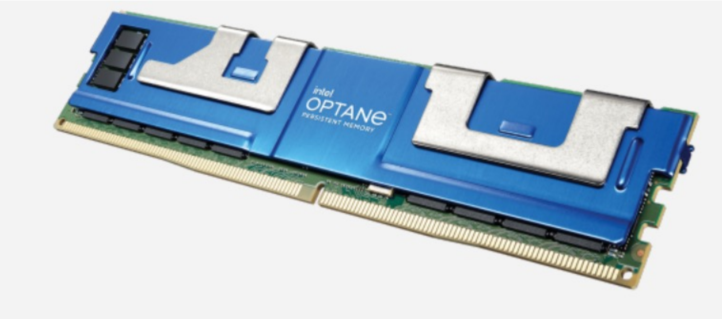
SoftPM: Software Persistent Memory

Yuanchao Xu, Wei Xu, Kimberly Keeton, David E. Culler

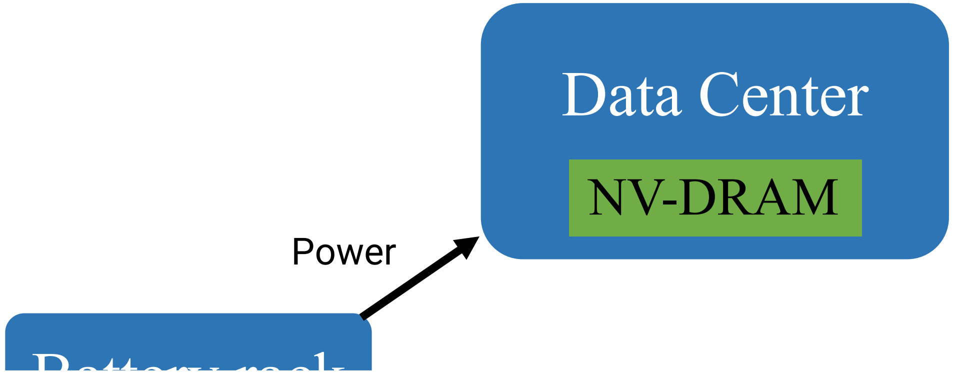
NC STATE UNIVERSITY

Google

Another Way to Embrace Directly Addressable Persistence



Intel Opta



Can NV-DRAM accelerate the adoption of **HardPM?**
Can NV-DRAM perform better than HardPM?

HardPM hasn’t becor
since the hardware revolution in 2014

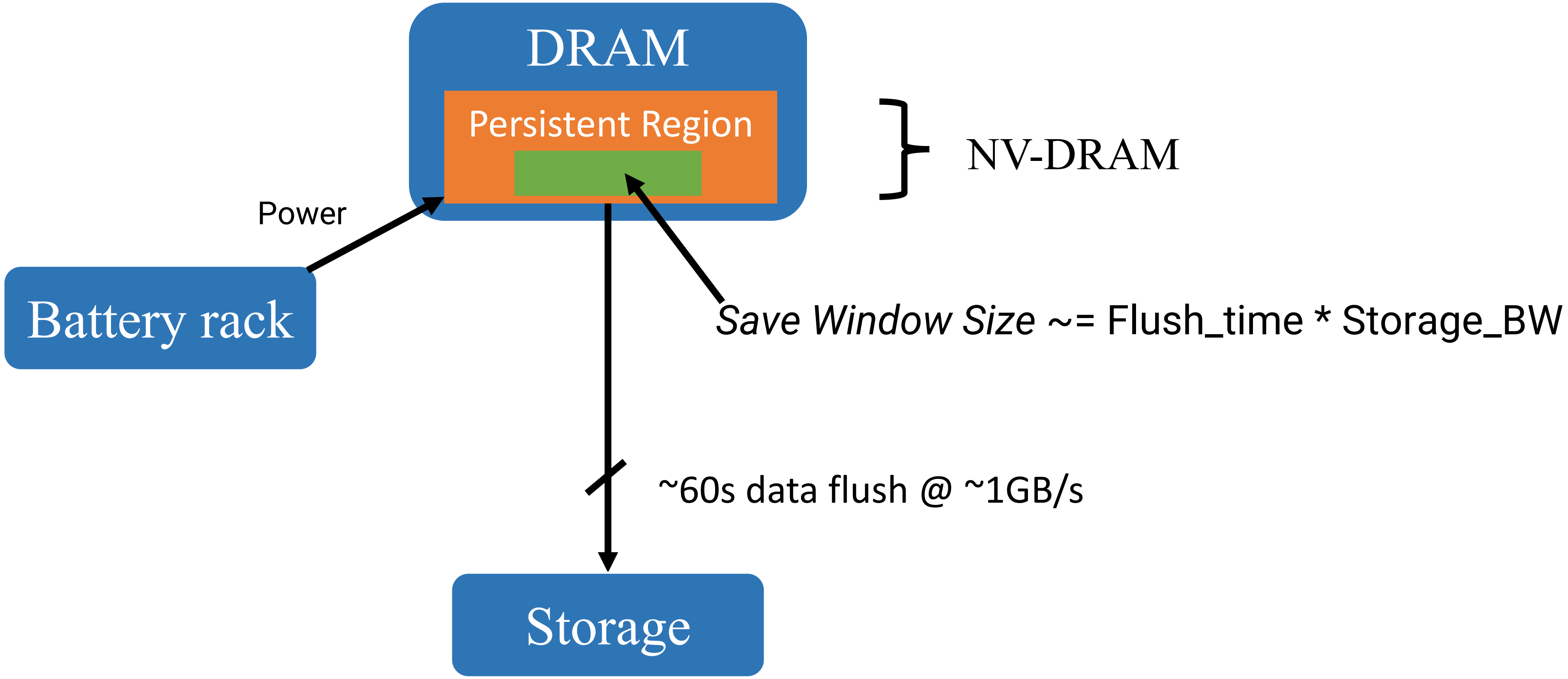
- Lower performance than DRAM
- System software redesigns
- Application redesigns

