Let's Talk About Storage & Recovery Methods for Non-Volatile Memory Database Systems

Joy Arulraj (CMU), Andrew Pavlo (CMU), Subramanya R. Dulloor (Intel Labs)

Storage Engines for NVM

❖ Non-Volatile Memory (NVM)

	DRAM	FLASH	DISK	NVM
Read Latency	1x	500x	10^5x	2-4x
Write Latency	1x	5000x	10^5x	2-8x
Persistence	✖	✓	✓	✓
Byte-level access	✓	✖	✖	✓
Write endurance	✓	✖	✓	✖

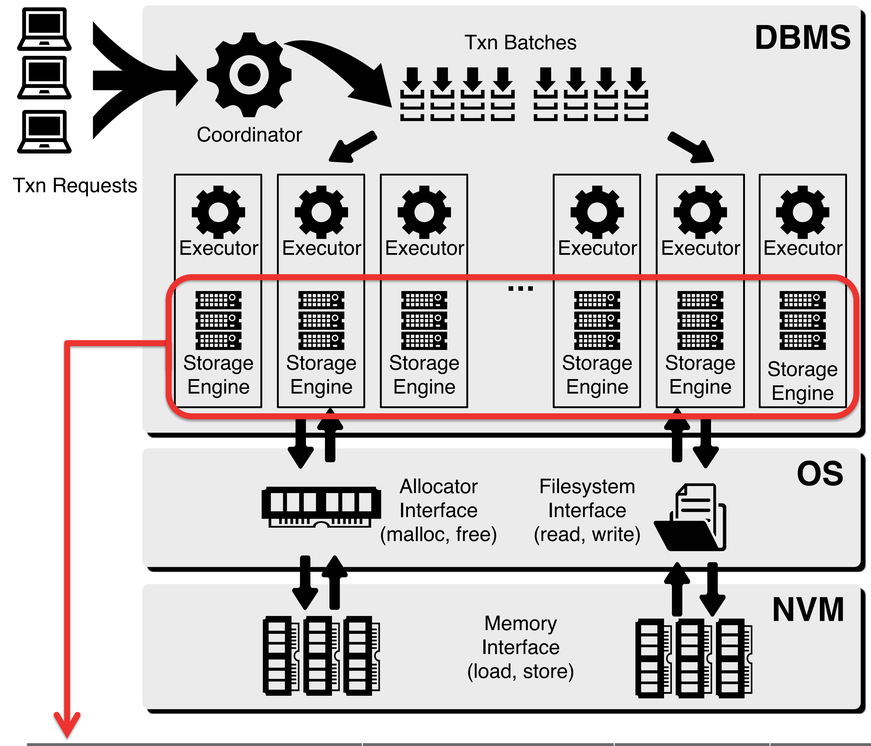
❖ Goal: Optimizing DBMS storage engines for NVM
- In-place updates (InP)
- Copy-on-Write updates (CoW)
- Log-structured updates (LOG)

❖ NVM hardware emulator (Intel Labs)
- Configure NVM load and store latency
- Throttle memory bandwidth
- File system and Allocator interfaces to NVM

❖ NVM-oriented optimizations
- Non-volatile pointer primitive
- Valid even after system restarts
- Non-volatile data structures used for indices, logging
- Built using an NVM-optimized memory allocator

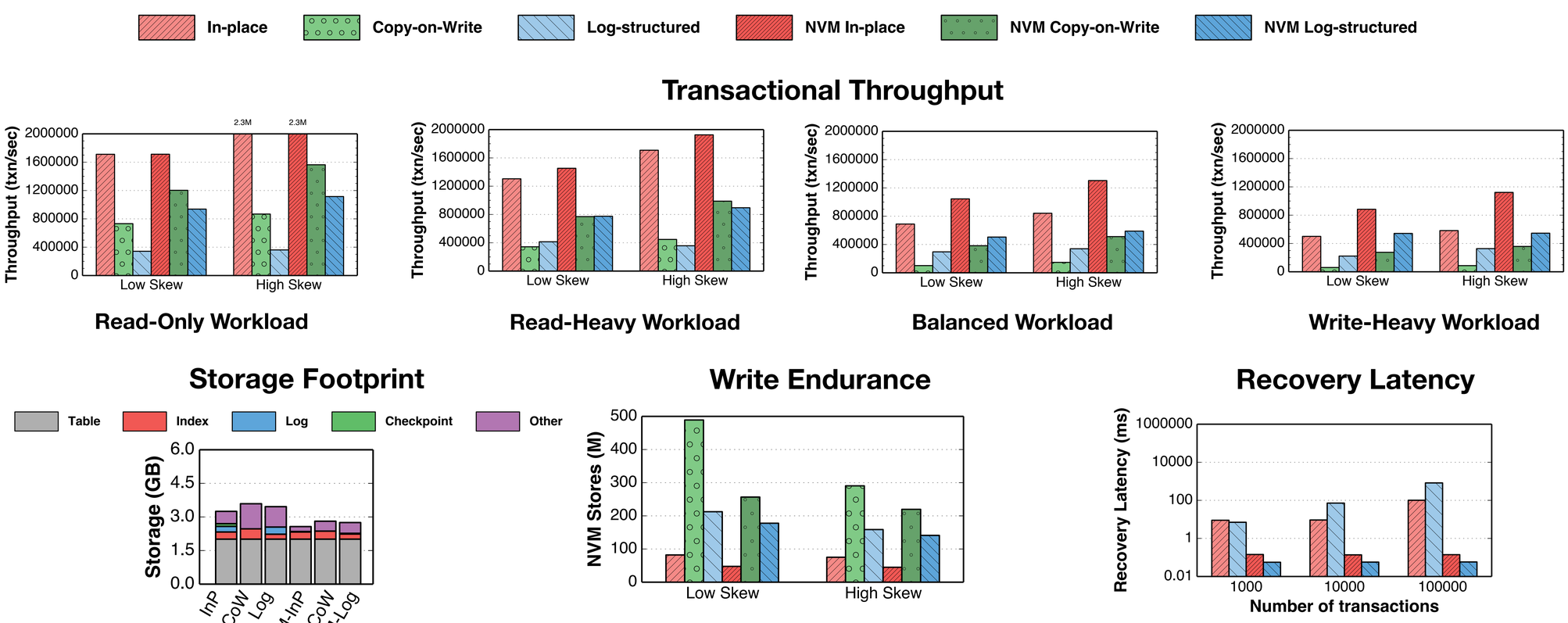
DBMS Testbed Platform

❖ Pluggable storage engine architecture
- Runs on NVM hardware emulator
- NVM-only design, no volatile DRAM



ENGINE TYPE	TABLE STORAGE	LOGGING	EXAMPLE
In-Place Updates	✓	✓	VoltDB
Copy-on-Write Updates	✓	✖	LMDB
Log-Structured Updates	✖	✓	LevelDB

Evaluation Results



Takeaways

❖ NVM-aware design pays off
- 2 - 5.5x higher throughput than traditional engines
- 1.5 – 2x longer device lifetime
- Smaller storage footprint on NVM
- Almost instantaneous recovery

❖ System design principles
- Non-volatile data structures are tricky
- Need a system-level rethink to leverage NVM

Future Work

❖ Peloton @ CMU
- Hybrid storage hierarchy (NVM + DRAM)
- Designed for HTAP workloads
- Real-time analytics and fast transactions
- Mixed row and columnar store

Carnegie Mellon University

CARNEGIE MELLON DATABASE GROUP

ISTC BIG DATA