Hibachi: A Cooperative Hybrid Cache with NVRAM and DRAM for Storage Arrays

Ziqi Fan, Fenggang Wu, Dongchul Park[1], Jim Diehl, Doug Voigt[2], and David H.C. Du

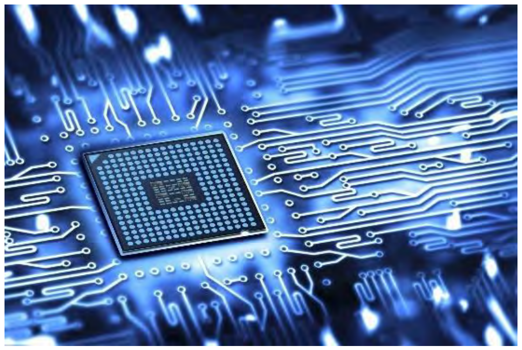
University of Minnesota, [1]Intel, [2]HP Enterprise

May 18, 2017

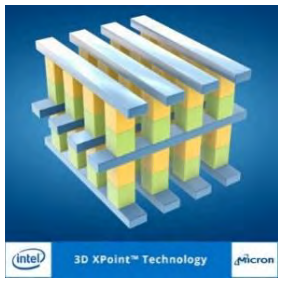
CRIS Center for Research in Intelligent Storage

University of Minnesota Driven to Discover℠

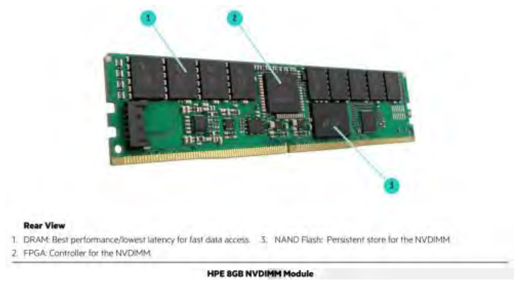
Hardware evolution leads to software and system innovation!



The hardware evolution of non-volatile memory (NVRAM)



3D Xpoint
(By Intel and Micron)


1
2
3
Rear View
1. DRAM: Best performance/lowest latency for fast data access.
2. FPGA: Controller for the NVDIMM
3. NAND Flash: Persistent store for the NVDIMM
HPE 8GB NVDIMM Module

NVDIMM
(By HPE)

STT-MRAM
(By Everspin)

- ✓ Non-volatile
- ✓ Low power consumption
- ✓ Fast (close to DRAM)
- ✓ Byte addressable
- ✓ …

Center for Research in Intelligent Storage

University of Minnesota Driven to Discover℠

How to innovate our software and system to exploit NVRAM technologies?

Many Possible Ways

Caching Systems

Application Upgrade

OS Optimization

Design NVRAM-based caching systems to improve storage performance

Research Contributions

Extend **solid state drive** lifespan

→ H-ARC (in MSST 2014 [1]) 1

→ WRB (under TOS Major Revision) 2

Increase **hard disk drive** I/O throughput

→ I/O-Cache (in MASCOTS 2015 [2]) 3

Improve **disk array** performance

→ Hibachi (in MSST 2017 [3]) 4

Parallel File System

Increase **PFS** checkpointing speed

→ CDBB (Under Submission) 5

UNIVERSITY OF MINNESOTA

A Cooperative Hybrid Cache with NVRAM and DRAM for Disk Arrays

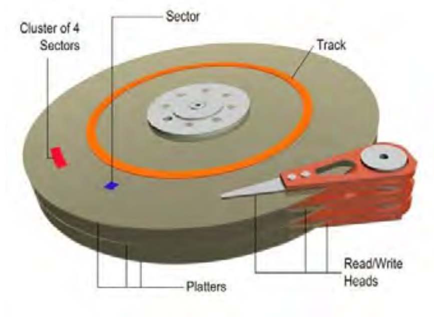
Outline

- Motivation
- Related Work
- Design Challenges
- **Our Approach**
- Evaluation
- Conclusion