Directly addressable persistence

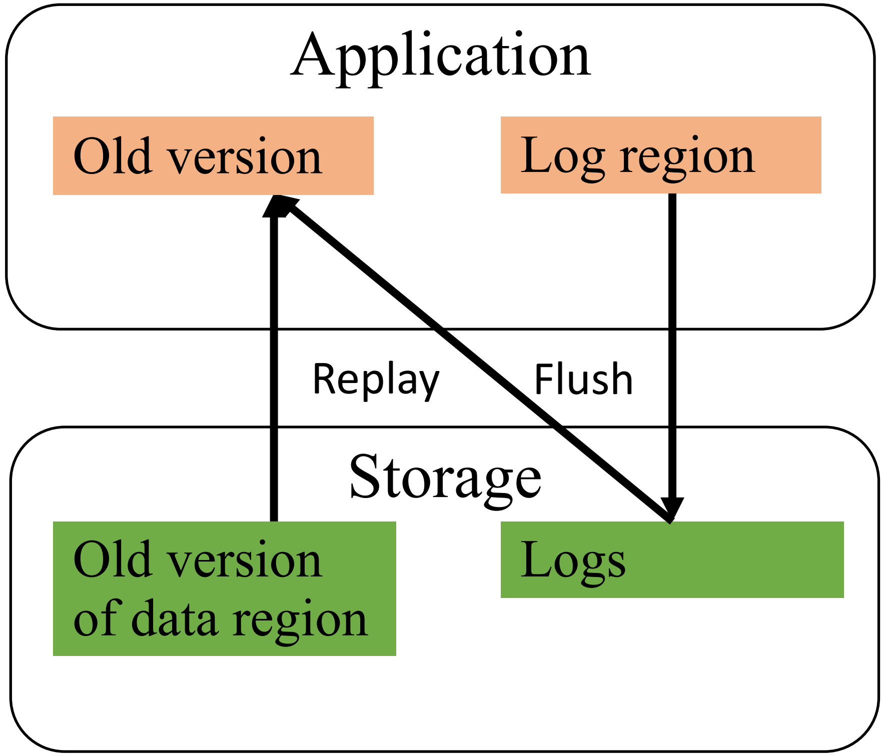
DRAM performance

Limited capacity *

NV-DRAM Limitation: Bounded Save Window

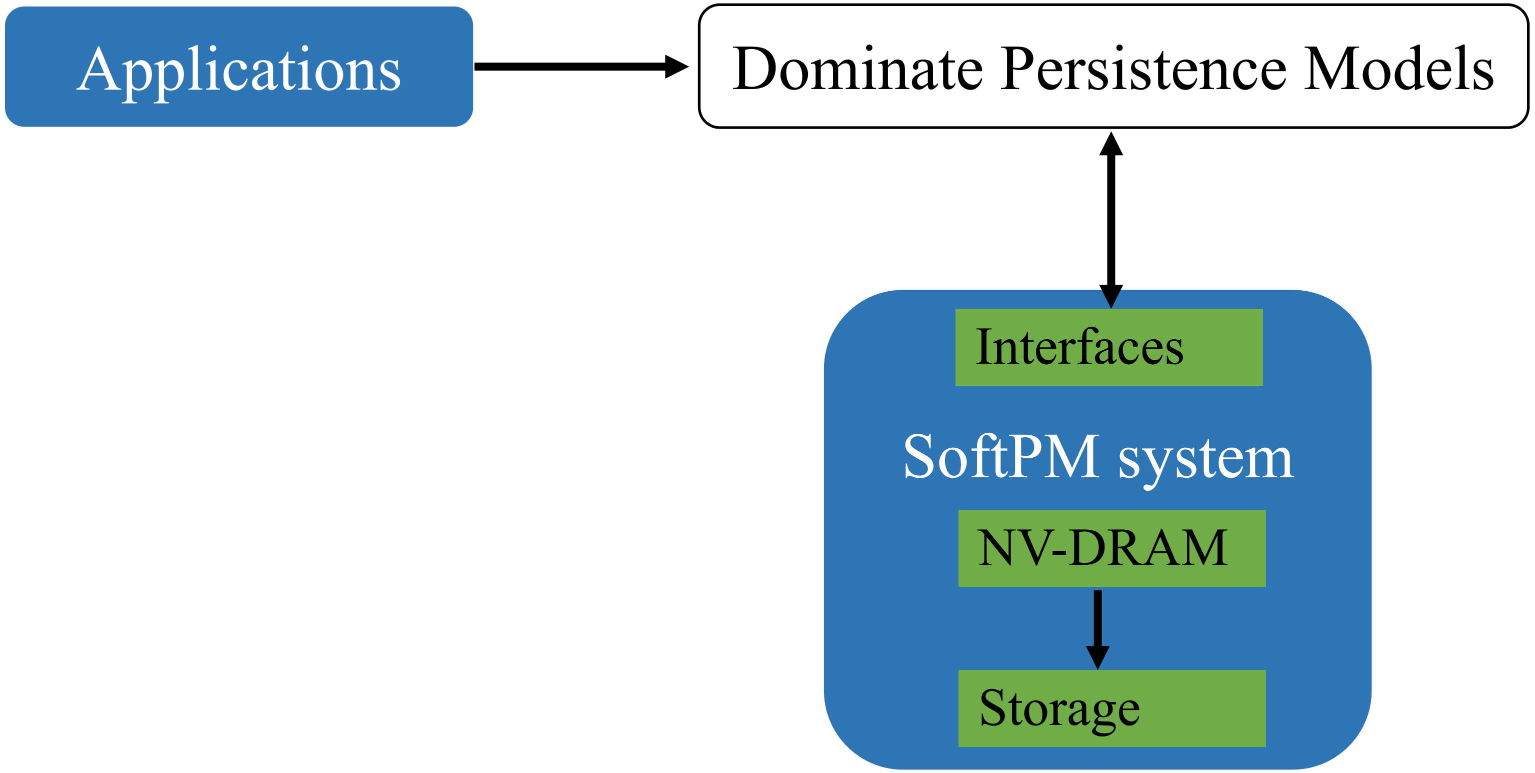


Applications are Architected with Persistence Models



Necessary for persistence

Software Persistent Memory Methodology



Applications

Dominate Persistence Models

Interfaces

SoftPM system

NV-DRAM

Storage

Agenda

- Application persistence models
- SoftPM design and implementation
- Multiple versions of Redis
- Evaluation

Persistence Models

Static:

Applications fully manage the fixed-size region

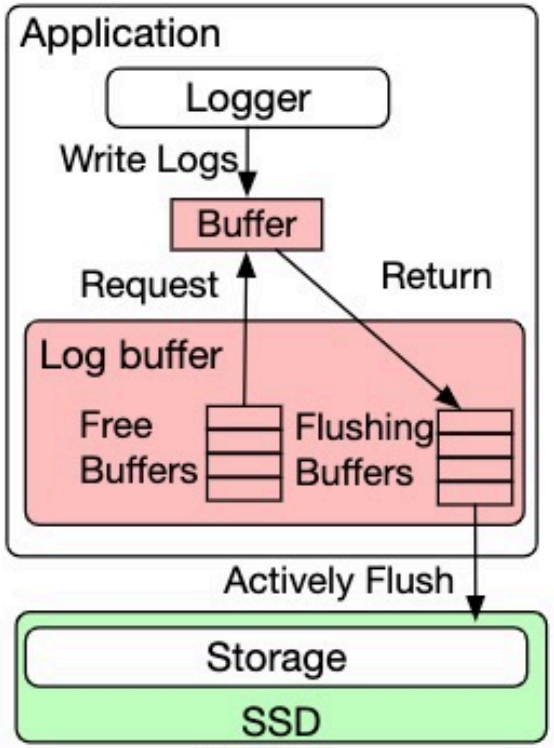
Append-only:

Applications consistently append logs to the in-memory region

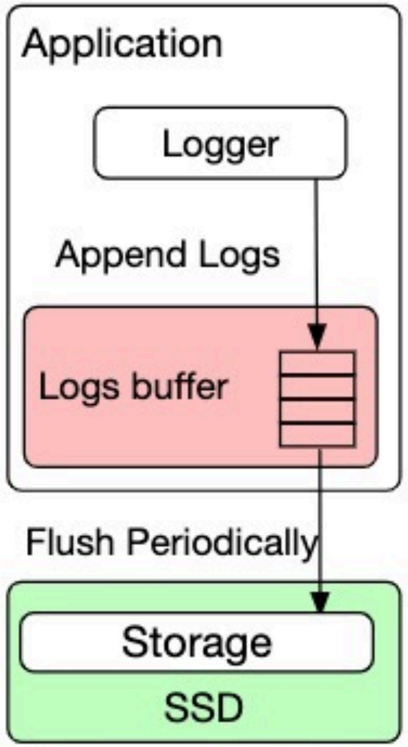
Large in-memory persistence: Data flushed from cache to memory is persistent

