Caching on PMEM: An Iterative Approach

Flash Memory Summit

Yao Yue (@thinkingfish)
Twitter

Caching at Twitter

Testing at Intel

Testing at Twitter

A New Design

Flash Memory Summit

Caching at Twitter

Flash Memory Summit

Caching at Twitter

Clusters
>300 in prod

Hosts
many thousands

Instances
tens of thousands

Job size
2-6 core, 4-48 GiB

Provisioning

QPS
max 50M (single cluster)

SLO
p999 < 5ms*

Performance

Maintainable
- Same codebase
- Retain high-level APIs

Operable
- Flexible configuration
- Predictable performance

Constraints

A large scale analysis of hundreds of in-memory cache clusters at Twitter [OSDI'20]

Quick Cache Facts

Memory hungry

Throughput bottleneck on host networking stack

Mission critical ⇒ *availability*

Large resource footprint ⇒ *cost*

Lots of instances ⇒ *fast restart*

Why PMEM?

- **Cache more data per instance**
  - Reduce TCO if memory bound
  - Improve hit rate

- **Take advantage of persistency**
  - Graceful shutdown and faster rebuild
  - Improve data availability during maintenance

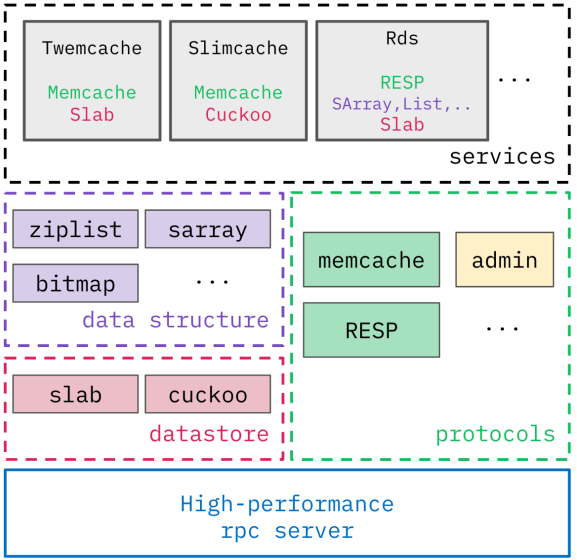
availability

cost

fast restart

Master Plan

- Use a modular framework
- Test
  - Intel's equipment
  - Twitter's equipment
- A New Design