Introduction

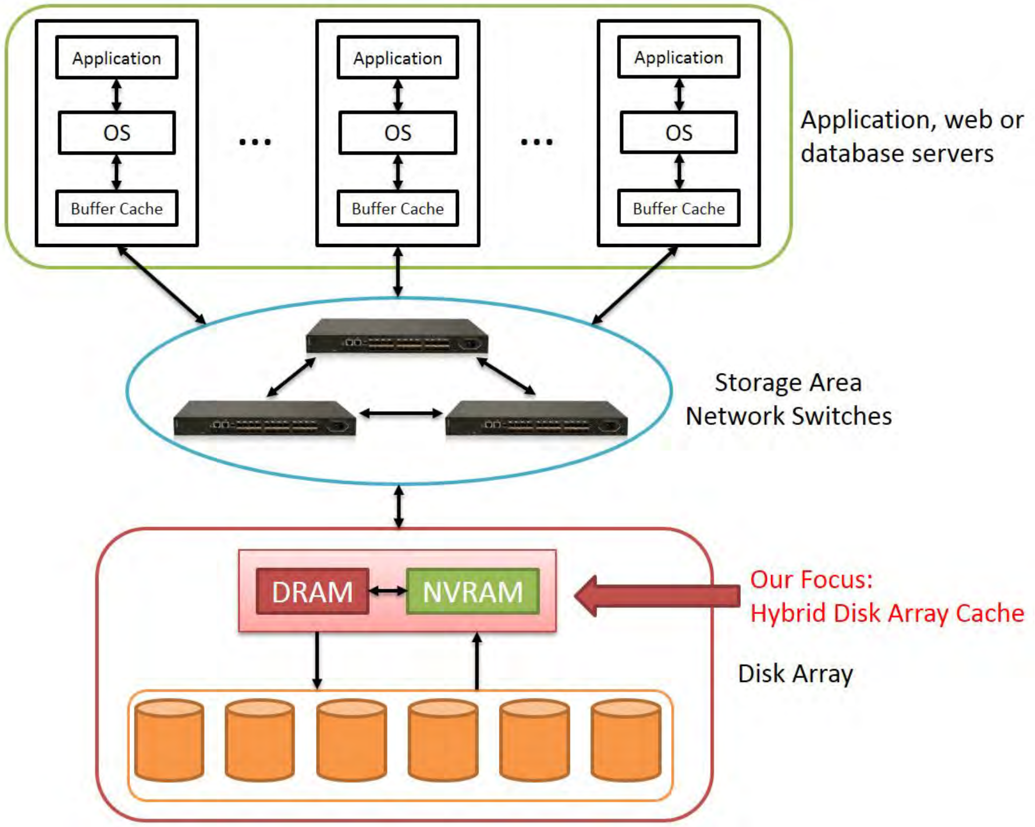
- Despite the rise of SSDs, disk arrays are still the backbone storage, especially for large data centers
- HDDs are much cheaper in capacity/$ and do not wear out easily

- However, as rotational devices
  - HDDs sequential throughput: ~100MB/s
  - HDDs random throughput : < 1MB/s



[4]

Introduction



- To improve disk performance, we use NVRAM and DRAM as caching devices
  - Disk cache is much larger than page cache and DRAM is more cost-effective than NVRAM
  - DRAM has lower latency than some types of NVRAM

Crux: How to design a hybrid disk cache to fully utilize scarce NVRAM and DRAM resources?

Center for Research in Intelligent Storage

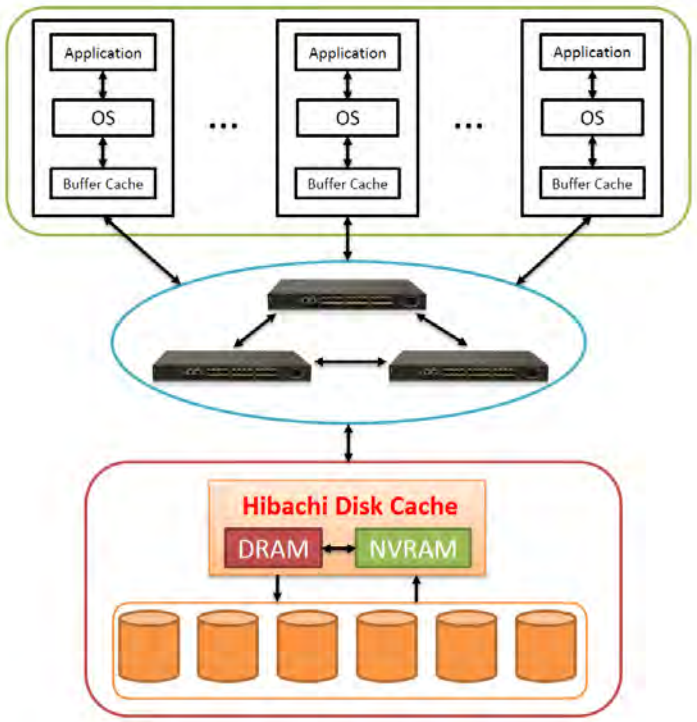
University of Minnesota
Driven to Discover℠

