SNIA CMSI | COMPUTE, MEMORY, AND STORAGE

So You Wanna' Program Persistent Memory?

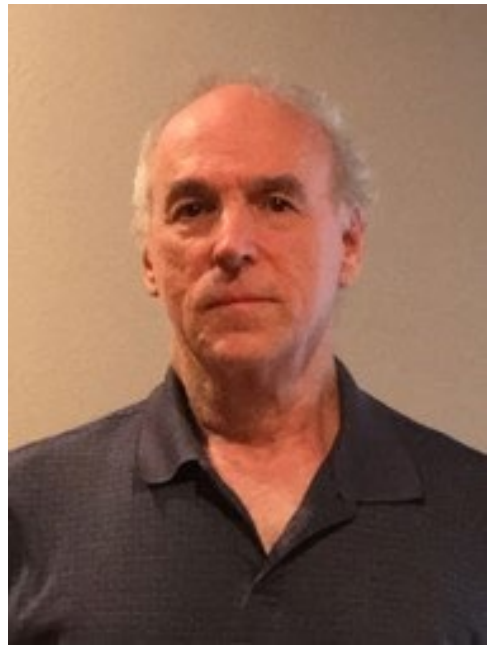
A Challenge Primer

A SNIA Compute, Memory, and Storage Initiative Webcast

Today's Speakers

Moderator:
Alex McDonald
Co-Chair, SNIA Compute, Memory, and Storage Initiative

Presenter:
Jim Fister
Director, SNIA Persistent Memory Enabling



Presenter:
Steve Heller
President, Chrysalis Software Corporation

 ©2020 Storage Networking Association. All Rights Reserved.

SNIA Legal Notice

- The material contained in this presentation is copyrighted by the SNIA unless otherwise noted.
- Member companies and individual members may use this material in presentations and literature under the following conditions:
  - Any slide or slides used must be reproduced in their entirety without modification
  - The SNIA must be acknowledged as the source of any material used in the body of any document containing material from these presentations.
- This presentation is a project of the SNIA.
- Neither the author nor the presenter is an attorney and nothing in this presentation is intended to be, or should be, construed as legal advice or an opinion of counsel. If you need legal advice or a legal opinion please contact your attorney.
- The information presented herein represents the author's personal opinion and current understanding of the relevant issues involved. The author, the presenter, and the SNIA do not assume any responsibility or liability for damages arising out of any reliance on or use of this information.

  NO WARRANTIES, EXPRESS OR IMPLIED. USE AT YOUR OWN RISK.

 ©2020 Storage Networking Association. All Rights Reserved.

Keys to Programming Persistent Memory

- Consistent Windows/Linux architecture model
- Variety of open-source tools and libraries
  - Persistent Memory Development Kit (PMDK)
  - Direct programming models
  - Multiple open-source file systems
- Our ‘19 promise at SNIA: Programming/ Conversational opportunities
  - Persistent Memory Hackathons, global reach
  - **NVDIMM Programming Challenge**
  - A new, industry-focused conference (PIRL)

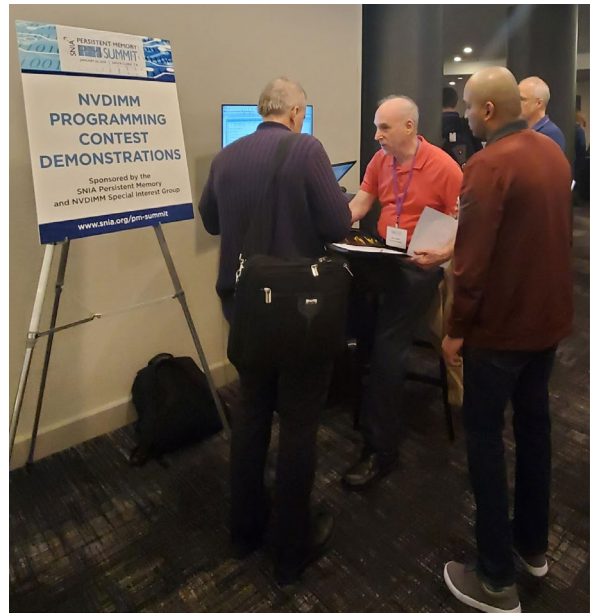
SNIA 2020 PERSISTENT MEMORY HACKATHON

pmhackathon@snia.org

 ©2020 Storage Networking Association. All Rights Reserved.

