Benchmarking Persistent Memory in Computers

NVDIMM-N Testing with MongoDB

Presenter: Adam McPadden

Co-Authors: Moshik Hershcovitch and Revital Eres

Overview

- Objective
- Background – System Configuration
- MongoDB Testing
- Results
- Conclusions

Goals

1. Implement a Persistent Memory (PMEM) architecture in an Enterprise class POWER8 server running Linux

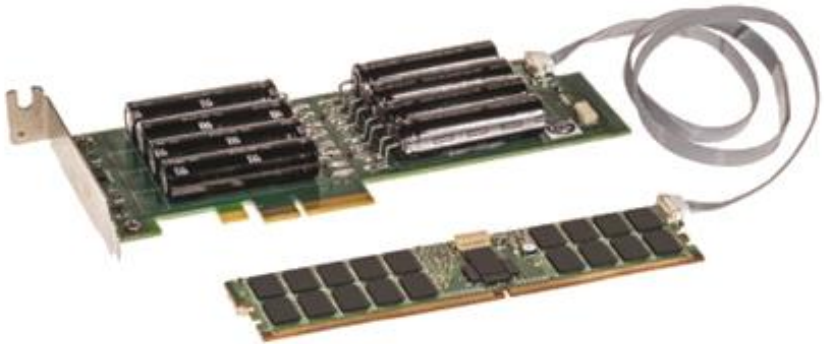
2. Measure application-level performance benefits of NVDIMM-N with MongoDB

3. Extrapolate benefits for Storage Class Memory (SCM)

Background – NVDIMM-N

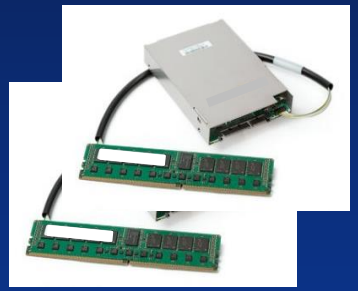
2.5” DASD SuperCap card

PCIe SuperCap card



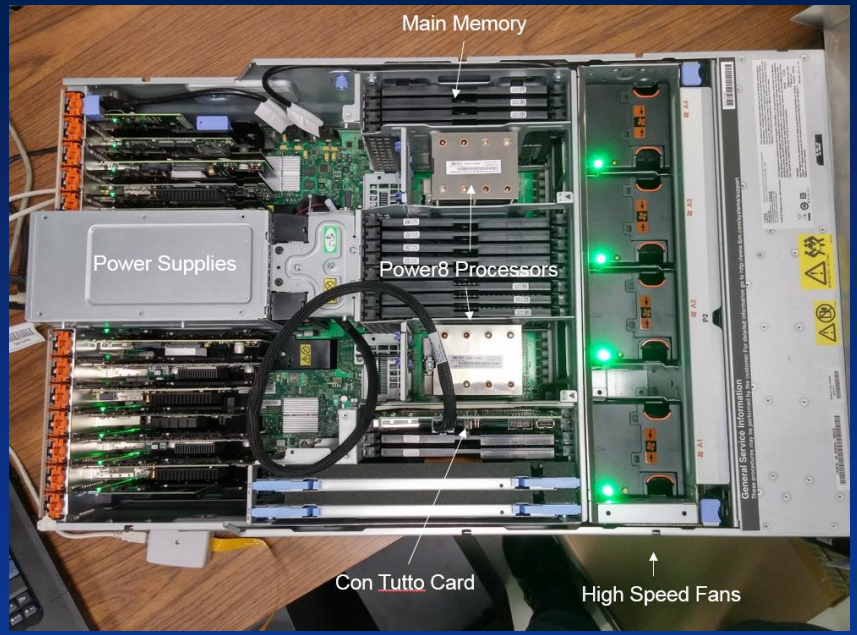
- DRAM is mapped / host accessible
- DRAM data saved to FLASH upon power loss