Related Work

- Cache policies designed for main memory (first-level cache)
  - **Not directly applicable to disk cache**
  - LRU, ARC[5], H-ARC [1]
- Multilevel buffer cache (including both first-level and second-level caches)
  - **Concentrate on improving read performance**
  - **Not considering NVRAM**
  - MQ [6], Karma [7]
- Disk cache with DRAM and NVRAM
  - **DRAM as read cache and NVRAM as write buffer $\rightarrow$ lack cooperation**

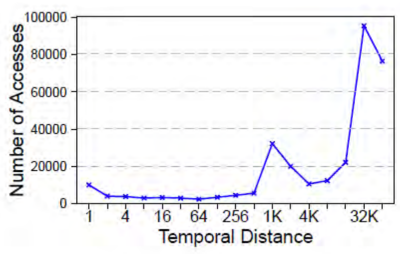
Design Challenges

- How to analyze and utilize I/O traces after first-level cache to design disk cache as second-level cache?
- How to utilize DRAM to maximize read performance?
  - Low access latency (high cache hit rate)
- How to utilize NVRAM to maximize write performance?
  - High I/O throughput
- How to exploit the synergy of both NVRAM and DRAM?
  - Help each other out according to workload properties

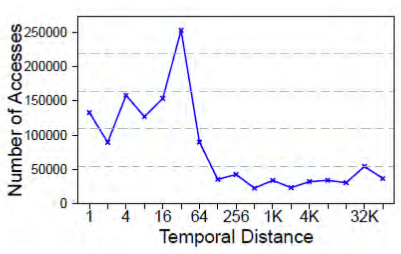


I/O Workload Characterization of Traces after First-level Cache

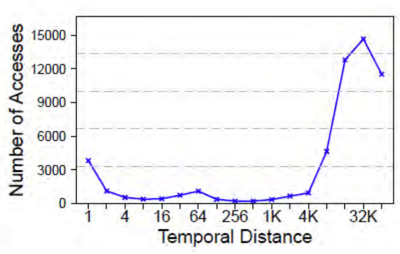
- Existing work only characterizes read requests [10]
- On top of existing work, we characterize both read and write requests



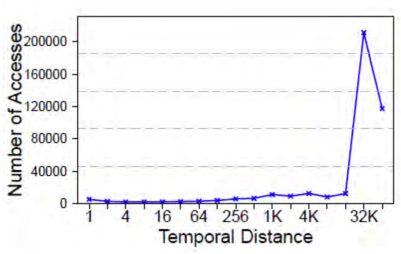
(a) web_0 read after read



(b) web_0 write after write



(c) web_0 read after write



(d) web_0 write after read

Temporal distance histograms of a storage server I/O workload.