NVDIMM Programming Challenge

- Sponsored by SNIA Persistent Memory and NVDIMM Special Interest Group members SMART Modular Technologies, AgigA Tech, and Supermicro
- Program ran for six months
  - Multiple submissions
  - Panel review by technologists and sponsors
- Challenge Winner:
  - Steve Heller, Chrysalis Software Corporation
    - Three Misses, Persistent Hash Table
    - Utilizes PM to ease set-up time as well as the DRAM speed of NVDIMM
    - Submission at www.threemisses.com



 ©2020 Storage Networking Association. All Rights Reserved.

The Winning Submission

 ©2020 Storage Networking Association. All Rights Reserved.

We Need a New Type of Hash Table

- Standard hash table implementations assume everything is in memory
- Standard database program implementations assume that memory accesses are irrelevant and CPU use is nearly so
- Persistent memory upends these assumptions because it requires both memory access optimization and storage access optimization

 ©2020 Storage Networking Association. All Rights Reserved.

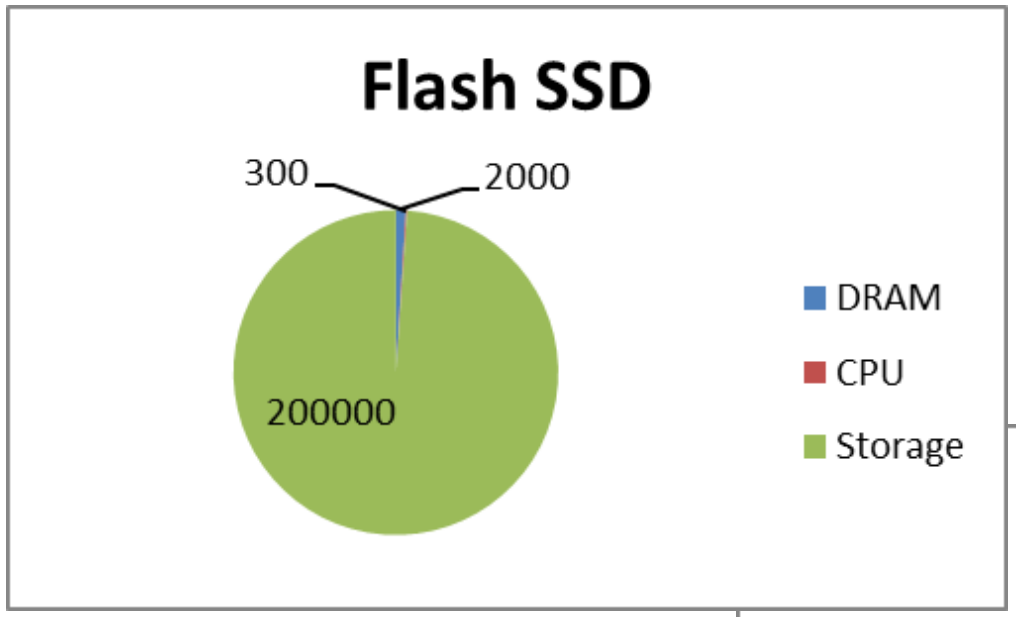
SSD, Memory, CPU Assumptions

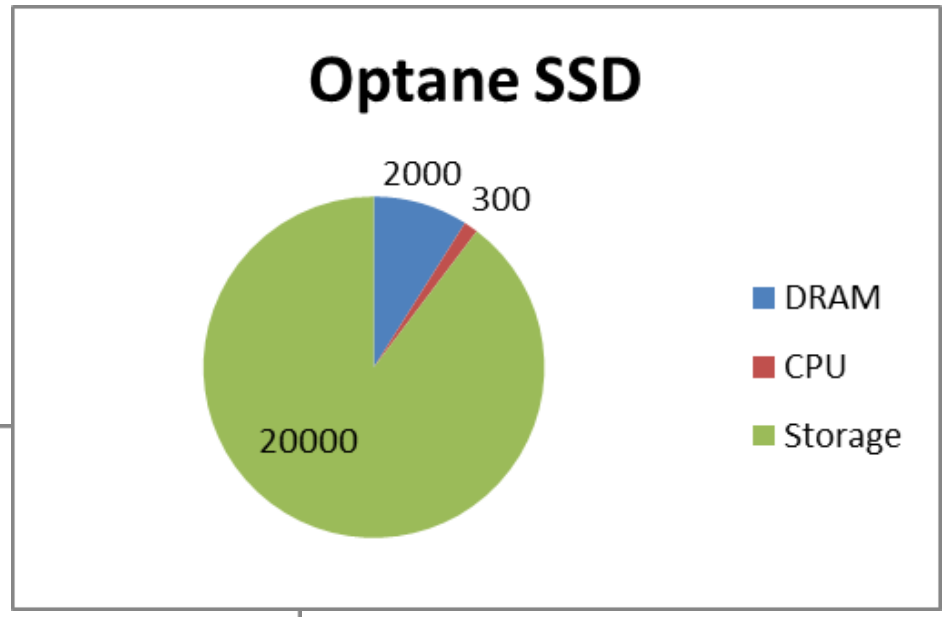
- Assume the following time scales for accessing a record in a hash table (rough approximations):
  - DRAM latency: 100 nanoseconds
  - CPU processing time: 300 nanoseconds
  - Flash SSD latency: 100,000 nanoseconds
  - Optane SSD latency: 10,000 nanoseconds
  - Optane PM latency: 150 nanoseconds
  - 20 cache-hostile DRAM accesses per record

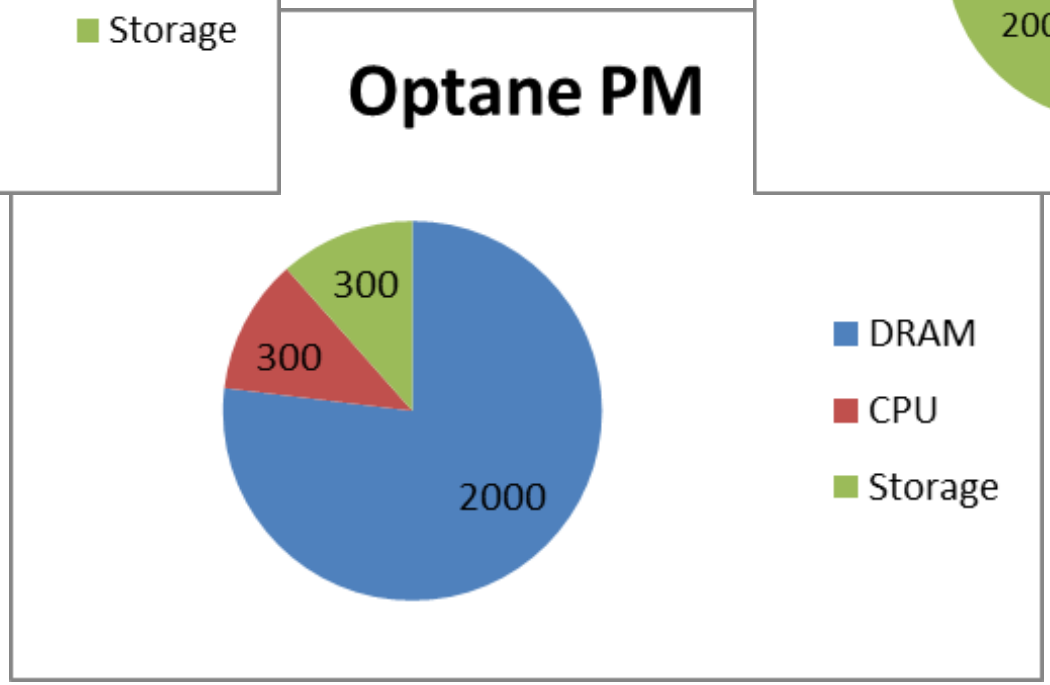
- What do the overall times look like?

 ©2020 Storage Networking Association. All Rights Reserved.

