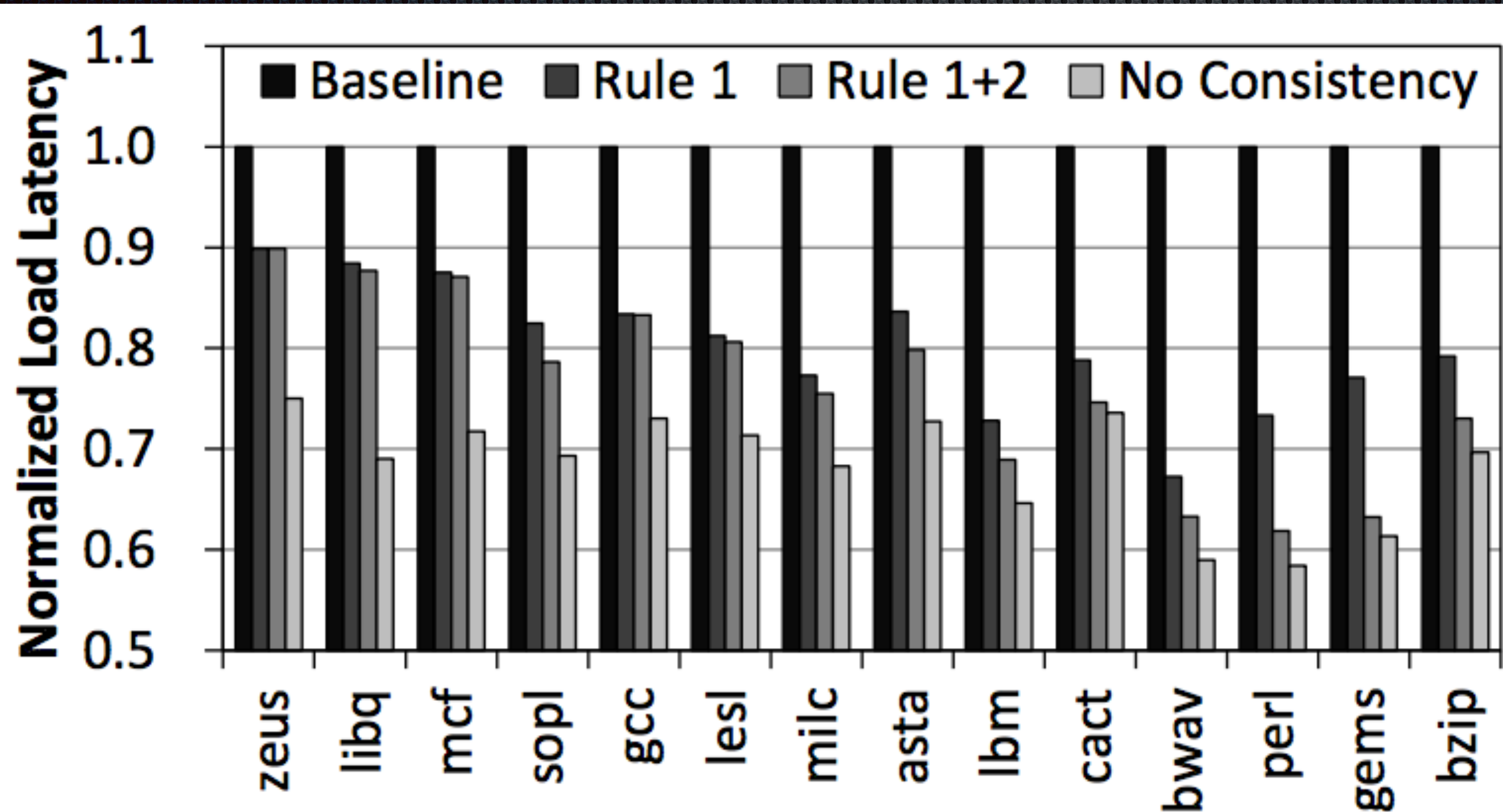


Figure 11: Benefits of Duet Scheduler for load latency

Results

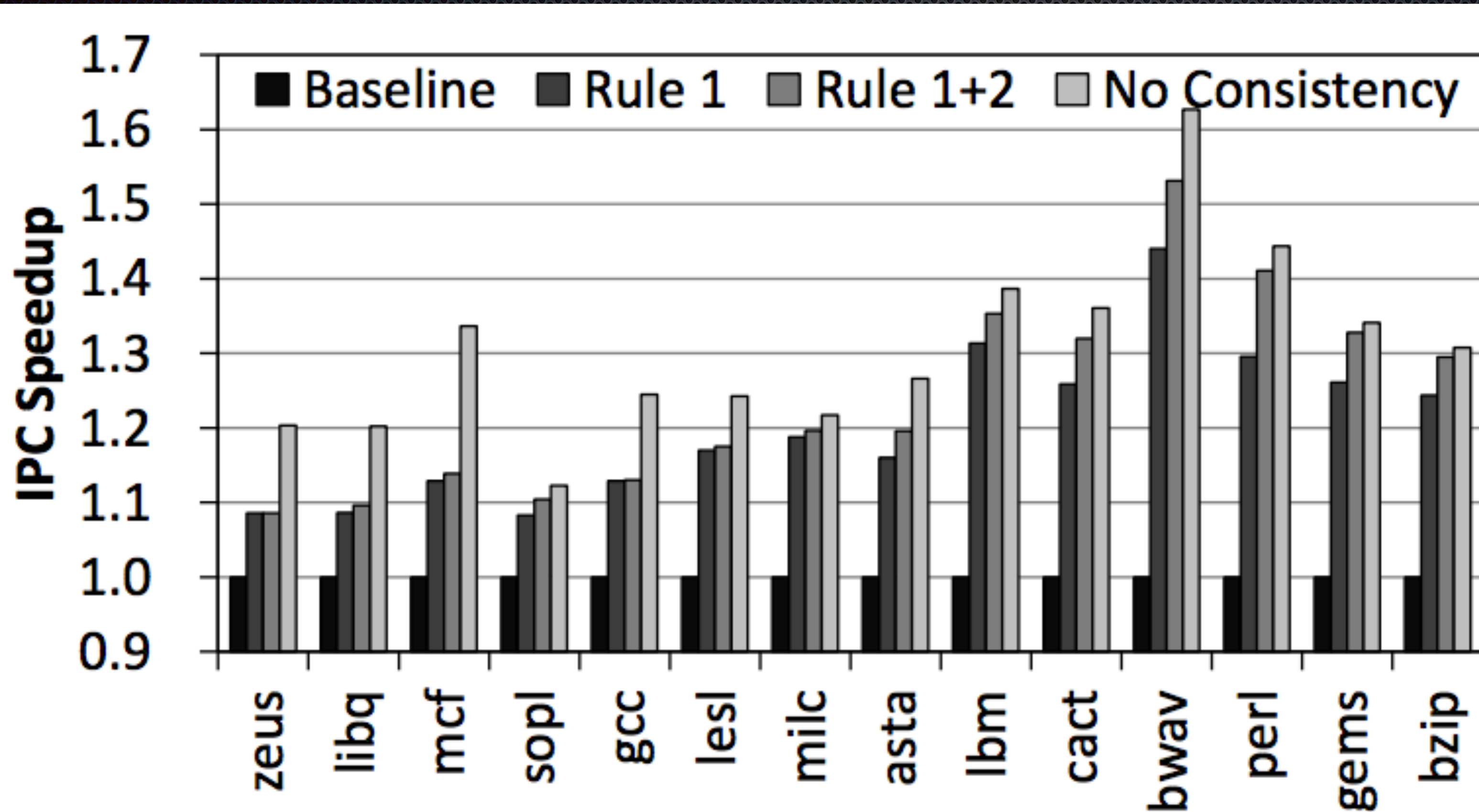


