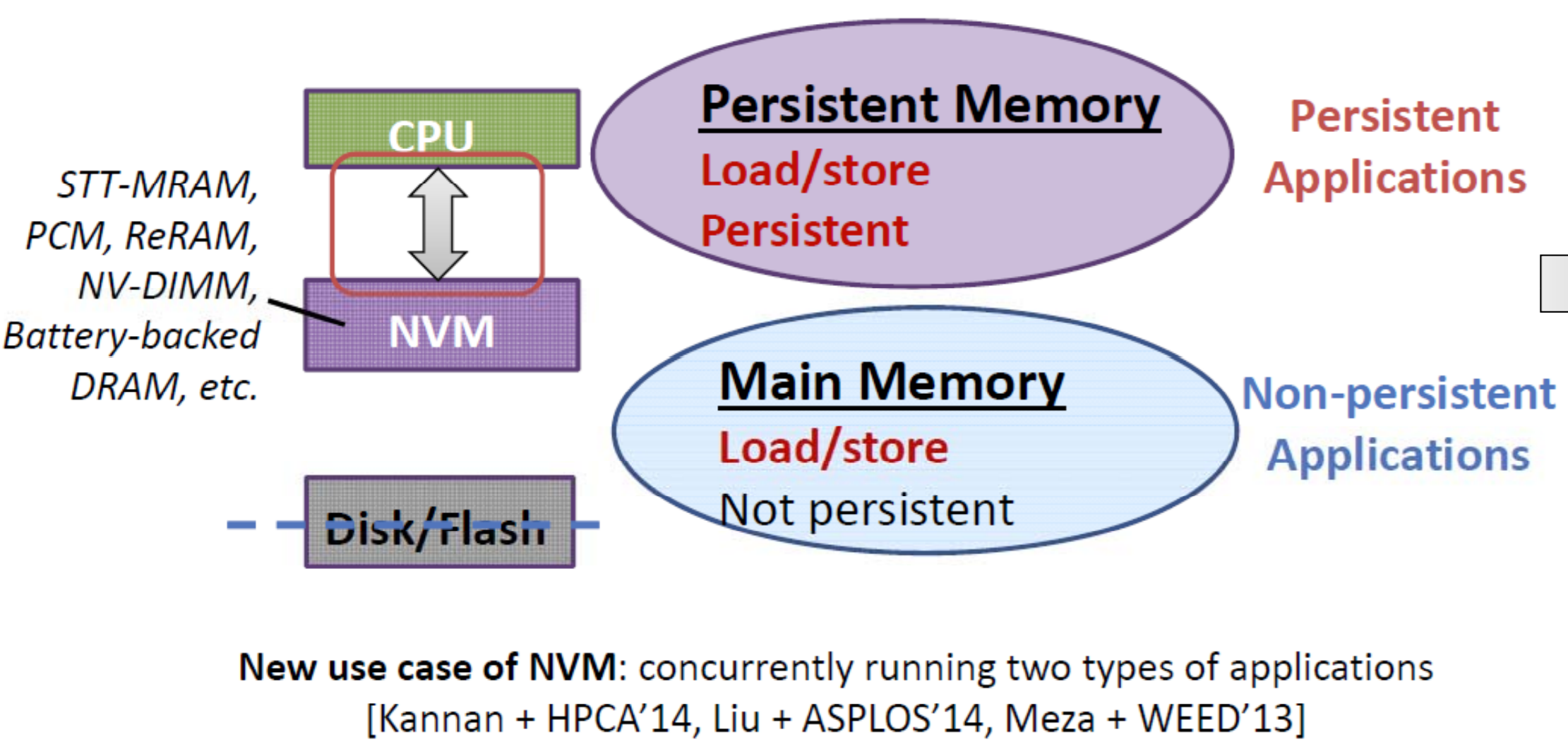


FIRM: Fair and High-Performance Memory Control for Persistent Memory Systems

Jishen Zhao, Onur Mutlu, Yuan Xie

New Use Case of NVMs



Why Another Memory Control Scheme?