Visualizing the SSD, Memory, and CPU Assumptions







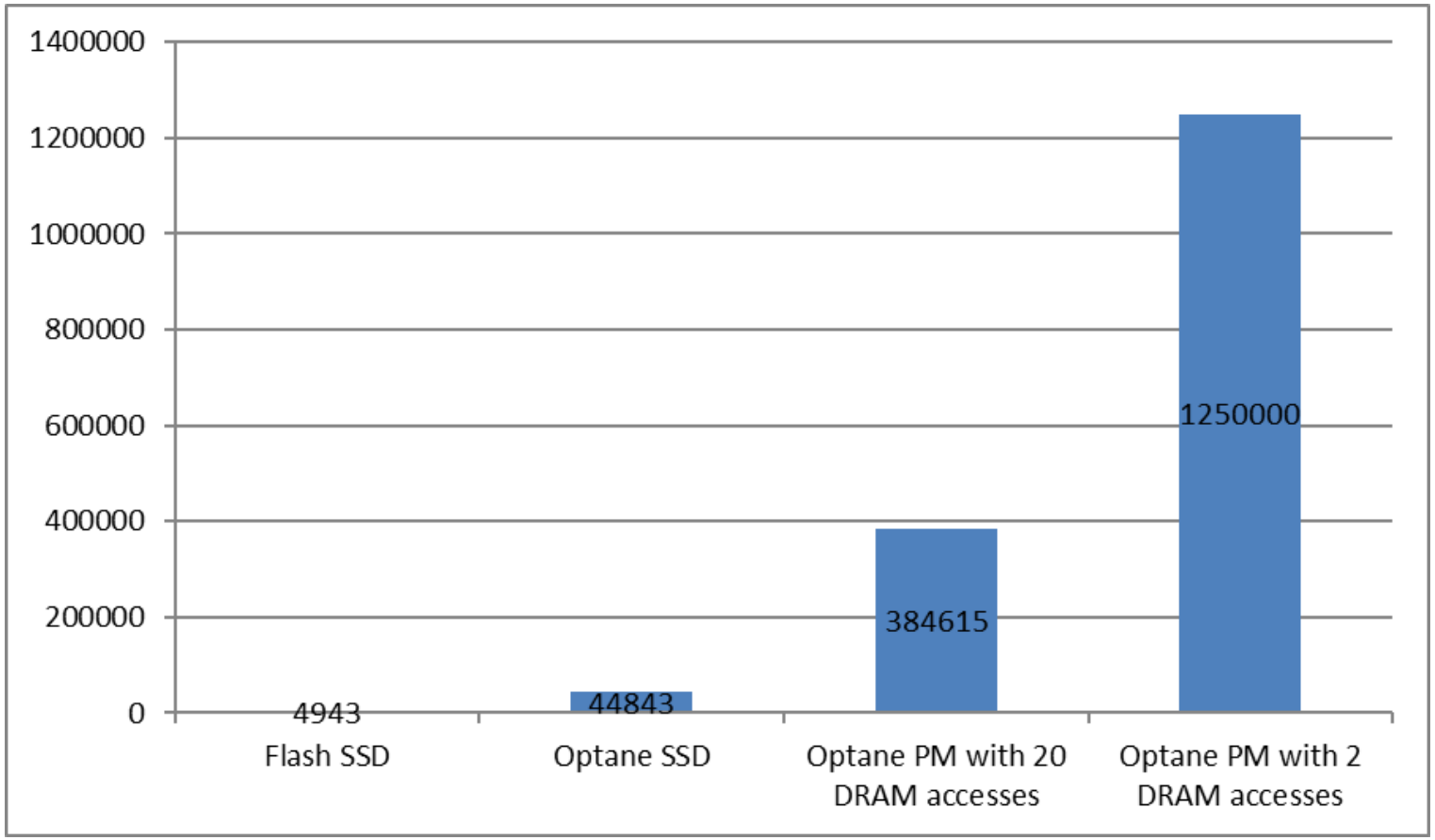
 ©2020 Storage Networking Association. All Rights Reserved.

What Matters for Performance?

- Flash SSD: CPU & DRAM latency are negligible.
- Optane SSD: CPU & DRAM latency are noticeable but not very important.
- Optane PM: CPU & DRAM latency are crucial!
- What level of performance improvement over Flash SSD might we obtain with Optane PM, if we minimize cache-hostile DRAM accesses?