Figure 12: Benefits of Duet Scheduler for IPC

Dual Retention w/Refresh

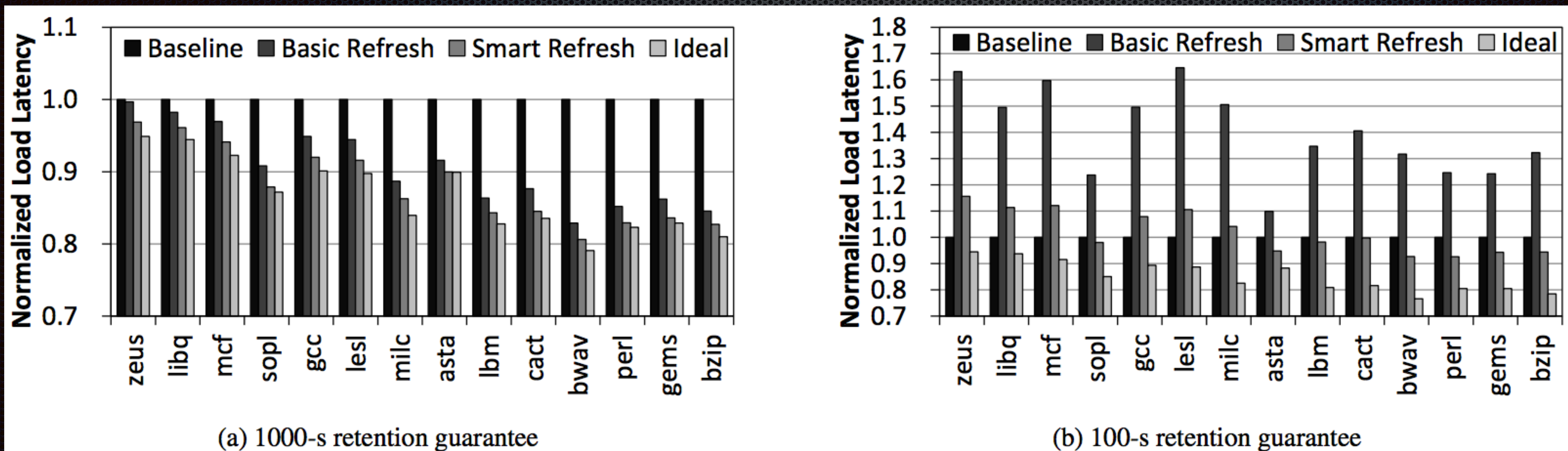


Figure 14: Benefits of Dual-Retention PCM for load latency

Discussion