✓ For read requests, stack distance is large -> recency is bad

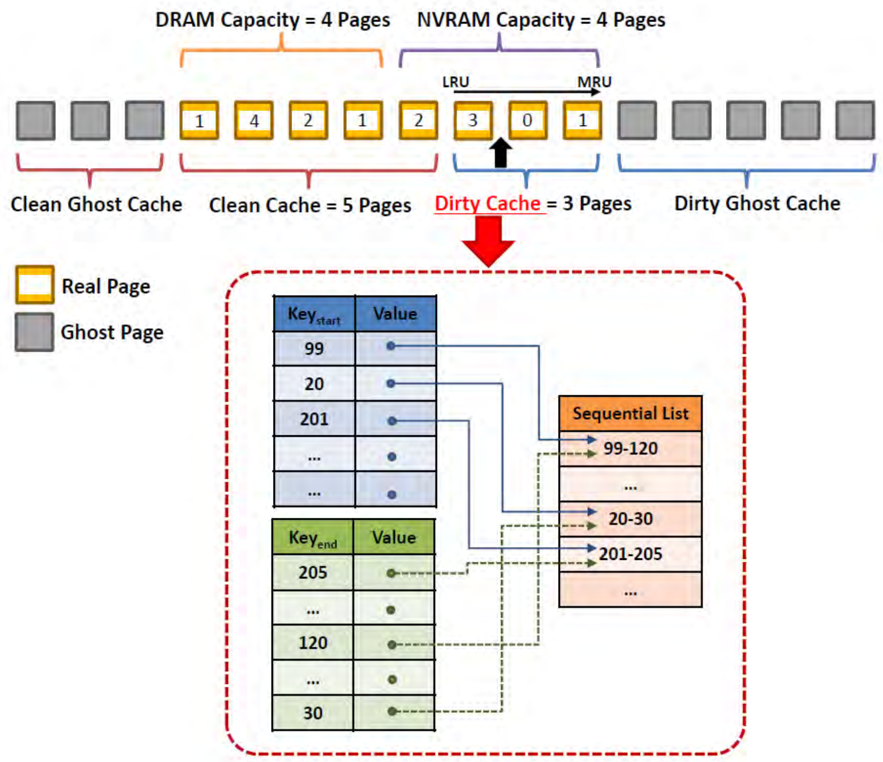
✓ For write requests, stack distance is relatively short -> recency can be useful for cache design