🟥 Volatile

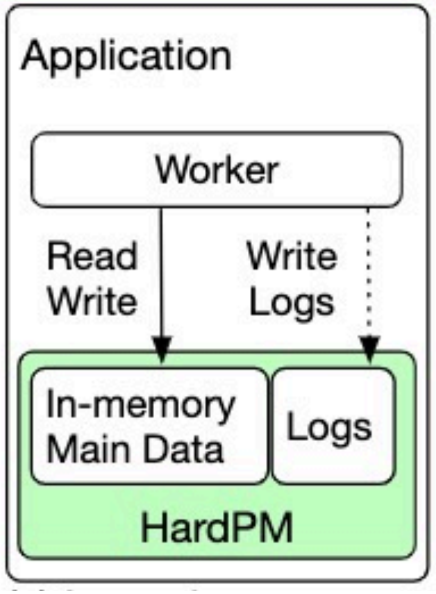
🟩 Persistent



(a) Static[1]



(b) Append-only[2]



(c) Large in-memory

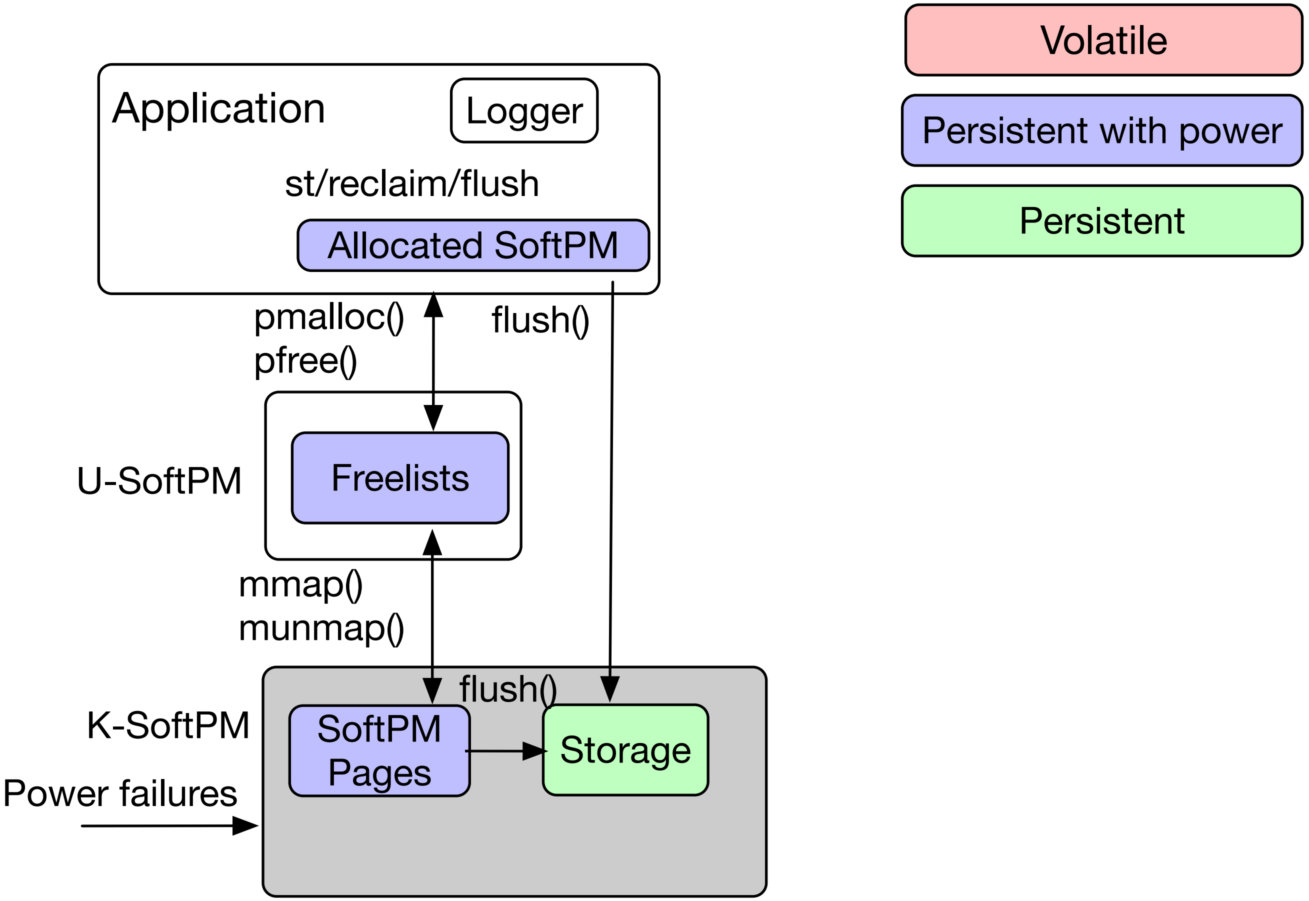
[1] Fast Databases with Fast Durability and Recovery Through Multicore Parallelism, Zheng et al. (OSDI 2014)