Pelikan: A Modular Cache

Test Setup

Instance density

18-30 instances / host

Object size

64 - 2048 bytes

Dataset size

4GiB - 32 GiB / instance

# of Connection / instance

100 / 1000

Focus on…

- Latency first, throughput second
- PMEM vs. DRAM
- Memory mode vs. AppDirect

Understand…

- Scalability with dataset size
- Bottleneck analysis

Testing at Intel

Flash Memory Summit

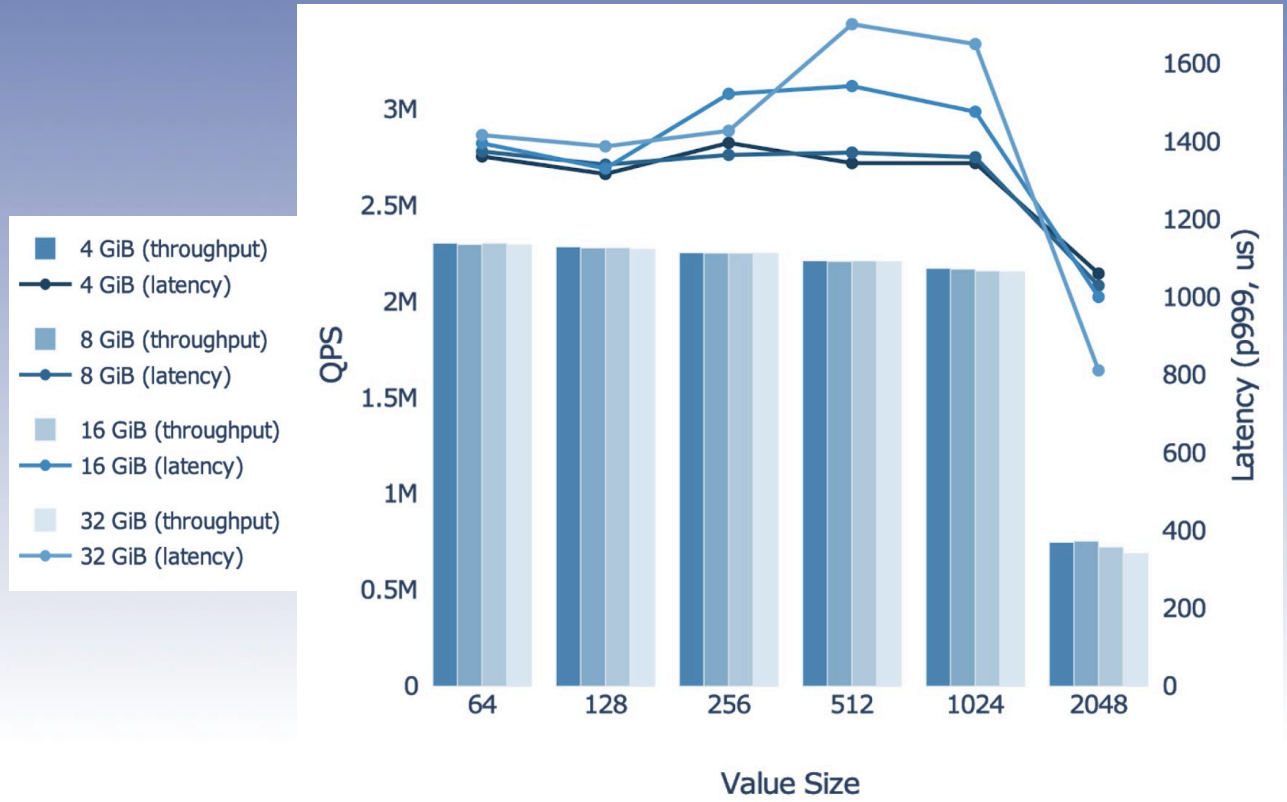
Intel: Memory Mode, Unmodified Code

Hardware Config (Intel lab)

- 2 X Intel Xeon 8160 (24)
- 12 X 32GB DIMM
- **12 X 128GB AEP**
- 2-2-2 config
- 1 X 25Gb NIC