System Configuration – Hardware



2x - 8GB NVDIMM-N

Con Tutto FPGA Card



S824L Power8 Server

Memory Config: NVDIMM-N = 16GB, DRAM = 256GB
SSD: 400GB 6G SAS
Note: Con Tutto FPGA contributes ~300ns to latency measurements

MongoDB - Definitions

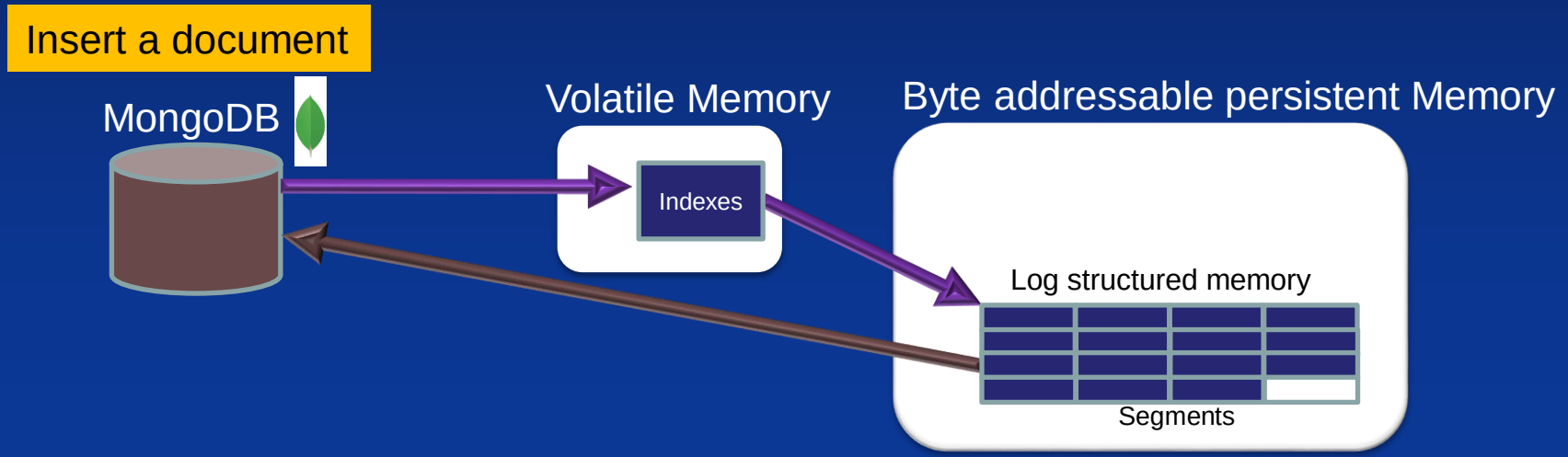
- **MongoDB**: an Open Source, highly popular cross platform document-oriented NoSQL db
- **Durability**: the guarantee that written data has been saved
- **Storage Engine** (SE): Manages how data are stored to memory/disk
  - MMAPv1: The original MongoDB SE (I/O's to disk for durability)
  - WiredTiger: The standard SE from V3.2 (I/O's to disk for durability)
  - In-Memory: Stores data in memory, does not guarantee persistence