[2] Reids. https://redis.io/docs/manual/persistence/

SoftPM Static

Design

- Tag all allocated regions as save window region



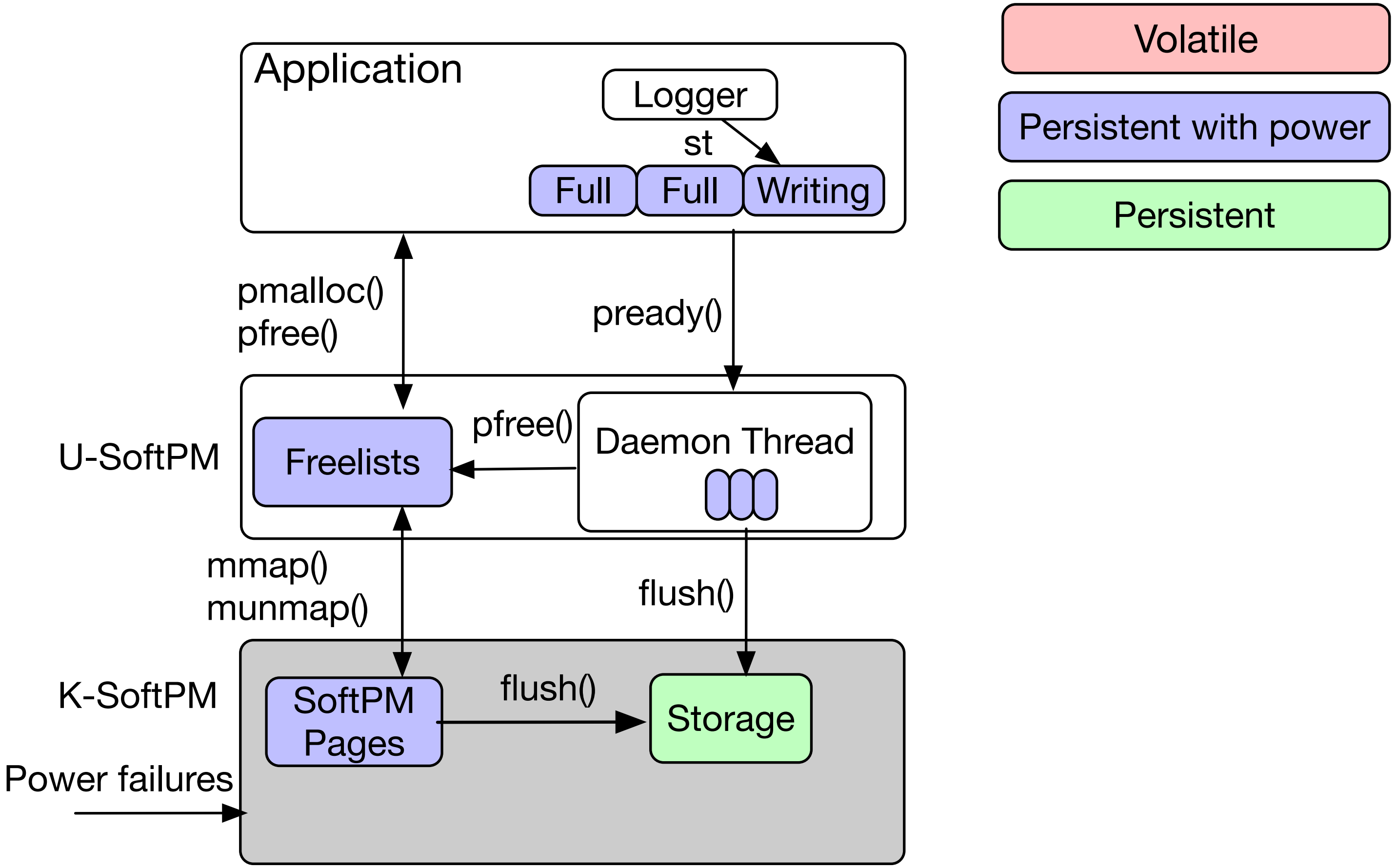
SoftPM User-directed

Characteristics of append-only persistence model

- Appended logs are not overwritten again

Design

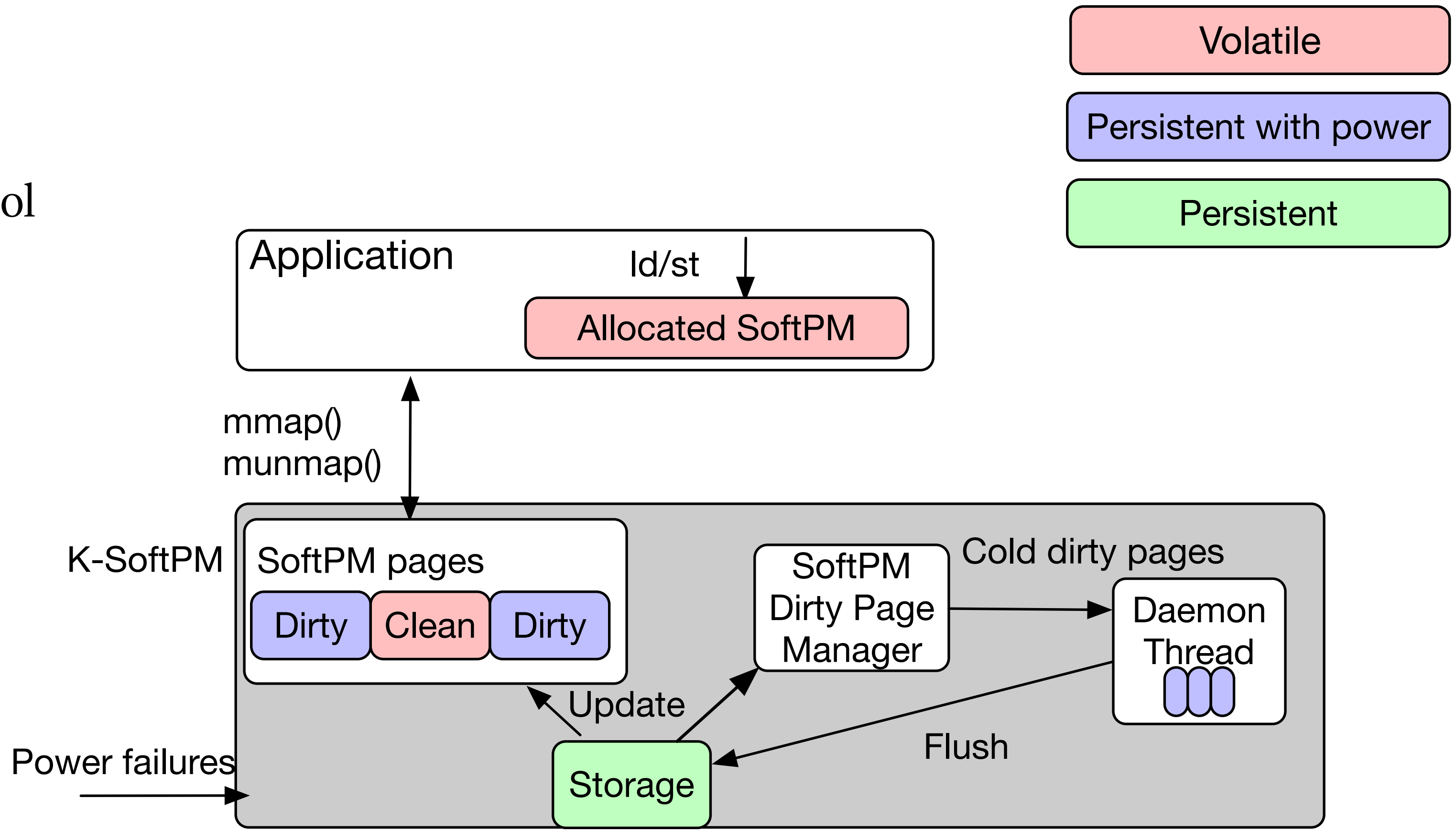
- Background flushing
- Users can deliver read-only hints to the SoftPM system
- Save window size control



