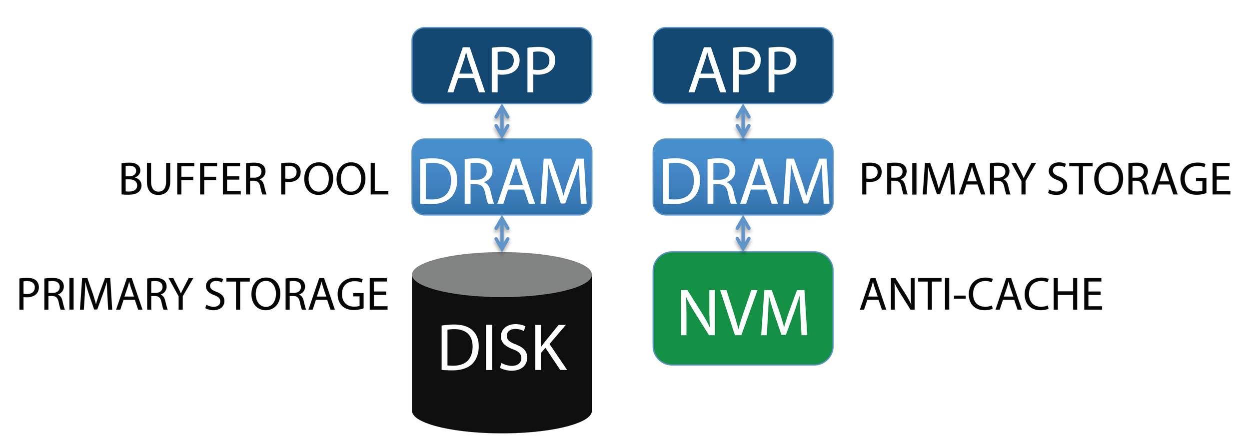
OLTP on the NVM SDV: YMMV

Joy Arulraj (CMU), Justin DeBrabant (Brown), Andrew Pavlo (CMU), Michael Stonebraker (MIT), Col. Stan Zdonik (Brown)

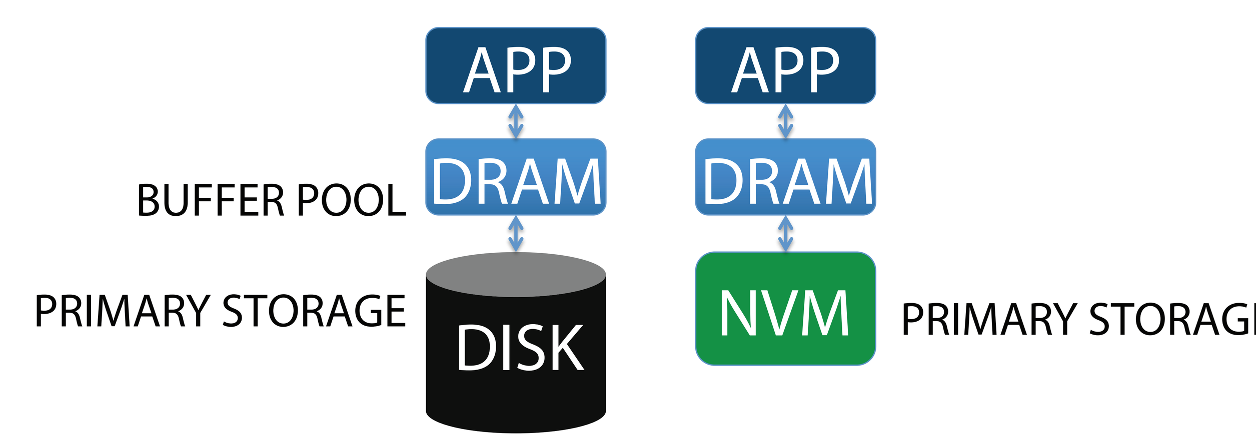
ANTI-CACHING ON NVM

- Recovery mechanism
  - Snapshot of table and index data, including anti-caching structures, persisted on NVM
  - Command-log (for redo) persisted on NVM
  - Recovery restores state from latest snapshot and replays transactions in command log
- Implementation
  - Asynchronous fine-grained eviction of coldest tuples from DRAM to NVM (LRU policy)
  - Data exists in exactly one location
  - Non-blocking data fetches on demand



DIRECT NVM

- Recovery mechanism
  - Table and index data persisted directly on NVM
  - No need for command logging
  - Recovery undoes uncommitted transactions
- Implementation
  - MMAP-based storage manager directly uses persistent memory file system
  - STL allocator based on MMAP storage manager
  - Table, Index and Pool data persisted directly on NVM