 ©2020 Storage Networking Association. All Rights Reserved.

Would You Believe **250x**?



Some Basic Tips:

- Minimize cache-hostile accesses to memory and PM devices
- Avoid big vectors even if they fit in the L3 cache
- Avoid time-consuming algorithmic operations like string copying

 ©2020 Storage Networking Association. All Rights Reserved.

The ThreeMisses Solution

- **ThreeMisses** is a file-format, (natively serialized) persistent, hash table that provides an interface similar to the C++ unordered_map<string,string> data type, as well as a more advanced interface for higher performance
- It is currently implemented on Windows 10 and Ubuntu 19.10

 ©2020 Storage Networking Association. All Rights Reserved.

ThreeMisses Implementation

- ThreeMisses uses Robin Hood hashing to store index records
- No index search requires more than two cache-hostile storage accesses
- Any value that won't fit in the index file is stored in one contiguous piece
- Following these rules, we can access any record with a maximum of three cache-hostile storage accesses

 ©2020 Storage Networking Association. All Rights Reserved.

How to Use ThreeMisses

- Instantiate a **ThreeMisses** variable to represent the data, providing paths for the index and data files
- Then you can use the **ThreeMisses** variable as though it were a memory-resident hash table like "unordered_map<string, string>"
- Sample code:

```
#include "UMVMCommon.h"

int main()
{
   UMVM::ThreeMisses TestMap("f:\\test.kvi", "f:\\test.kvd");

   TestMap["abc"] = "def";
}
```

 ©2020 Storage Networking Association. All Rights Reserved.

Features By Edition

Base Features

- Enter record in hash table by key (size_t or string address)
- Retrieve record in hash table by key (size_t or string address)
- Visual Studio 2019 .lib file or clang 9.0 .a file
- Variable-length key and value lengths up to size of the full file
- Sample program
- Advanced minimal-copy access
- Non-resident data file size up to 2^42 bytes

Additional Features via Pro Subscription

- Index file size limited only by storage (up to 2^42 bytes)
- Iterators similar to those for unordered_map
- Optimizer for table scan speed and index reorganizer
- Overflow option to new volume
- Tuning performance for large record sizes
- Additional sample program
- Priority technical support

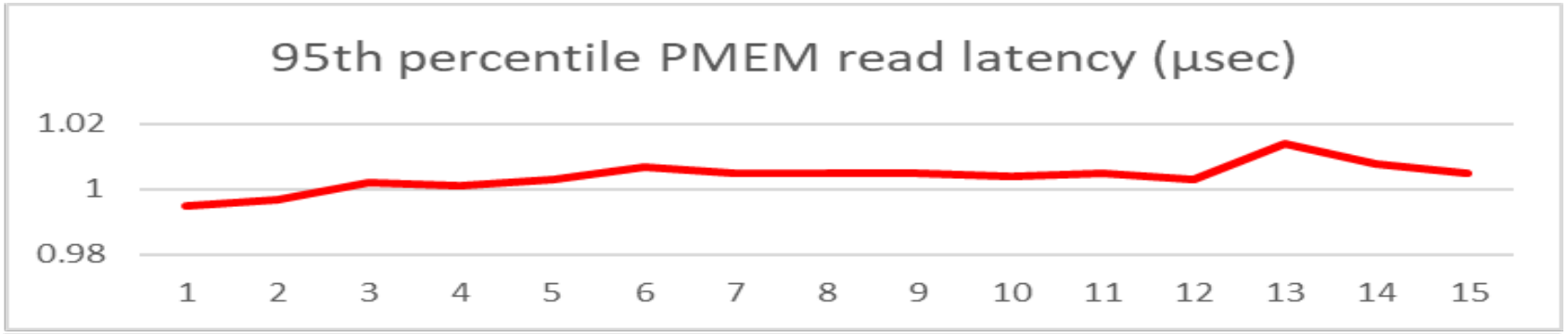
 ©2020 Storage Networking Association. All Rights Reserved.