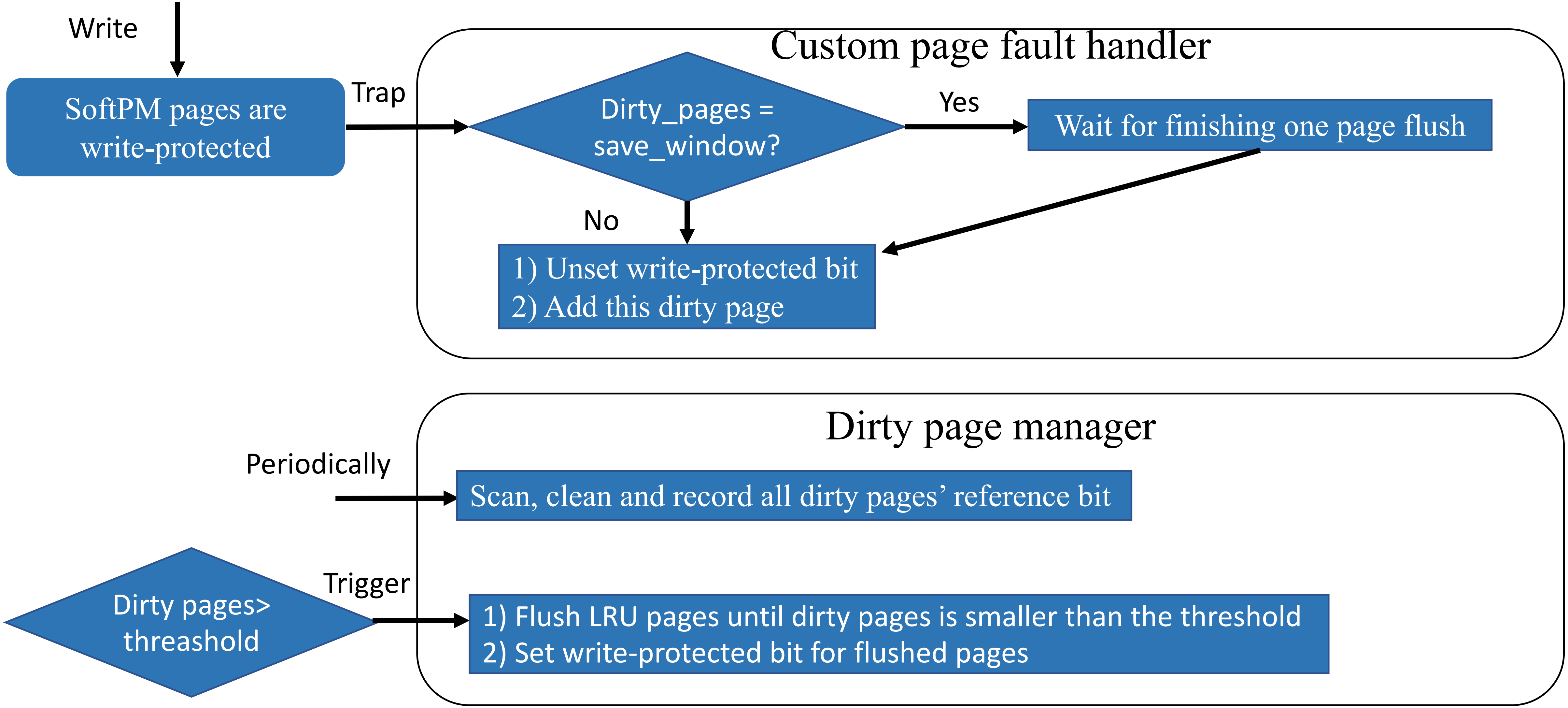
SoftPM Transparent

Design

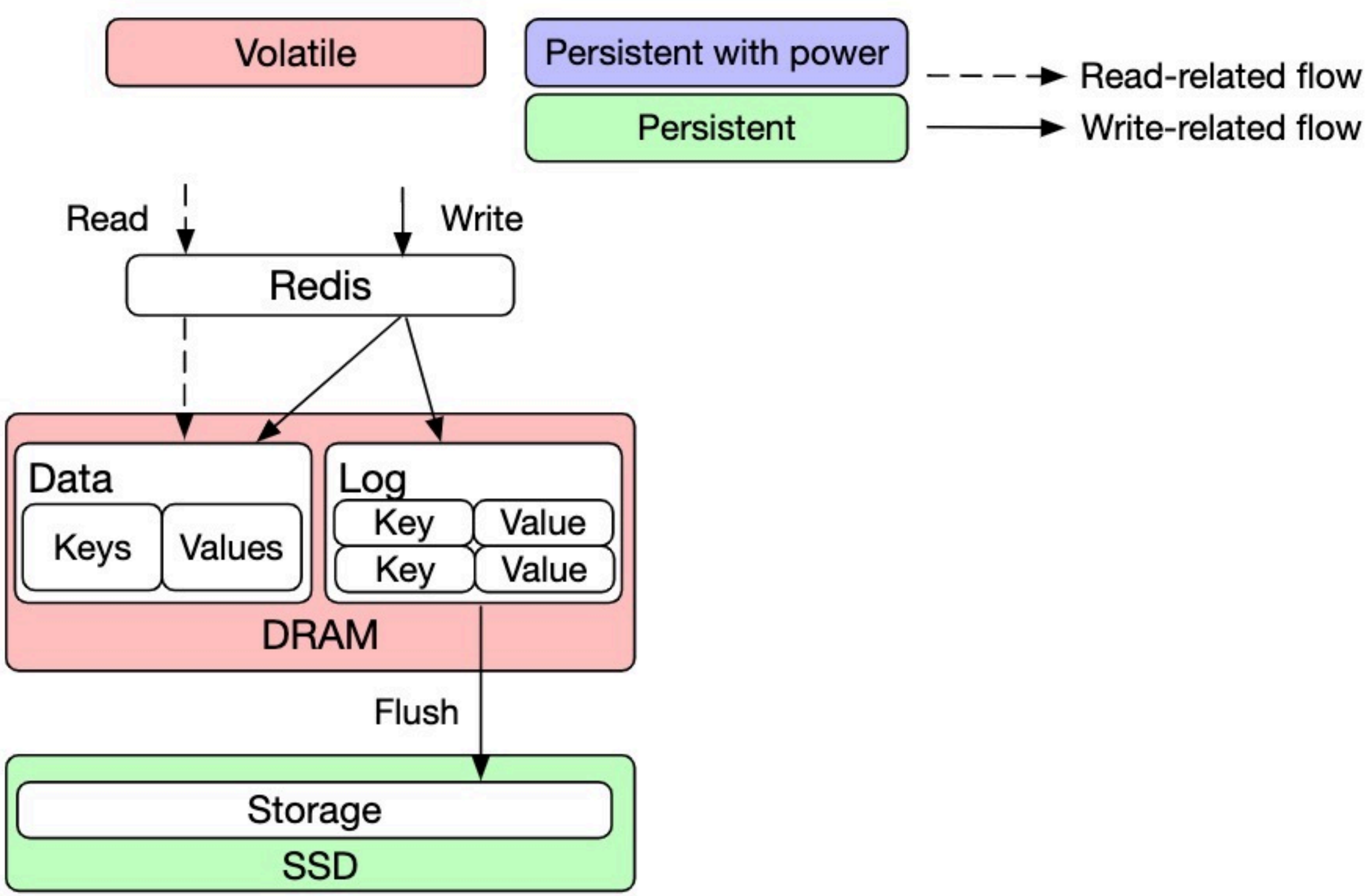
- Dirtiness-aware save window size control
- User-transparent background flushing