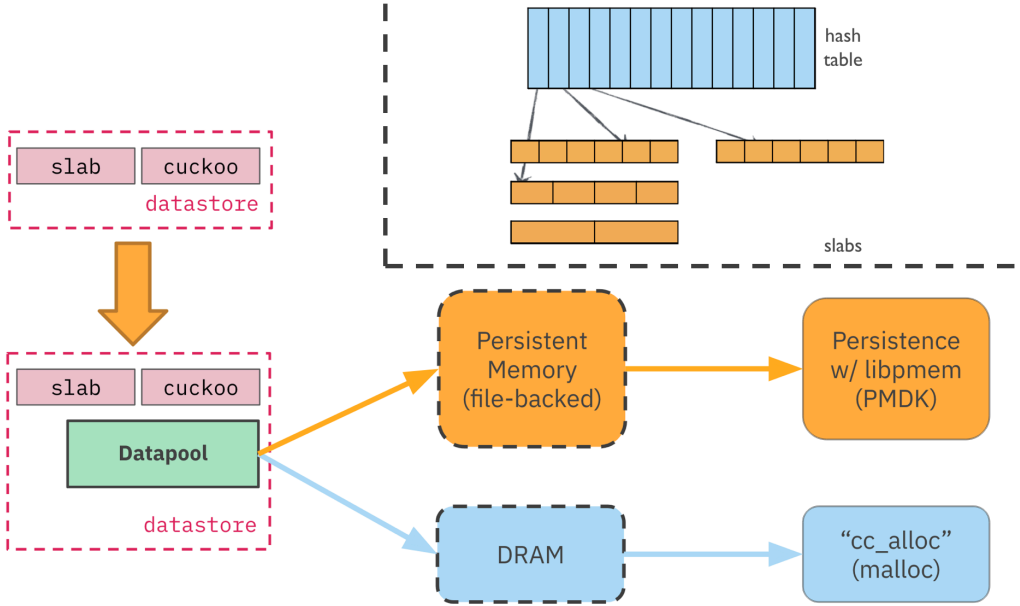
Test Config

- 30 jobs/host
- key size 32B
- 100 conn/job
- 90R:10W



Intel: AppDirect Mode, *Slightly* Modified Code

~300 LOC in C



Datapool Abstraction with PMDK

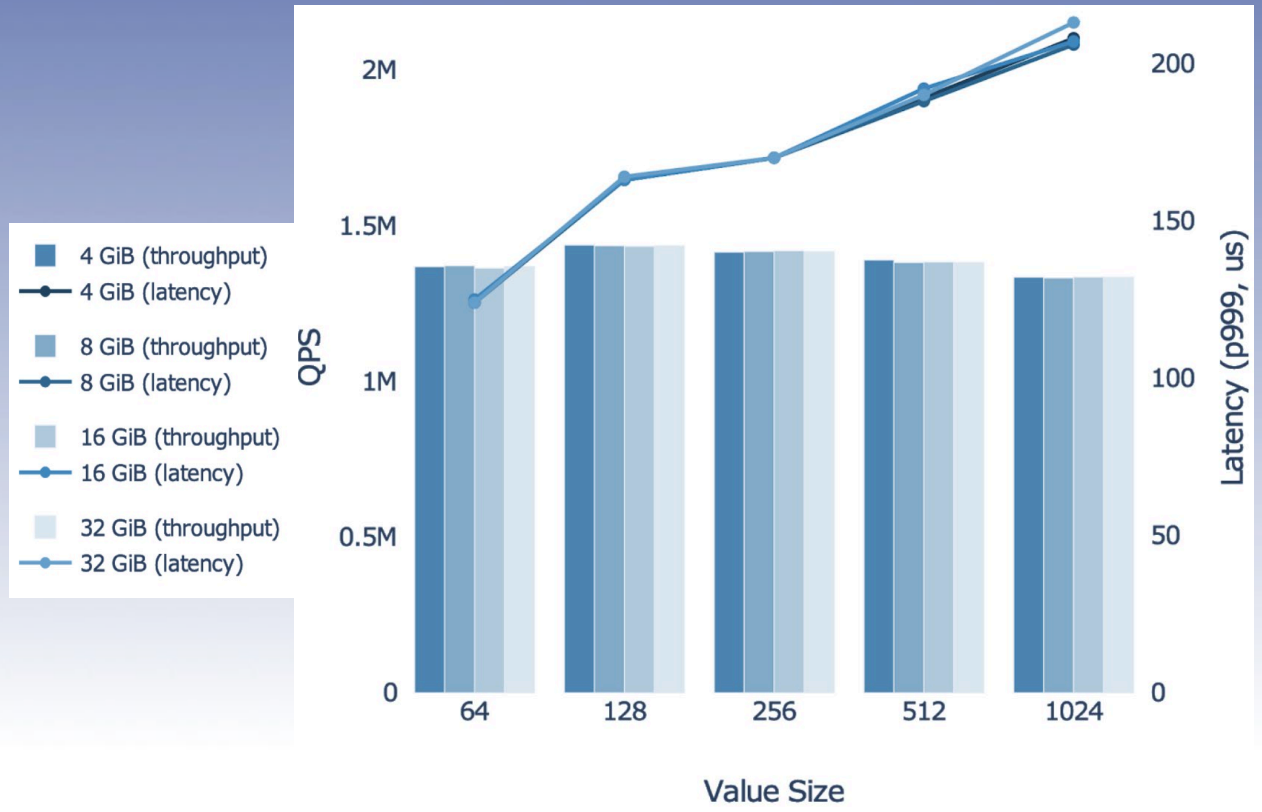
Intel: AppDirect Mode, *Slightly* Modified Code

Hardware Config (Intel lab)

- 2 X Intel Xeon 8160 (24)
- 12 X 32GB DIMM
- **12 X 128GB AEP**
- 2-2-2 config
- 1 X 25Gb NIC