The Challenge

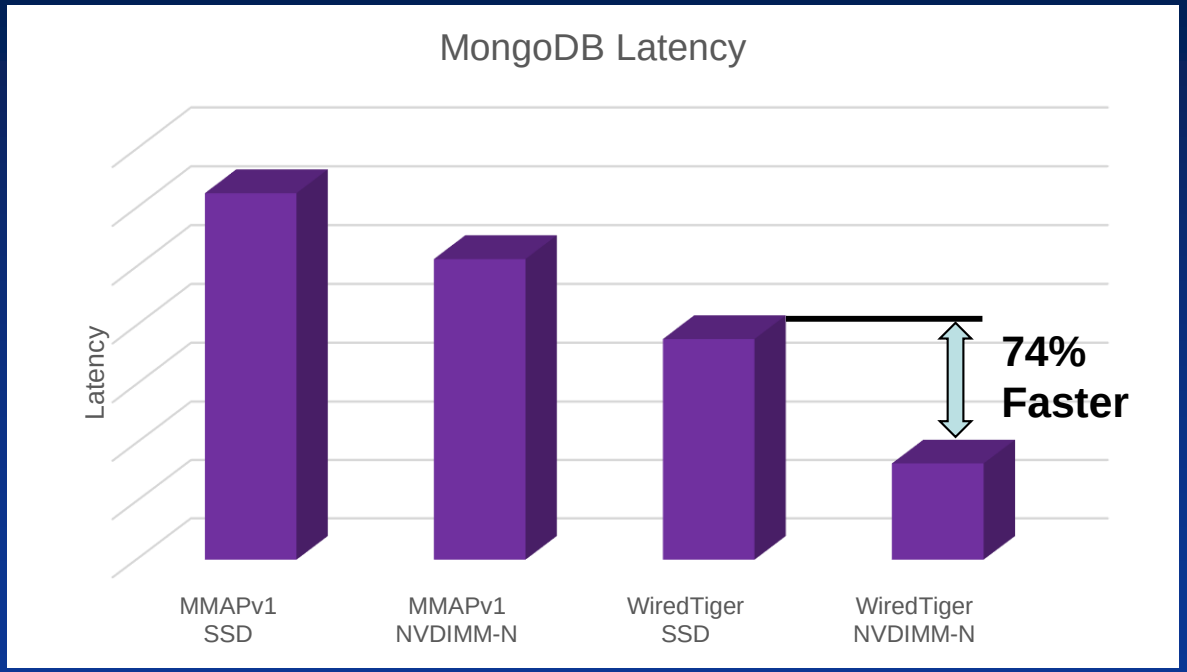
- Objective
  1. Achieve the performance benefits of running “in-memory”
  2. Guarantee durability / persistence
  3. Lower write amplification
- Solution
  1. Create a PMEM/SCM aware Storage Engine (IBM Research Haifa)
  2. Use a fast, Non-volatile memory
  3. Quantify MongoDB (YCSB) performance benefit with
     - NVDIMM-N using existing Storage Engines
     - NVDIMM-N using a PMEM aware Storage Engine

SE Architecture

- The record store resides in a log structured array of segments in PM
- Indexes are implemented with a B-tree and stored in the volatile memory
- SE has DAX access to PM