SoftPM Transparent Implementation

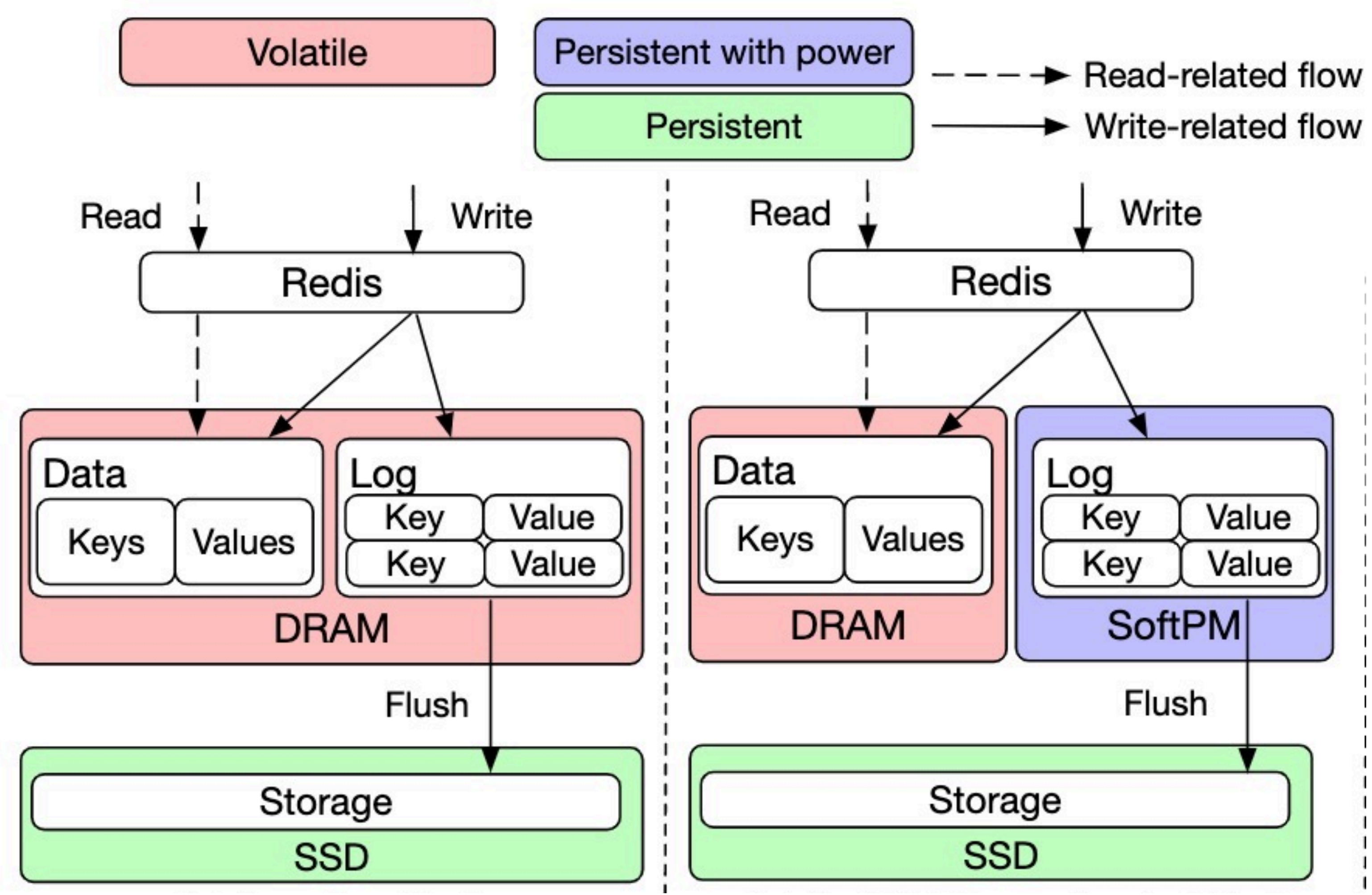

Write
SoftPM pages are write-protected
Trap
Custom page fault handler
Dirty_pages = save_window?
Yes
Wait for finishing one page flush
No
1) Unset write-protected bit
2) Add this dirty page
Dirty page manager
Periodically
Scan, clean and record all dirty pages' reference bit
Trigger
Dirty pages> threashold
1) Flush LRU pages until dirty pages is smaller than the threshold
2) Set write-protected bit for flushed pages