Test Config

- 24 jobs/host
- key size 32B
- 100 conn/job
- 90R:10W



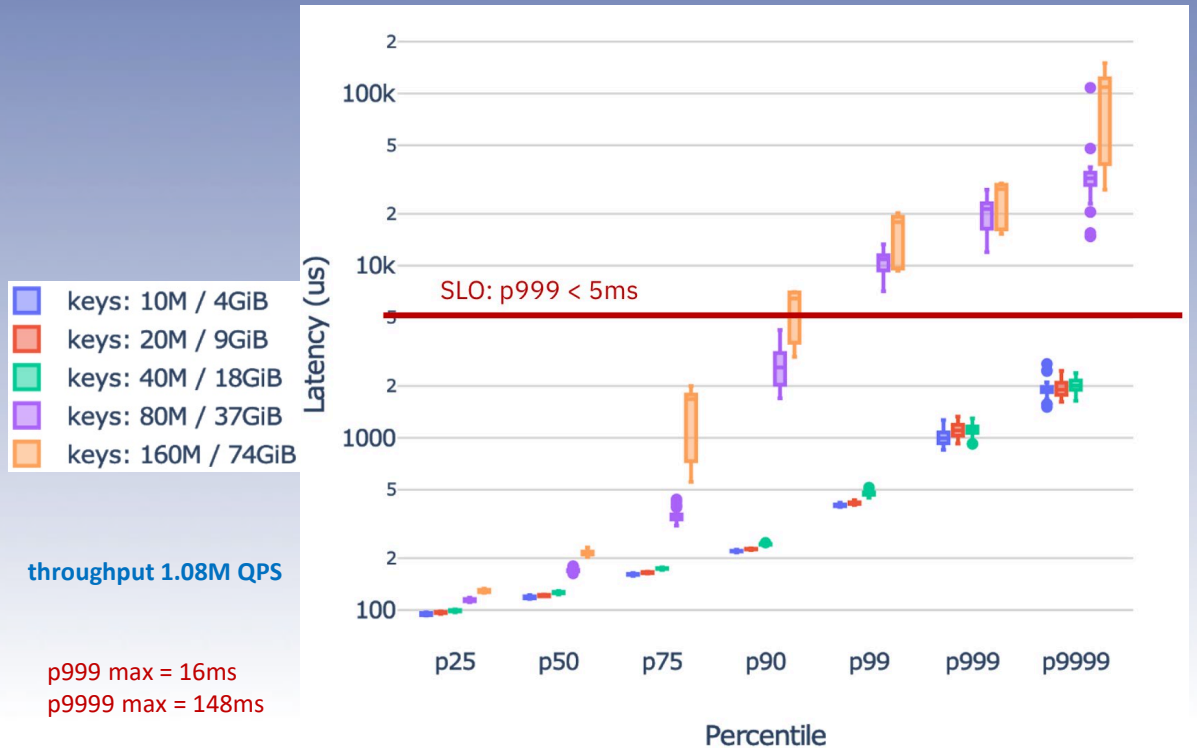
Testing at Twitter

Flash Memory Summit

Twitter: memory mode, SLO

Hardware Config (Twitter prod)
- 2 X Intel Xeon 8160 (20)
- 12 X 16GB DIMM
- **4 X 512GB AEP**
- 2-1-1 config
- 1 X 25Gb NIC

Test Config
- 20 jobs/host
- key size 64B
- **1000 conn/job**
- Read-only



throughput 1.08M QPS

p999 max = 16ms
p9999 max = 148ms

Twitter: AppDirect mode, SLO

Hardware Config (Twitter prod)

- 2 X Intel Xeon 8160 (20)
- 12 X 16GB DIMM
- **4 X 512GB AEP**
- 2-1-1 config
- 1 X 25Gb NIC

