OLTP on the NVM SDV: YMMV
2013

January Retreat
Job Interviews
Moved to CMU
Thesis Defense
December Retreat

ISTC
Big Data
Prison Life

G O O D

Washing Dishes
Not Fighting
Repentant

E V I L

Cafeteria Thievery
Shankings
Making Pruno
NVM OLTP

DRAM  

Lightweight CC  
Logical Logging  
Snapshots

SSD/HDD  

Heavyweight CC  
ARIES Logging  
Making Pruno
Project Overview

- Understand the performance characteristics of NVM to develop an optimal DBMS architecture for it.
Current Research Status

• Compare DBMS architectures:
  – H-Store with Logging $\rightarrow$ NVM
  – Anti-Caching $\rightarrow$ NVM (block)
  – Anti-Caching $\rightarrow$ NVM (tuple)
  – H-Store with MMAP $\rightarrow$ NVM
H-Store — NVM Edition

• Converted storage manager to use NVM through MMAP:
  – *Table tuple storage.*
  – *VARCHAR/BLOB data pool.*
  – *Table indexes.*
H-Store — NVM Edition

• Minor changes to exec engine:
  – *Group commit (msync)*

• Current implementation is not transactional fail-safe.
Experimental Evaluation

• Compare two architectures:
  – H-Store with MMAP → NVM
  – MySQL → NVM

• Yahoo! Cloud Serving Benchmark:
  – 10 million records (~10GB)
  – 8x database / memory MySQL
YCSB // Read-Only Workload
2x Latency Relative to DRAM

TXN/SEC

H-Store→NVM
MySQL→NVM

HIGH SKEW

LOW SKEW
YCSB // Read-Only Workload
16x Latency Relative to DRAM

- H-Store→NVM
- MySQL→NVM

↓11%
↓29%

TXN/SEC
HIGH SKEW
LOW SKEW

ISTC
BIG DATA
YCSB // 50% Reads / 50% Writes Workload
2x Latency Relative to DRAM

100,000
75,000
50,000
25,000
0

TXN/SEC

H-Store→NVM
MySQL→NVM

HIGH SKEW

LOW SKEW
YCSB // 50% Reads / 50% Writes Workload
16x Latency Relative to DRAM

H-Store→NVM
MySQL→NVM

TXN/SEC
0
25,000
50,000
75,000
100,000

HIGH SKEW
↓ 13%

LOW SKEW
↓ 8%
Discussion

• Overhead is only from reads and `msync` syscalls.

• We’re not doing anything special with the NVM.

  – Contents of NVM are ignored for crash recovery.
What’s Next?

• Port snapshot code to use NVM.
• Improve MMAP storage manager.
• Full comparison + experiments.
• Start building the next system...
N-Store — Possibility #1

• Keep DRAM.
• Hybrid OLTP/OLAP architecture:
  – Row Store $\rightarrow$ DRAM
  – Column Store $\rightarrow$ NVM
• Multi-faceted indexes.
N-Store — Possibility #2

• Ditch DRAM.
• Instant recovery:
  – *No snapshots.*
  – *No logging.*
• NVM-friendly concurrency control scheme.
What do we need?

• Be able to tune the `msync` latency of the NVM SDV.
• Custom `malloc` implementation that runs entirely of PMFS.