Multiple Versions of Redis



(a) Baseline

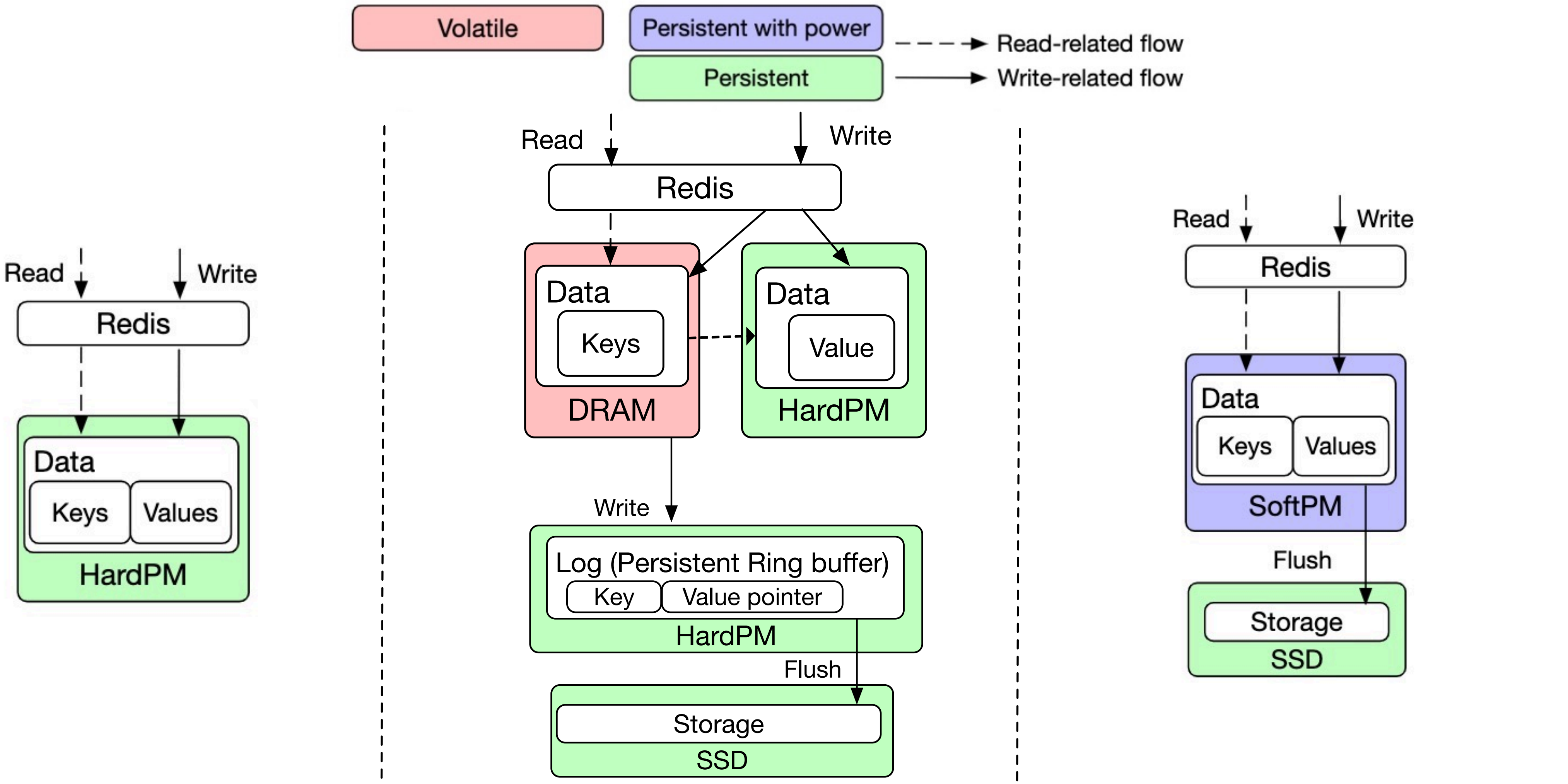
Multiple Versions of Redis



(a) Baseline

(b) SoftPM User-directed

Multiple Versions of Redis

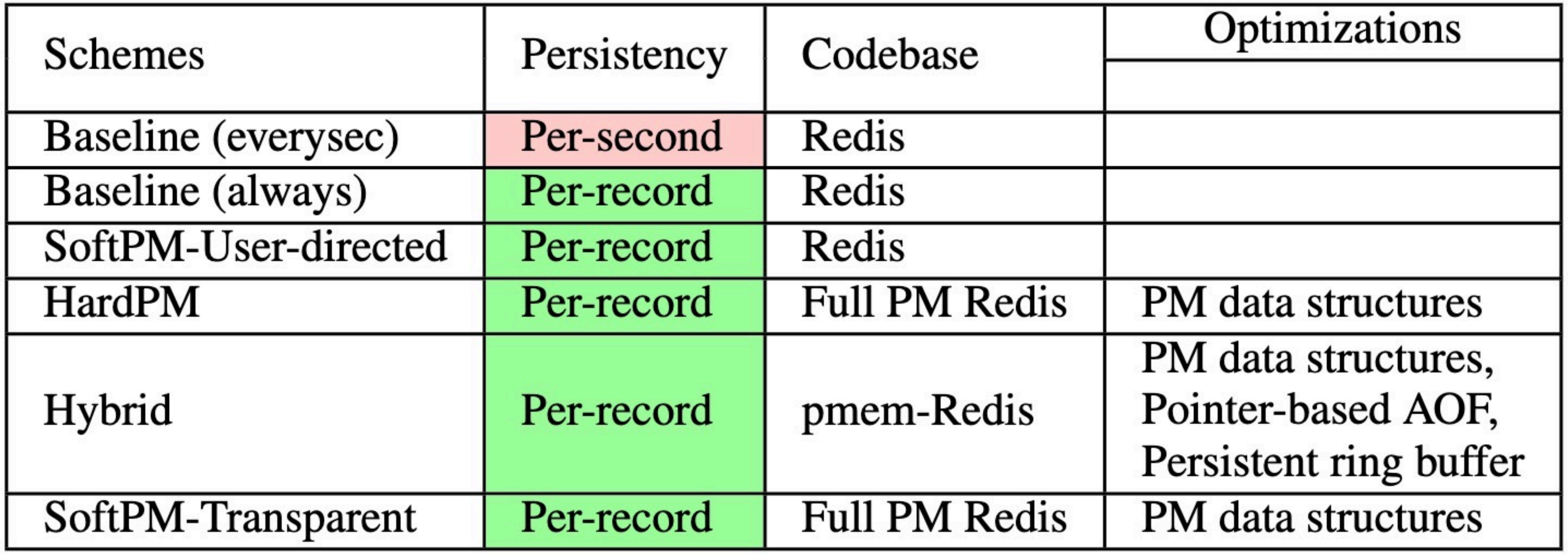


(c) HardPM

(d) Hybrid

(e) SoftPM Transparent

Summary of Multiple Versions of Redis



Schemes	Persistency	Codebase	Optimizations
Baseline (everysec)	Per-second	Redis	
Baseline (always)	Per-record	Redis	
SoftPM-User-directed	Per-record	Redis	
HardPM	Per-record	Full PM Redis	PM data structures
Hybrid	Per-record	pmem-Redis	PM data structures, Pointer-based AOF, Persistent ring buffer
SoftPM-Transparent	Per-record	Full PM Redis	PM data structures

Evaluation

Experiment setup

- 28-core Intel Xeon Platinum 8273CL@2.20GHz
- 6-channel HardPM, 6-channel DRAM and a SSD
- All server and client threads are running on different cores in one socket

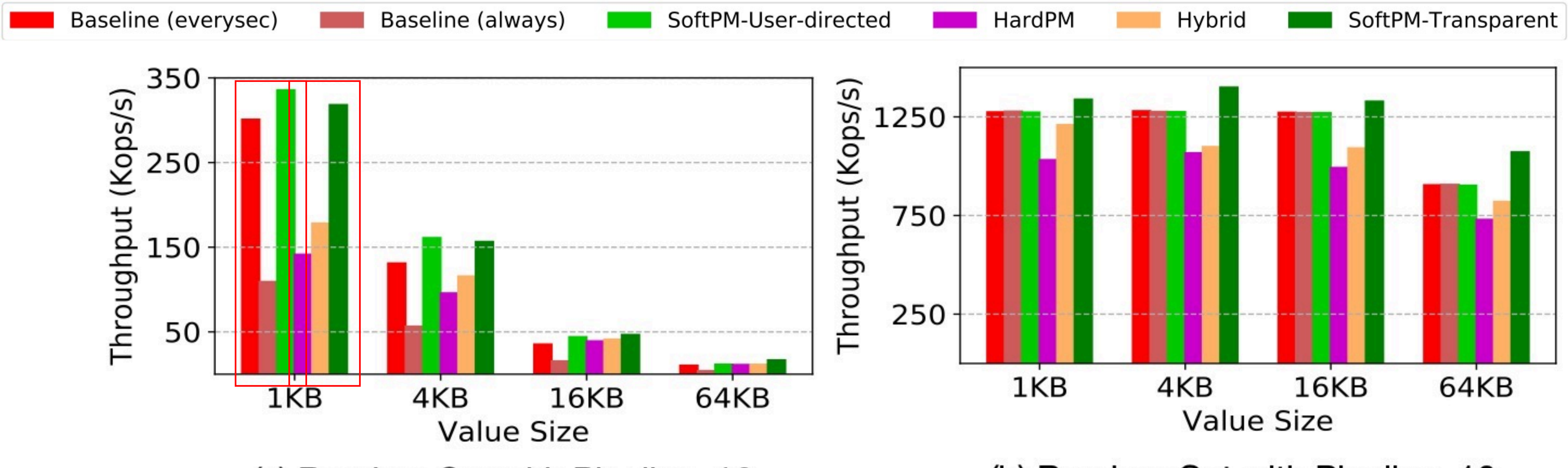
Unlimited save window size

- Microbenchmark
- YCSB

Limited save window size (30 GB SoftPM data, 3G (10%)-30G(100%) save window)