Memory Accesses in Persistent Memory Systems

Reads / Writes / Persistent Writes

Assumptions Made by Previous Memory Schedulers

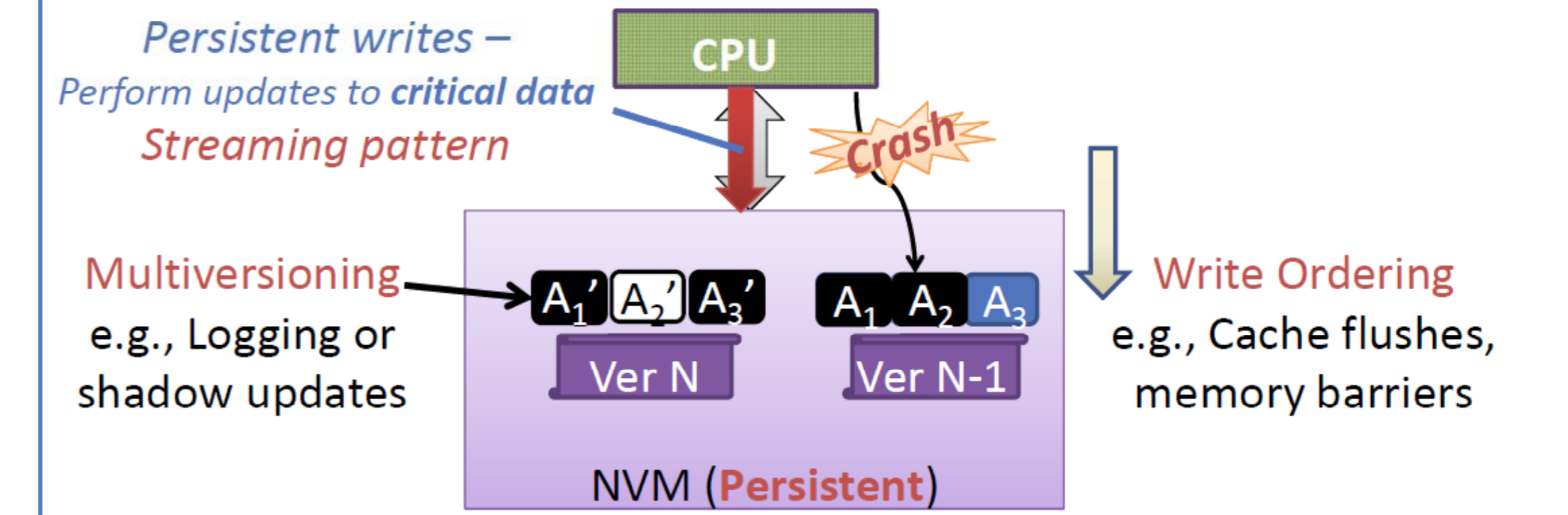
Reads are on the critical path of application execution