Test Config

- 20 jobs/host
- key size 64B
- **1000 conn/job**
- Read-only

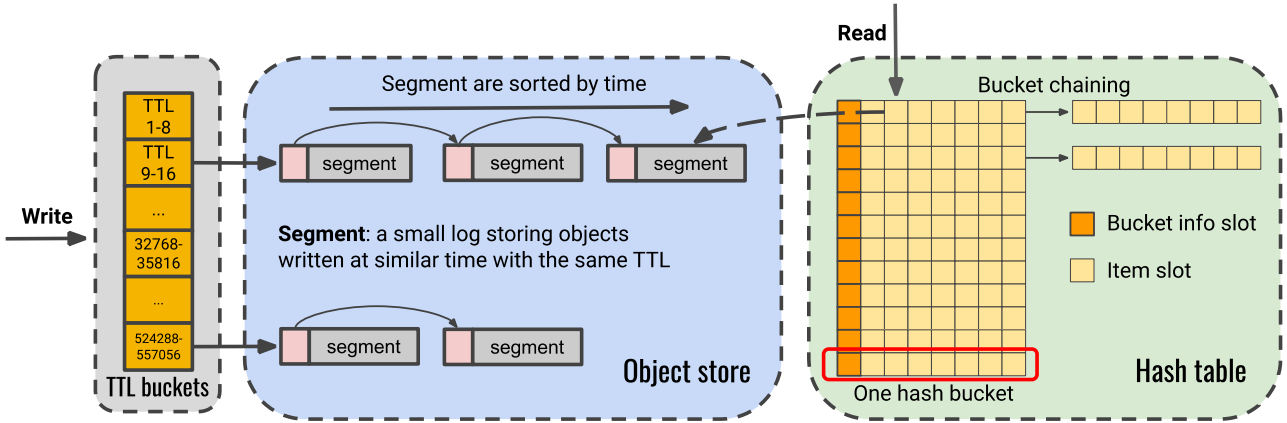
throughput 1.08M QPS

p999 max = 1.4ms
p9999 max = 2.5ms

A New Design

Segcache: Log-like storage, compact hash table

- Segment-based storage (inspired by log structure)
- Hash table: multi-occupancy block, chained



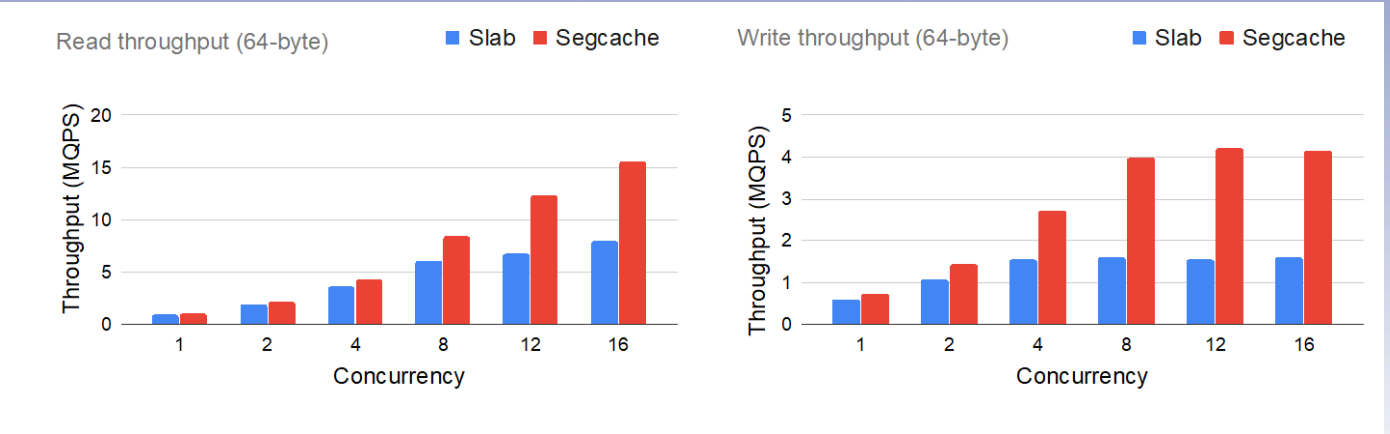
Goals

- Minimize random reads
- Minimize random writes
- Minimize metadata overhead

Segcache: AppDirect mode, Microbenchmark

Hardware Config (Twitter prod)

- 2 X Intel Xeon 8160 (20)
- 12 X 16GB DIMM
- **4 X 512GB AEP**
- 2-1-1 config
- 1 X 25Gb NIC



Caching on PMEM: Takeaways

- Avoid turning PMEM into new bottleneck
- AppDirect is a clear winner
  - But Memory Mode served its purpose along the way
- Due diligence pays off
- Innovate as needed

- Future work: graceful shutdown and fast rebuild

Thank You!

Flash Memory Summit

https://pelican.io

@pelican_cache