The Storage Hierarchy is Not a Hierarchy: Optimizing Caching on Modern Devices with Orthus

Kan Wu, Zhihan Guo, Guanzhou Hu, Kaiwei Tu, Ramnatthan Alagappan, Rathijit Sen, Kwanghyun Park, Andrea Arpaci-Dusseau and Remzi Arpaci-Dusseau

* The Storage Hierarchy is Not a Hierarchy: Optimizing Caching on Modern Devices with Orthus (FAST 2021)
Storage Hierarchy is Important

- Simplified two-layer hierarchy
  - **Performance Device**: fast, expensive, small
  - **Capacity Device**: slow, cheap, large

- Caching
  - Replicating popular data in **Performance Device**
Caching Wisdom: **Maximizing Hit Rates**

- Strives to direct most accesses to Performance Device
Caching Wisdom: Maximizing Hit Rates

- Strives to direct most accesses to Performance Device
- Caching delivers ~Performance Device speed along with Capacity Device capacity
- Traditionally, very good!
  - **Performance**: Performance Device >> Capacity device
  - E.g. DRAM vs. HDD (100x differences)
Problem: Caching is **Insufficient** in **Modern** Storage Hierarchies

- **Insight:**
  
  the assumption (Performance device $>>$ Capacity device) is broken
The Modern Storage Hierarchies

- Non-Volatile Memory-based devices are filling the performance gap
The Modern Storage Hierarchies

- Non-Volatile Memory-based devices are filling the performance gap
  - NVDIMM (300ns, ~7GB/s)
  - Low-latency SSD (10us, ~3GB/s)
The Modern Storage Hierarchies

- Non-Volatile Memory-based devices are filling the performance gap
  - NVDIMM (300ns, ~7GB/s)
  - Low-latency SSD (10us, ~3GB/s)

- The differences between today’s neighboring layers are less clear and even overlapping (depending on workloads)
The Modern Storage Hierarchies

- Non-Volatile Memory-based devices are filling the performance gap
  - NVDIMM (300ns, ~7GB/s)
  - Low-latency SSD (10us, ~3GB/s)

- The differences between today’s neighboring layers are less clear and even overlapping (depending on workloads)
The Modern Storage Hierarchies

- Non-Volatile Memory-based devices are filling the performance gap
  - NVDIMM (300ns, ~7GB/s)
  - Low-latency SSD (10us, ~3GB/s)

- The differences between today’s neighboring layers are less clear and even overlapping (depending on workloads)

- **Problem:** caching leaves huge performance available in capacity layers **unexploited**
Our Approach: Non-Hierarchica Caching (NHC)

- **Key idea:** augmenting caching with dynamic load admission and request offloading
  - Exploit the available performance in capacity devices during runtime
Design: Non-Hierarchical Caching

- Enable offloading: **tunable caching behaviors**
  - Classic caching is \((data\_admit = true, load\_admit\_ratio = 100%)\)
Design: Non-Hierarchical Caching

- feedback-based cache scheduler
  - Adjust tuning knobs (e.g., data_admit flag, load_admit ratio)
Design: Non-Hierarchical Caching

- feedback-based cache scheduler
  - Optimize a target performance metric
  - Target metric: user/device; throughput/ latency/ tail latency
    - \( f(X) \): a function to measure/compute the target metric
Peek into Evaluation

● NHC can exploit considerable performance in modern capacity layers
  ○ Up to 2X performance improvement in
    ■ DRAM/ NVDIMM
    ■ NVDIMM/ Optane SSD
    ■ Optane SSD /Flash
    ■ ...
● NHC is able to optimize **throughput, avg. latency**, and **tail latency**
● NHC is also able to adapt to complex, dynamic workloads such as Facebook ZippyDB workloads
Conclusion

● Evolving storage hierarchies have strong implications for caching
  ○ Quantitative comparisons across modern storage devices
  ○ Characterizing caching performance in both classic and modern hierarchies

● NHC optimizes classic caching, by dynamic load admission and request offloading
  ○ Is compatible with all classic caching policies
  ○ Requires no prior knowledge of devices and workloads
  ○ Adapts to dynamic workloads

Please check paper for more details
Thank you & Questions?

Contact: kanwu@cs.wisc.edu