Applications are usually read-intensive

Prioritize reads over writes

Delay writes until they fill up the write queue

Persistent Writes



Writes are also on the critical execution path

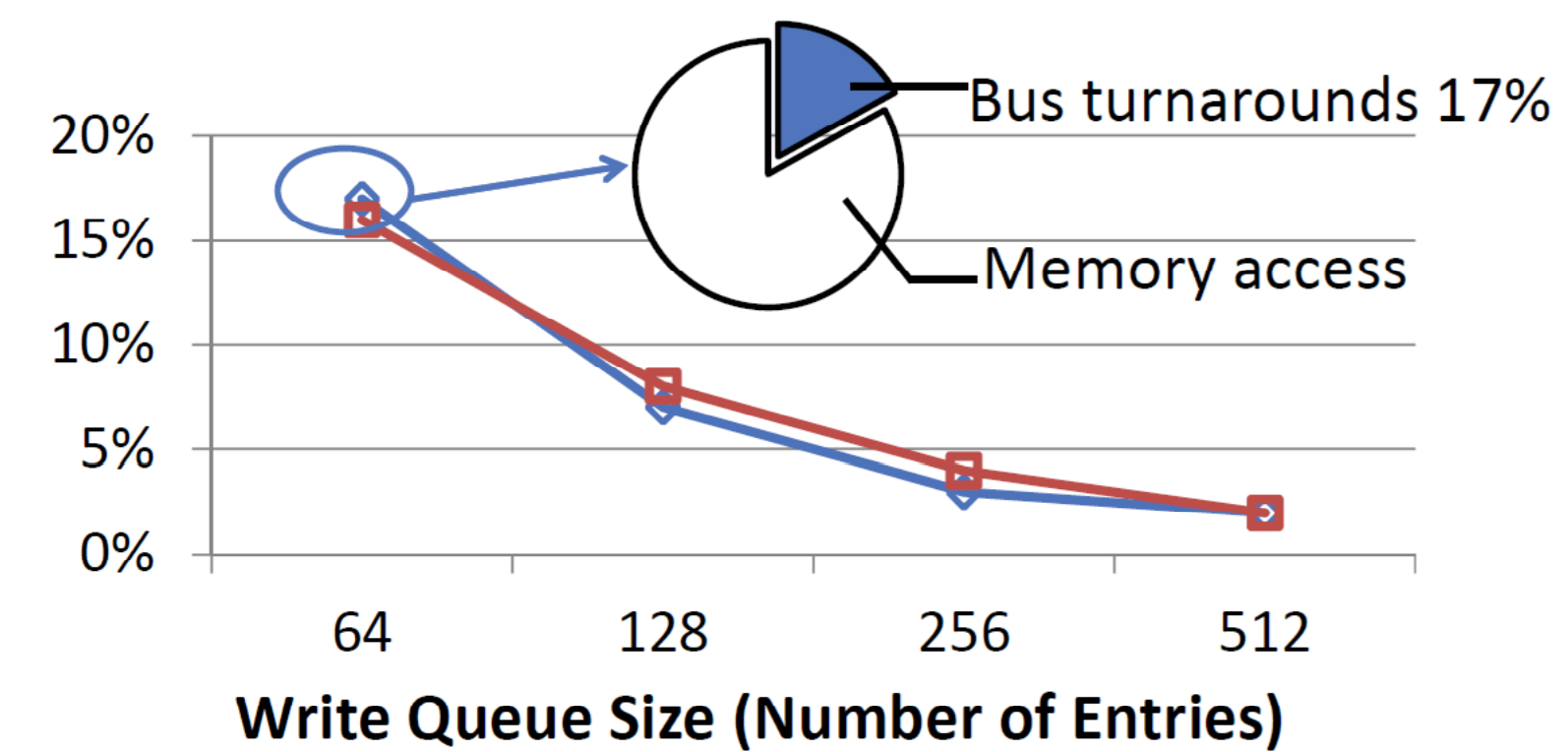
Persistent applications can be write-intensive