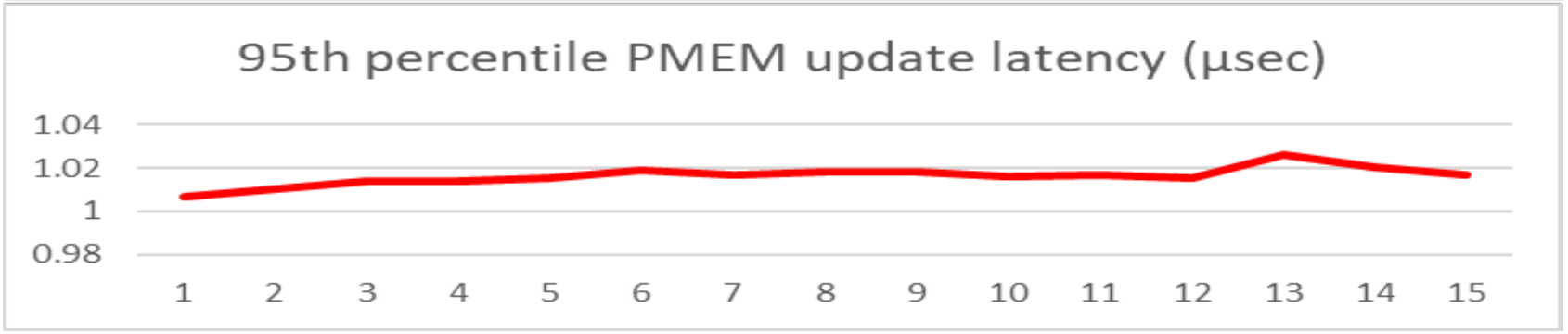
Testing Environment

- PMEM Server hardware configuration: Xeon Silver CPU, 128 GB DRAM, 4x128 Optane DIMMs
- NVDIMM Server hardware configuration: Xeon Silver CPU, 32GB DRAM, 96 GB NVDIMMs interleaved
- Half read, half read-and-update in place
- All data stored on PMEM or NVDIMM

 ©2020 Storage Networking Association. All Rights Reserved.