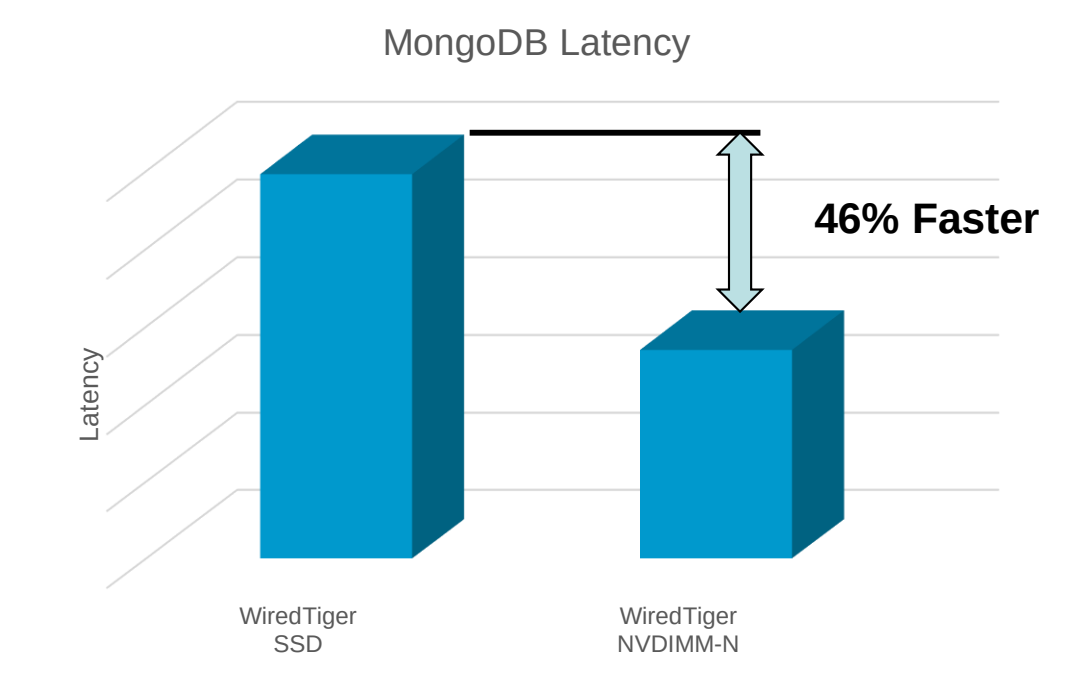


MongoDB: Latency 100% Writes



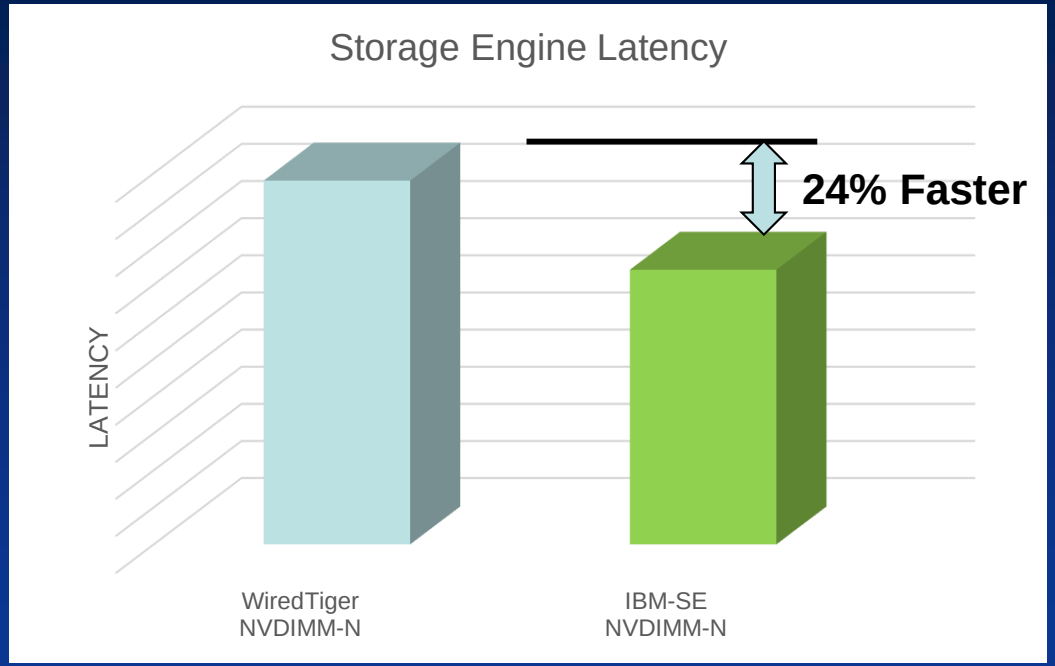
- Test = 100% Writes:
  - Single Thread
  - Load: 2M Writes
  - Run: 2M Writes
- Net: Significant latency improvement with NVDIMM-N

MongoDB: Latency Mixed R/W



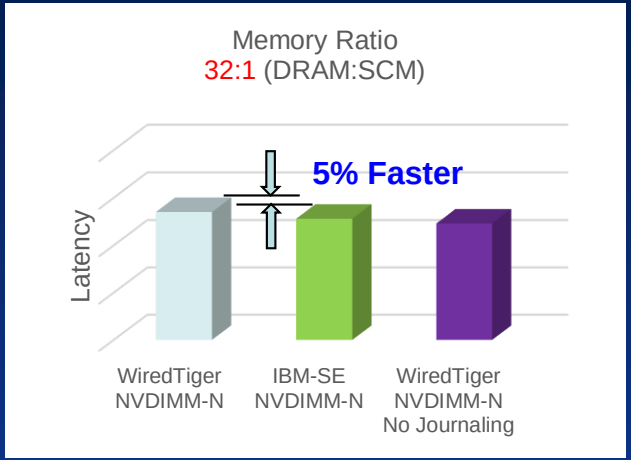
- Test = Mixed R/W:
  - Single Thread
  - Load: 3M Writes
  - Run: 4M R/W 50/50
  - WiredTiger SE
- Net: Significant latency improvement with NVDIMM-N