Unfairness

Performance Degradation

Bus Turnaround Overhead

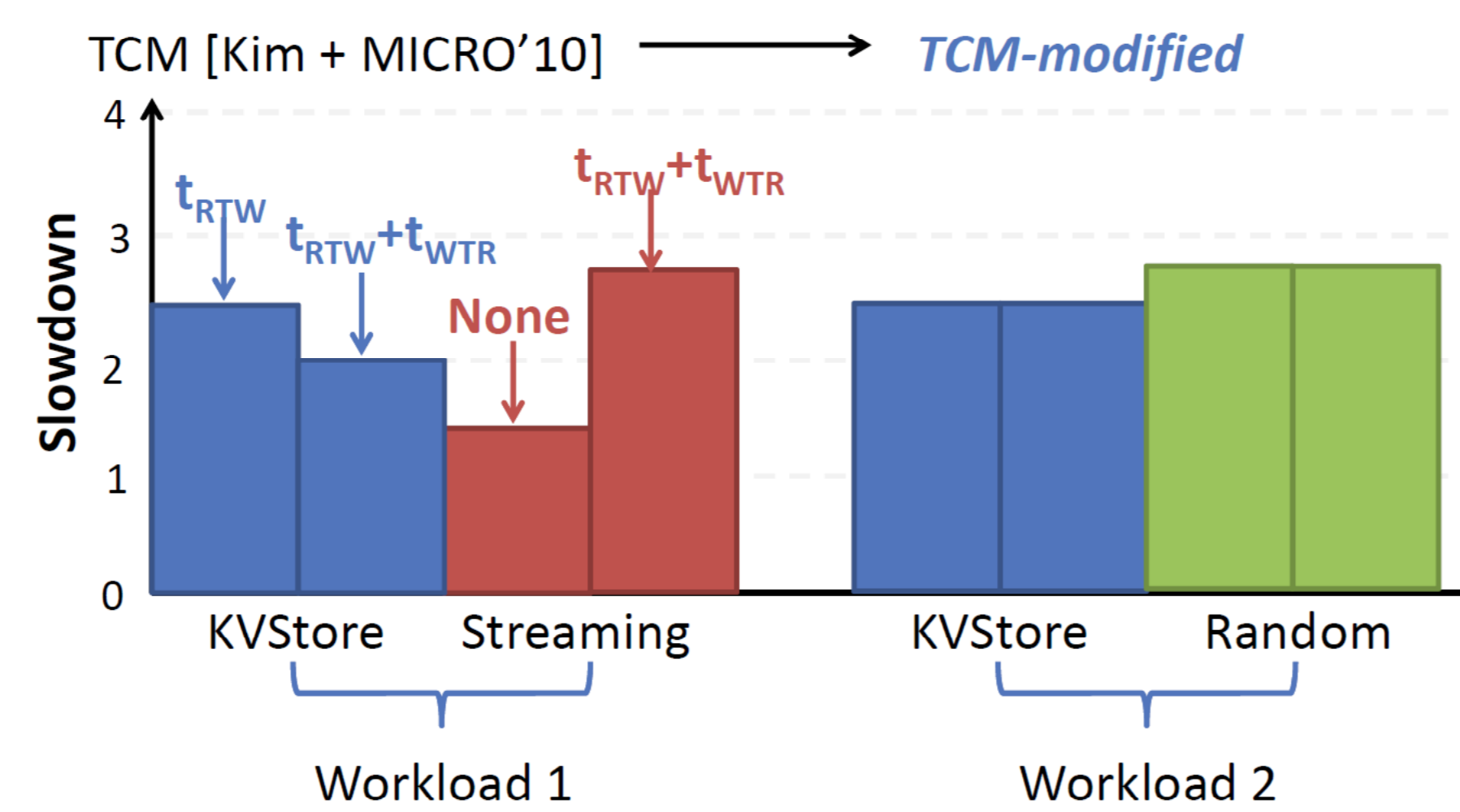
Fraction of memory access cycles wasted on bus turnarounds