PMEM Test Results

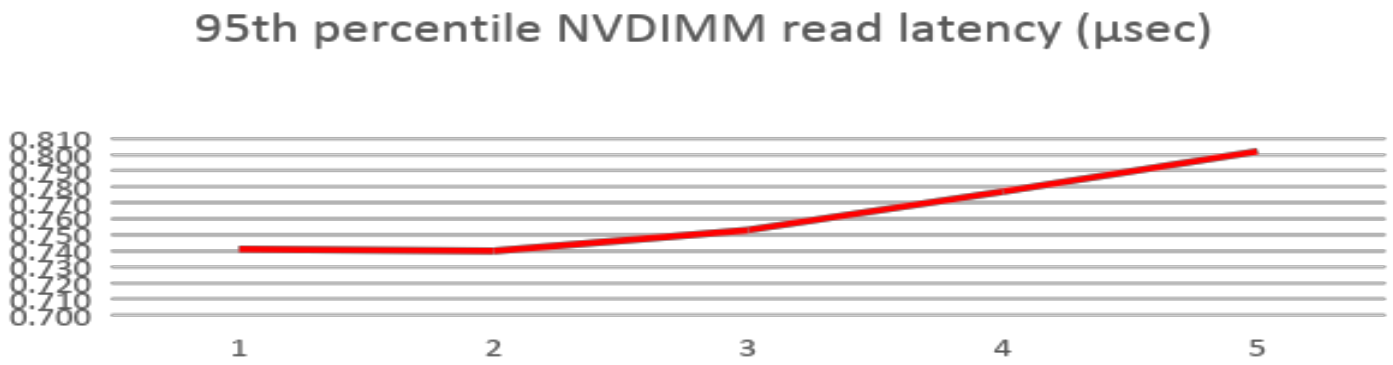


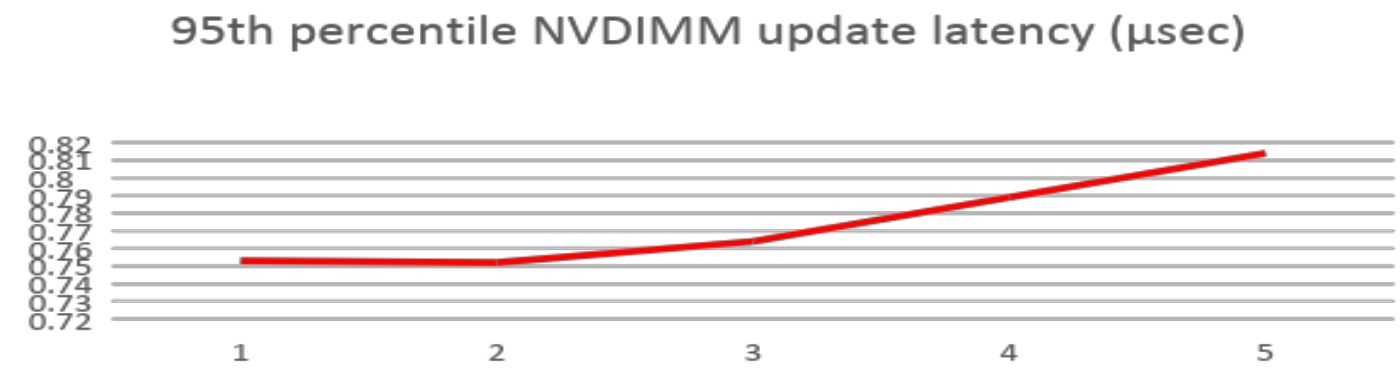


Fixed-length records with 8-byte keys, 8-byte values
Linux OS, Storage allocation = 30 x record count, Horizontal axis: Record count in billions

 ©2020 Storage Networking Association. All Rights Reserved.

NVDIMM Test Results





Variable-length records averaging 6-byte keys and 3-byte values
Linux OS, Storage allocation = 17 x record count Horizontal axis: Record count in billions

 ©2020 Storage Networking Association. All Rights Reserved.

Third Party Test Results

- Tested by a research group at UNC-Charlotte
- Results available at GitHub repo pmemids_bench
  - https://github.com/DIR-LAB/ycsb-storedsbench/blob/threemiss/UNCTesting/results/threemisses.csv
- They also have a paper titled “A Performance Study of Optane Persistent Memory: From Indexing Data Structures’ Perspective”, accepted in **MSST 2020** (https://www.storageconference.us)

 ©2020 Storage Networking Association. All Rights Reserved.

The Future

- ThreeMisses is a "killer app" for persistent memory
- Actively pursuing relationships with companies that will help get it into the hands of all PM developers.

For more details, see threemisses.com

 ©2020 Storage Networking Association. All Rights Reserved.

SNIA Efforts: What About 2020?

- Hackathons Continue
  - US, Europe, potential Asia opportunities
  - Don't want to wait? Host your own Hackathon!
  - NVDIMM and Optane Programming Challenges
  - NVDIMM Challenge Extended! Interested participants contact pmhackathon@snia.org

- Continued focus on expanding the PM conversation
  - Submit your own talk for a webcast via the PIRL Submission Form
  - Considering virtual conference opportunities, check our SNIA Compute Memory & Storage blog for details

 ©2020 Storage Networking Association. All Rights Reserved.

Additional Resources

- Webcast & PDF - The SNIA Persistent Memory Programming Model
  - https://www.snia.org/educational-library/persistent-memory-programming-model-2020
- Presentations from the 2020 SNIA Persistent Memory Summit
  - https://www.snia.org/pm-summit
- Book - Programming Persistent Memory – A Comprehensive Guide for Developers
  - https://www.apress.com/us/book/9781484249314
- Specification: SNIA NVM Programming Model
  - https://www.snia.org/forums/cmsi/nvmp

 ©2020 Storage Networking Association. All Rights Reserved.

Finally, Thanks for Watching Our Webcast

- Please rate this webcast and provide us with feedback
- This webcast and a PDF of the slides will be posted to the SNIA Compute Memory and Storage Initiative website and available on-demand at https://www.snia.org/forums/cmsi/knowledge/articles-presentations
- A Q&A from this webcast will be posted to the SNIA CMSI blog: www.sniasssiblog.com
- Follow us on Twitter @sniasolidstate

 ©2020 Storage Networking Association. All Rights Reserved.

Questions?

Thank you!

 ©2020 Storage Networking Association. All Rights Reserved.