- YCSB

Microbenchmark Evaluation

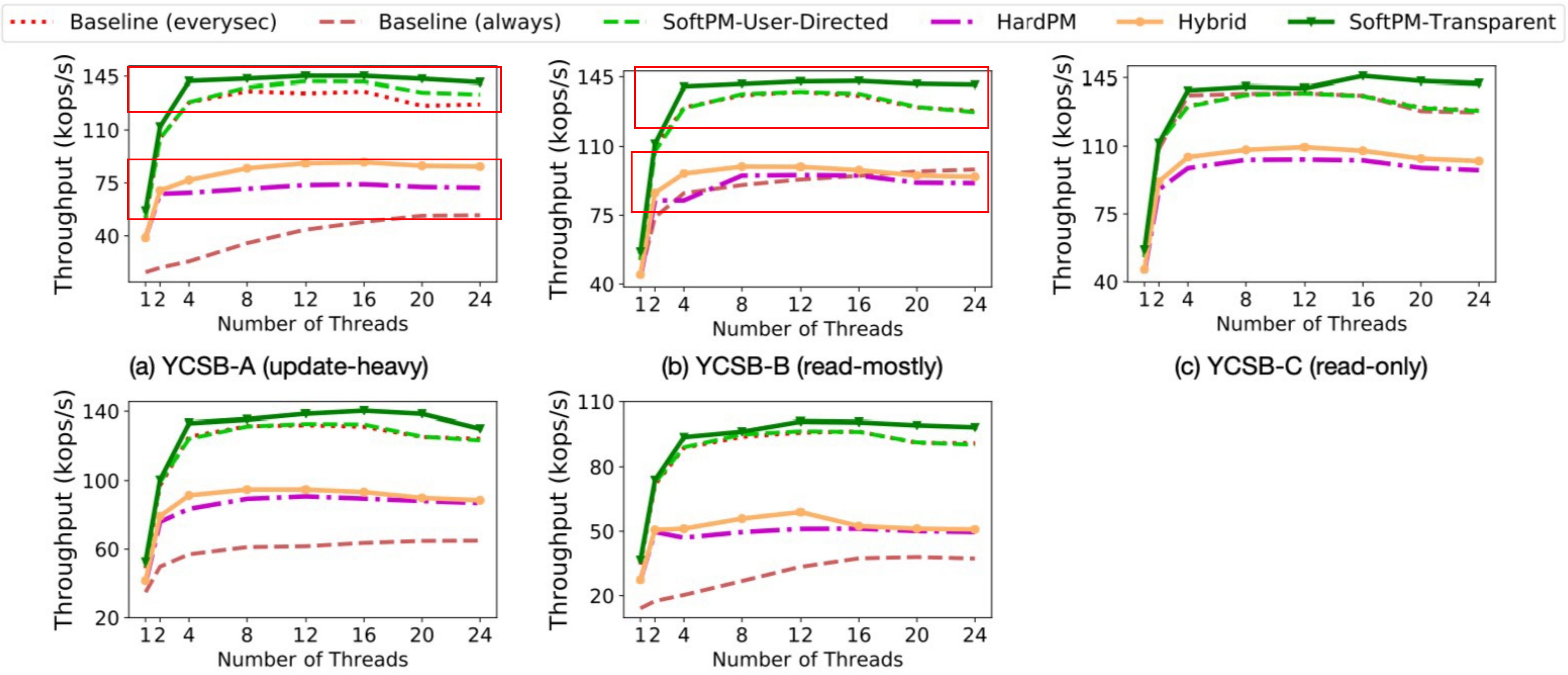


(a) Random Set with Pipeline=16

(b) Random Get with Pipeline=16

SoftPM Transparent Improves 7%-88% throughput over the Hybrid

YCSB Evaluation



(a) YCSB-A (update-heavy)

(b) YCSB-B (read-mostly)

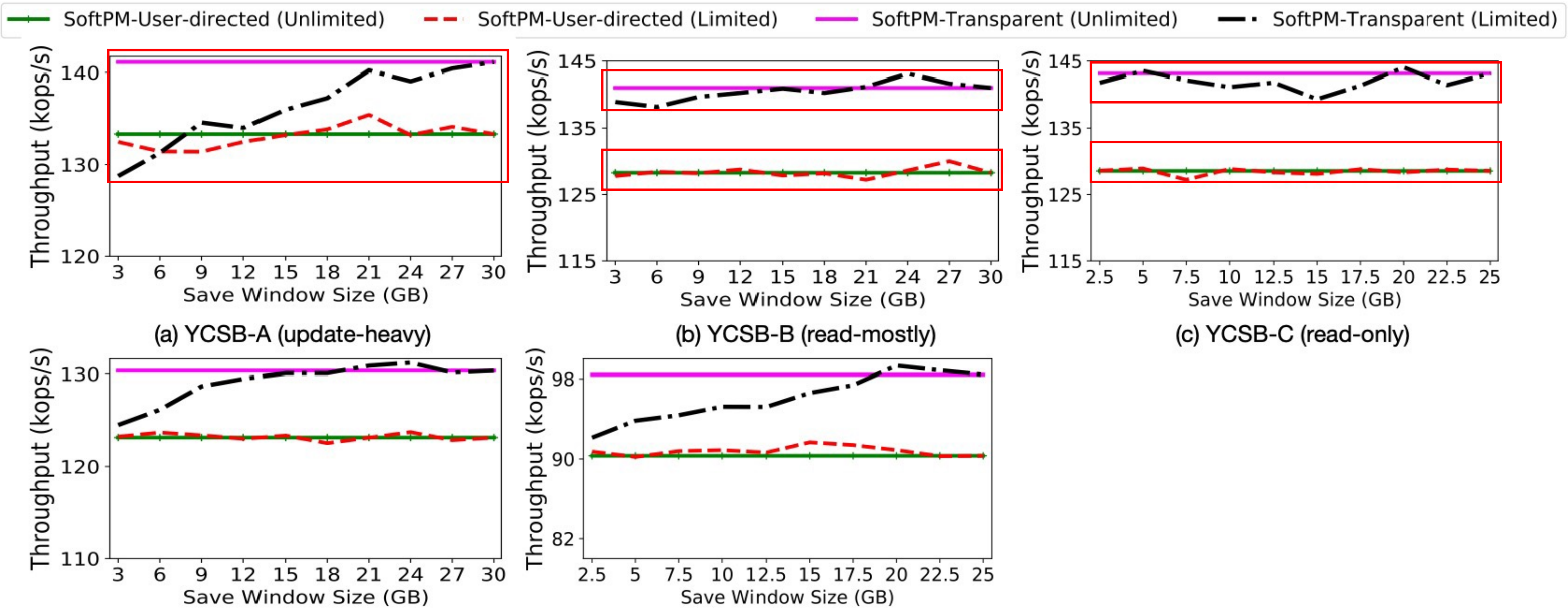
(c) YCSB-C (read-only)

(d) YCSB-D (read-latest)

(e) YCSB-F (read-modify-write)

SoftPM Transparent Improves 38%-93% throughput over the Hybrid

YCSB Evaluation with Limited Save window



(a) YCSB-A (update-heavy)

(b) YCSB-B (read-mostly)

(c) YCSB-C (read-only)

(d) YCSB-D (read-latest)

(e) YCSB-F (read-modify-write)

SoftPM incurs at most 9% overhead with 10% save window size

Conclusion

SoftPM is potentially available using technology already provisioned in data centers today

SoftPM improves 38%-93% throughput over the well-optimized Hybrid (DRAM and HardPM) versions.

- Real DRAM performance memory
- Background flushing
- Cleaner designs

SoftPM also sets the performance bar that future HardPM and its ecosystem will need to beat.