✓ Frequency is useful for both read and write

Center for Research in Intelligent Storage

University of Minnesota
Driven to Discover℠

Hibachi – Cooperative Hybrid Disk Cache

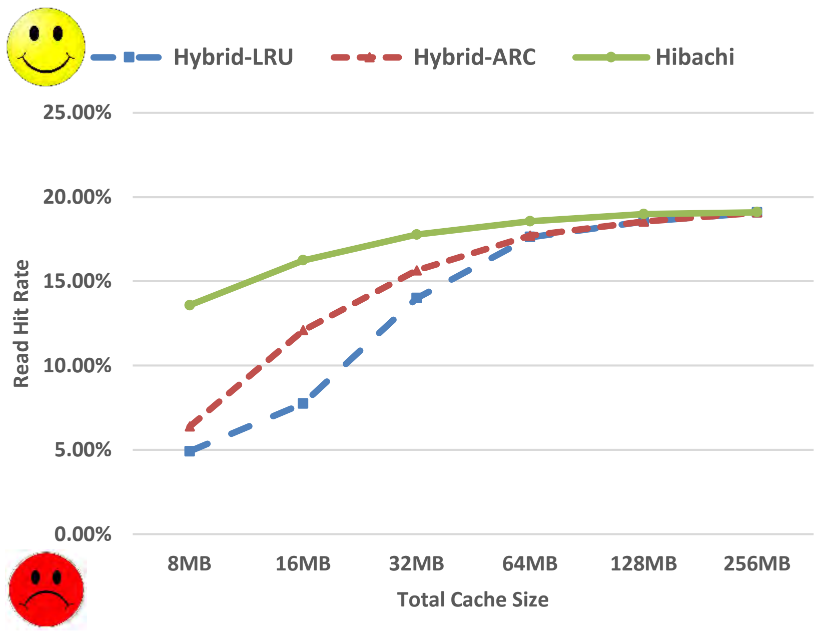


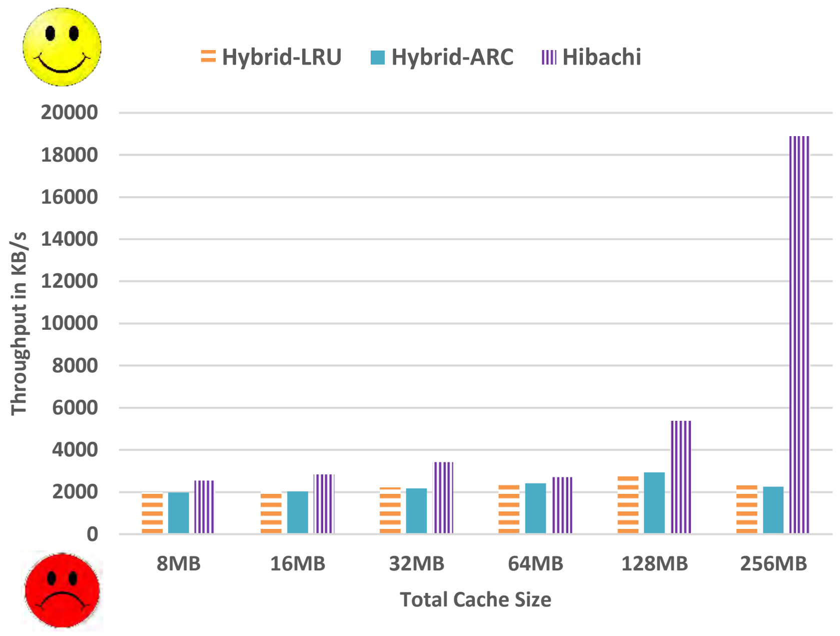
- Our Hibachi's **four** secret ingredients to make it "taste better"
  - Right Prediction → Improve cache hit ratio
  - Right Reaction → Minimize write traffic and increase read performance
  - Right Adjustment → Adaptive to workload
  - Right Transformation → Improve I/O throughput

Evaluation Setup

- Use Sim-ideal [9] to measure read performance
- Use software RAID with six disk drives to measure write performance
- Comparison algorithms:
  - Hybrid-LRU: DRAM is a clean cache for clean pages, and NVRAM is a write buffer for dirty pages. Both caches use the LRU policy.
  - Hybrid-ARC: An ARC-like algorithm to dynamically split NVRAM to cache both clean pages and dirty pages, while DRAM is a clean cache for clean pages.

Evaluation Results





- Hibachi outperforms Hybrid-LRU and Hybrid-ARC in
  - Read hit ratio
  - Write hit ratio
  - I/O throughput

Conclusion

- NVRAM as caching is a challenging and rewarding research topic
- We design Hibachi – a hybrid NVRAM and DRAM cache for disk arrays
  - Characterize storage-level workload to get design guidance
  - Our four features make Hibachi standing out
- Hibachi outperforms existing work in both read and write

References (1/2)

- [1] Z. Fan, D. H. C. Du and D. Voigt, "H-ARC: A non-volatile memory based cache policy for solid state drives," 2014 30th Symposium on Mass Storage Systems and Technologies (MSST), Santa Clara, CA, 2014, pp. 1-11.
- [2] Z. Fan, A. Haghdoost, D. H. C. Du and D. Voigt, "I/O-Cache: A Non-volatile Memory Based Buffer Cache Policy to Improve Storage Performance," 2015 IEEE 23rd International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, Atlanta, GA, 2015, pp. 102-111.
- [3] Z. Fan, F. Wu, D. Park, J. Diehl, D. Voigt and D. H. C. Du , "Hibachi: A Cooperative Hybrid Cache with NVRAM and DRAM for Storage Arrays," 2017 33rd Symposium on Mass Storage Systems and Technologies (MSST), Santa Clara, CA, 2017, pp. 1-11.
- [4] Figure from https://technet.microsoft.com/en-us/enus/library/dd758814(v=sql.100).aspx
- [5] N. Megiddo and D. S. Modha, "Outperforming LRU with an adaptive replacement cache algorithm," in Computer, vol. 37, no. 4, pp. 58-65, April 2004.

References (2/2)

- [6] Y. Zhou, Z. Chen, and K. Li, “Second-level buffer cache management,” IEEE Trans. Parallel Distrib. Syst., vol. 15, pp. 505–519, June 2004.
- [7] G. Yadgar, M. Factor, and A. Schuster, “Karma: Know-it-all replacement for a multilevel cache,” in Proceedings of the 5th USENIX Conference on File and Storage Technologies, FAST ’07, (Berkeley, CA, USA), pp. 25–25, USENIX Association, 2007.
- [8] M. Woods, “Optimizing storage performance and cost with intelligent caching,” tech. rep., NetApp, August 2010.
- [9] Sim-ideal. git@github.com:arh/sim-ideal.git
- [10] Y. Zhou, Z. Chen, and K. Li, “Second-level buffer cache management,” IEEE Trans. Parallel Distrib. Syst., vol. 15, pp. 505–519, June 2004.

Questions?

Ziqi Fan
fanxx234@umn.edu