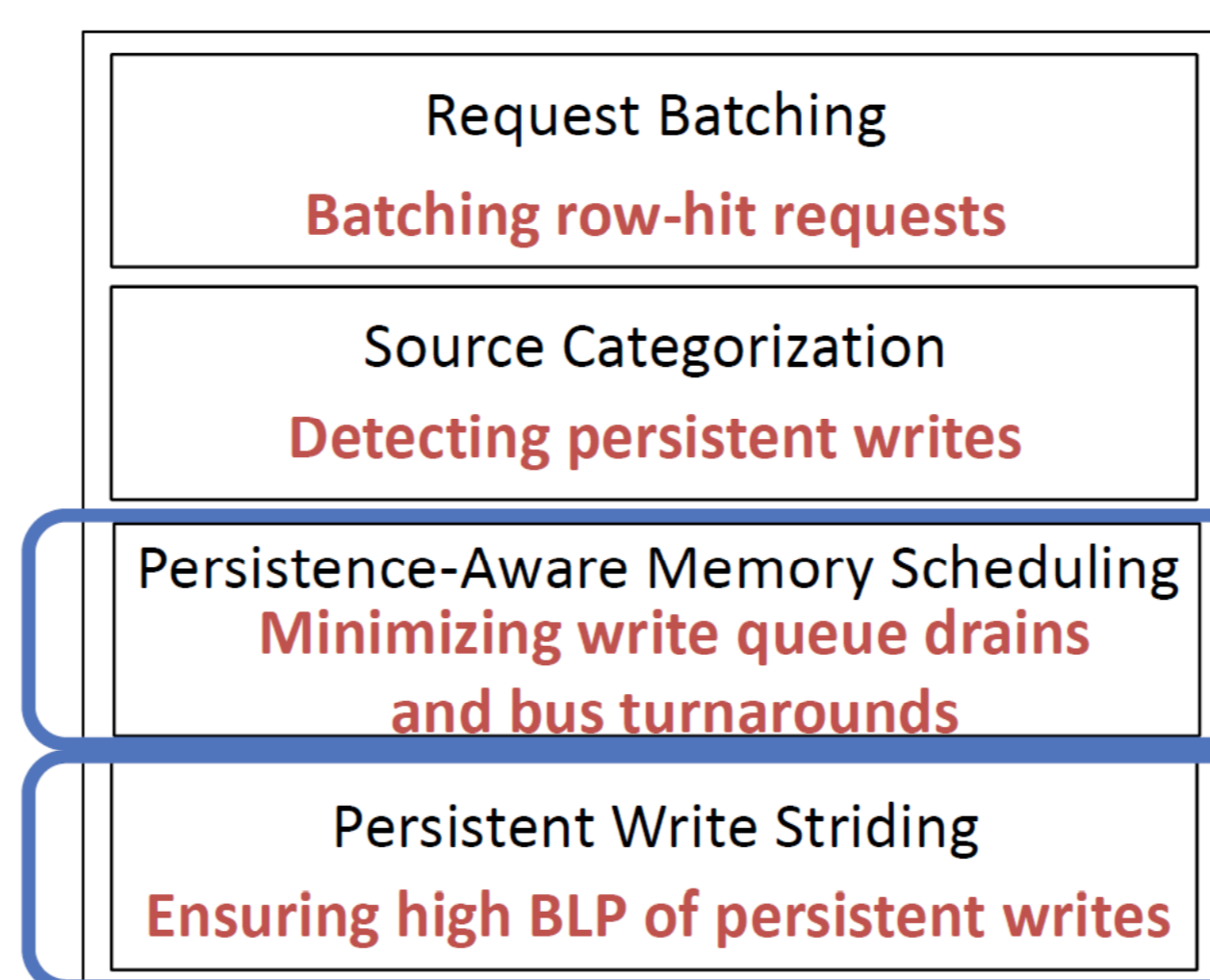
Workloads

- Streaming + Persistent key-value store (KVStore)