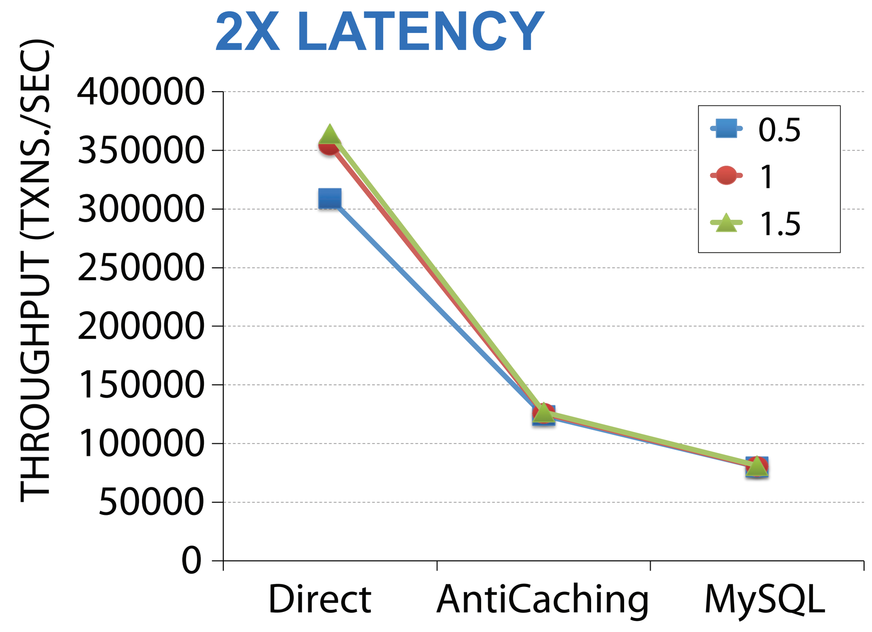


EXPERIMENTAL RESULTS

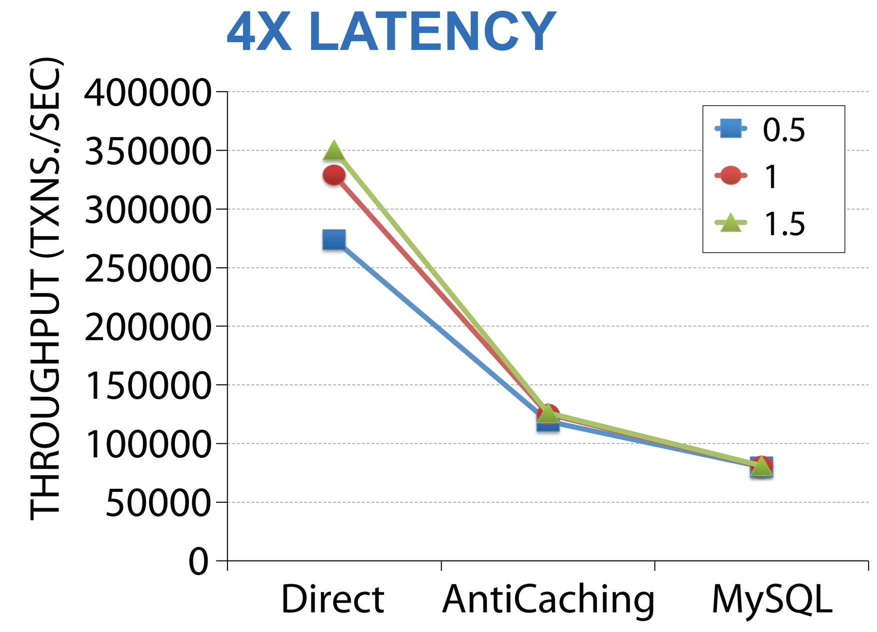
SETUP

- Intel NVM Emulator
  - Instrumented motherboard emulates NVM latency
  - 62 GB DRAM with tunable latency
- Persistent Memory File System
  - Efficient mmap interface to persistent memory
  - Internally uses CPU load/store instructions
- YCSB Benchmark
  - Zipfian skew in record accesses
  - Update Heavy (50% Updates, 50% Reads)

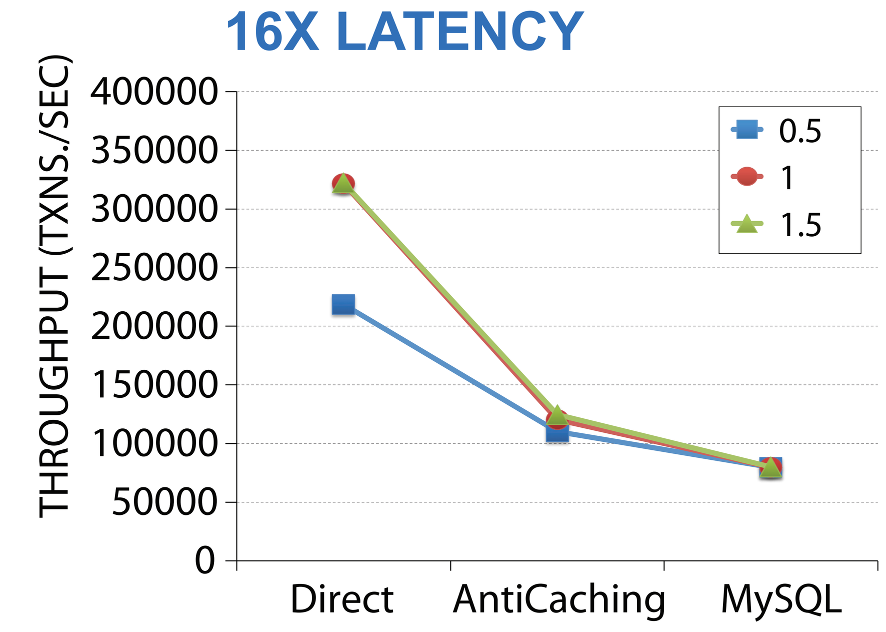
2X LATENCY



4X LATENCY



16X LATENCY



FINDINGS

- Anti-Caching on NVM
  - 1.6X improvement for skewed workloads over disk-based architecture
  - Better utilization of memory hierarchy
- Direct NVM
  - 4.5X improvement for skewed workloads over disk-based architecture
  - Throughput constrained by msync overhead

FUTURE WORK

- Anti-Caching on NVM
  - Reduced memory overhead (Bloom filters)
  - Relaxed consistency for OLAP workloads
  - Intelligent eviction strategies
  - Block reorganization
  - Multi-tiered storage
- Direct NVM
  - Need a new design
  - Concurrency control protocol
  - Recovery mechanism