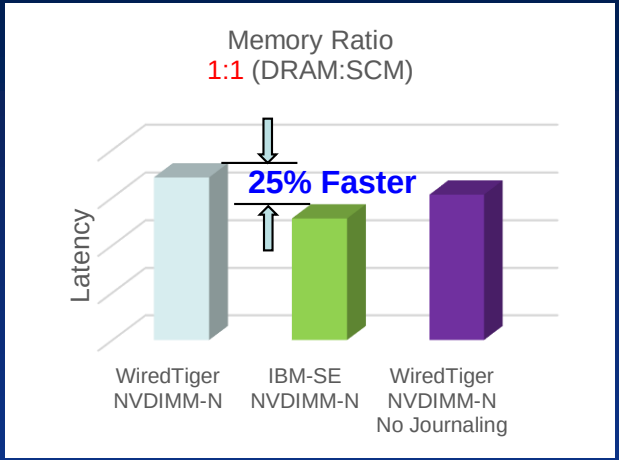
PMEM aware Storage Engine

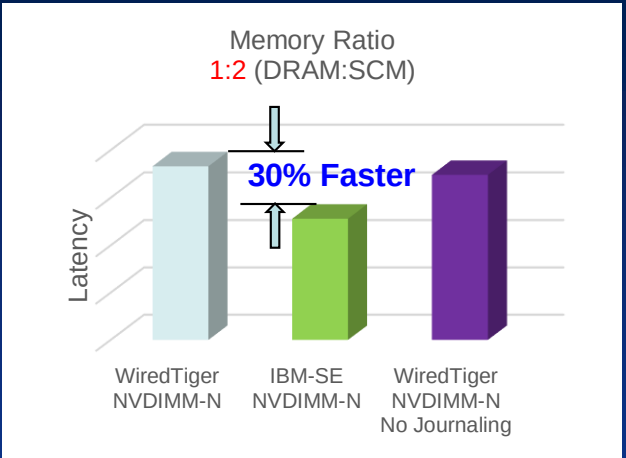


- Test = 100% Writes:
  - Single Thread
  - Load: 2M Writes
  - Run: 2M Writes
- Net: Additional opportunity exists for a PMEM aware Storage Engine

MongoDB: SCM Extrapolation







- Storage Class Memory Extrapolation
  - Latency: NVDIMM-N + Con Tutto FPGA ≈ SCM Latency Domain (low hundreds of ns)
  - Capacity: Modify the DRAM, NVDIMM-N Capacities to emulate different DRAM:SCM Ratios
- Results: Performance with a pmem aware Storage Engine increased with higher SCM:DRAM ratios. DBs which don't fit in WiredTiger DRAM cache, but can fit in larger SCM pmem space will be faster

Conclusions

- Significant application level performance gains are achievable with NVDIMM-N and a PMEM architecture
- NVDIMM-N show appreciable latency benefits with existing MongoDB Storage Engines
- A Storage Class Memory/PMEM aware durable Storage Engine can
  - further improve MongoDB performance
  - Lower Write Amp: Journaling to disk is unnecessary
- Quantifying the end customer value proposition is critical in driving industry adoption of persistent memory with NVDIMM-N and Storage Class Memories

Additional Resources

http://openpowerfoundation.org/

OpenPOWER Contutto Presentation

Programmable Near Memory Acceleration on Contutto

Making Unforgettable MRAM Memory with OpenPOWER