- Random + KVStore

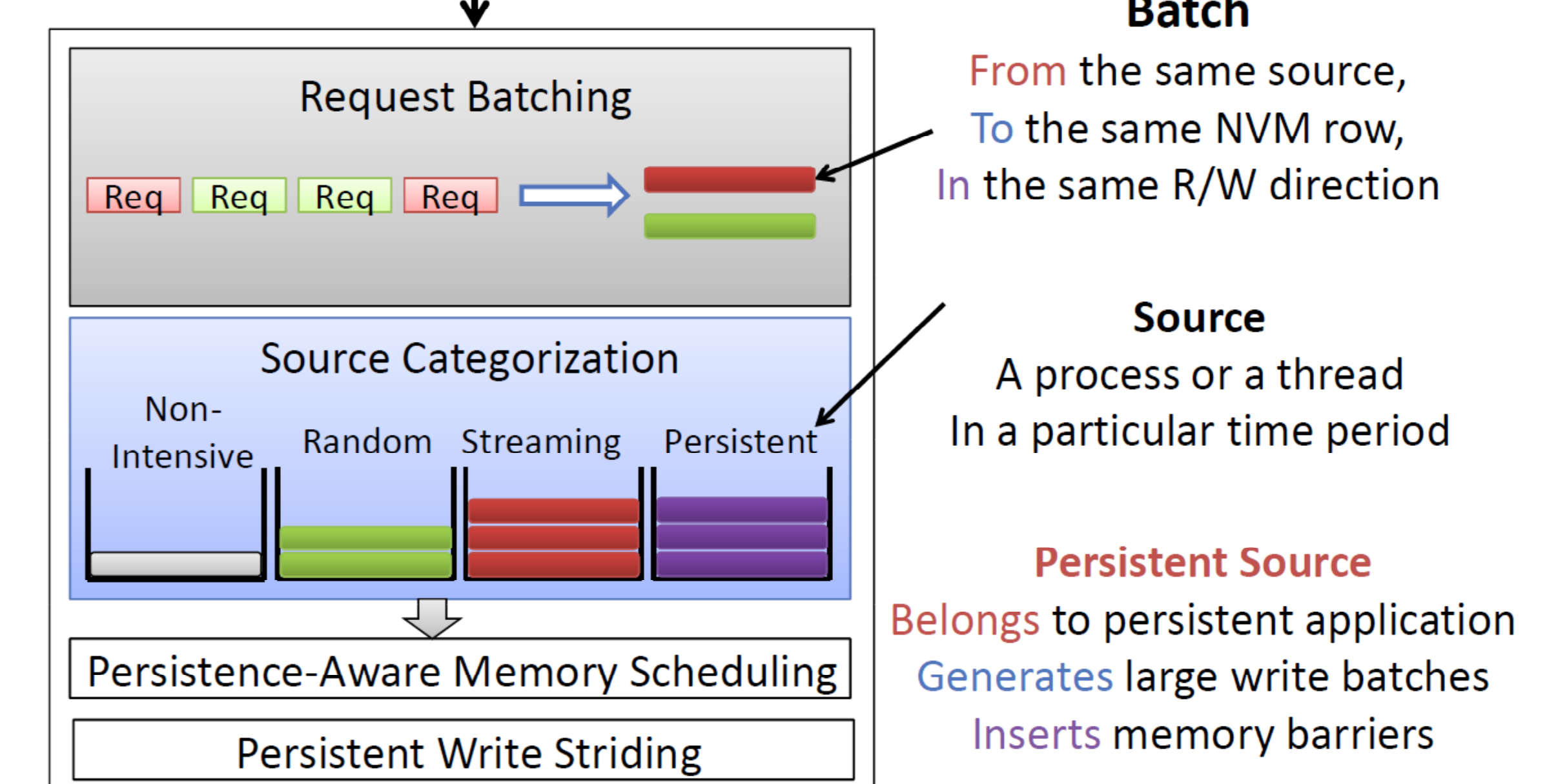
Simple Extension of Conventional Memory Schedulers



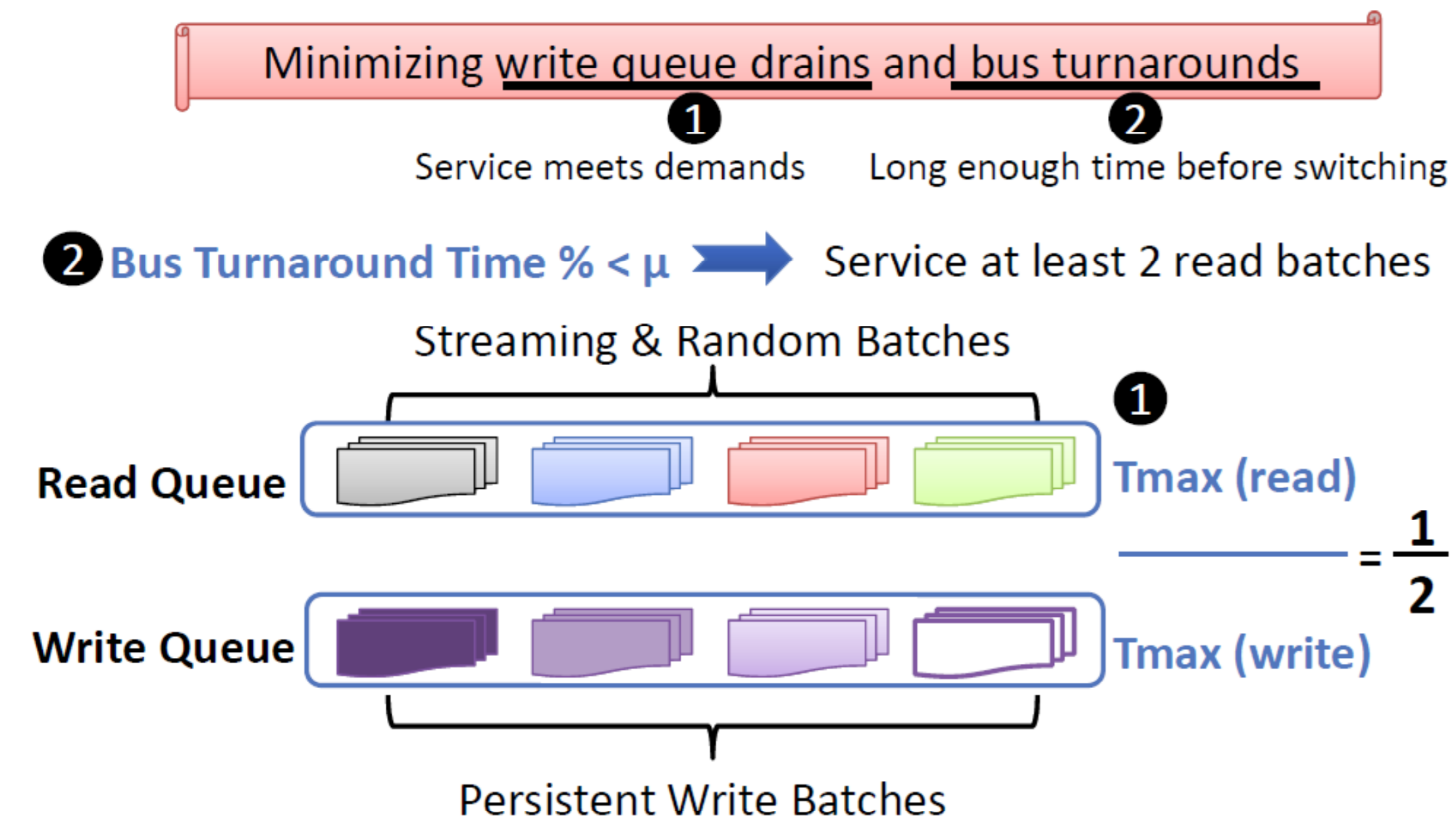
FIRM Components



Memory Requests



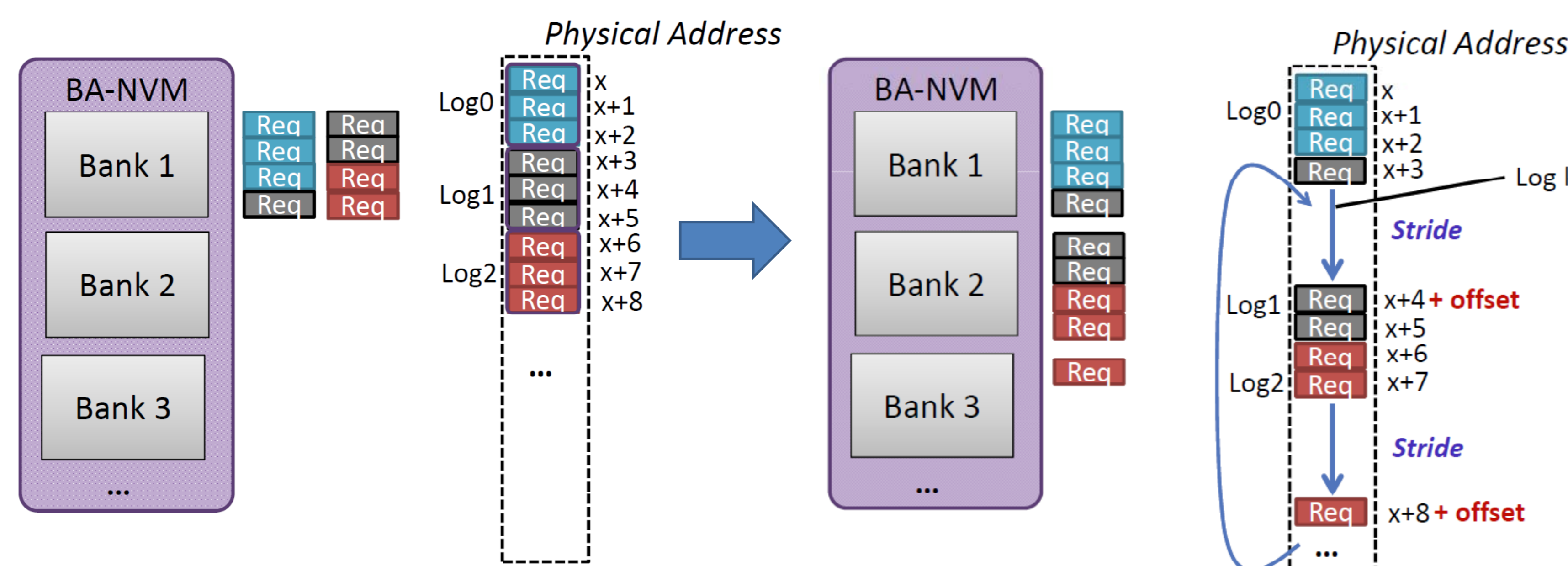
Persistence-aware Memory Scheduling



Persistent Write Striding

Ensuring high BLP of persistent writes

A Log (Spans Multiple Banks)



Fairness and Performance

