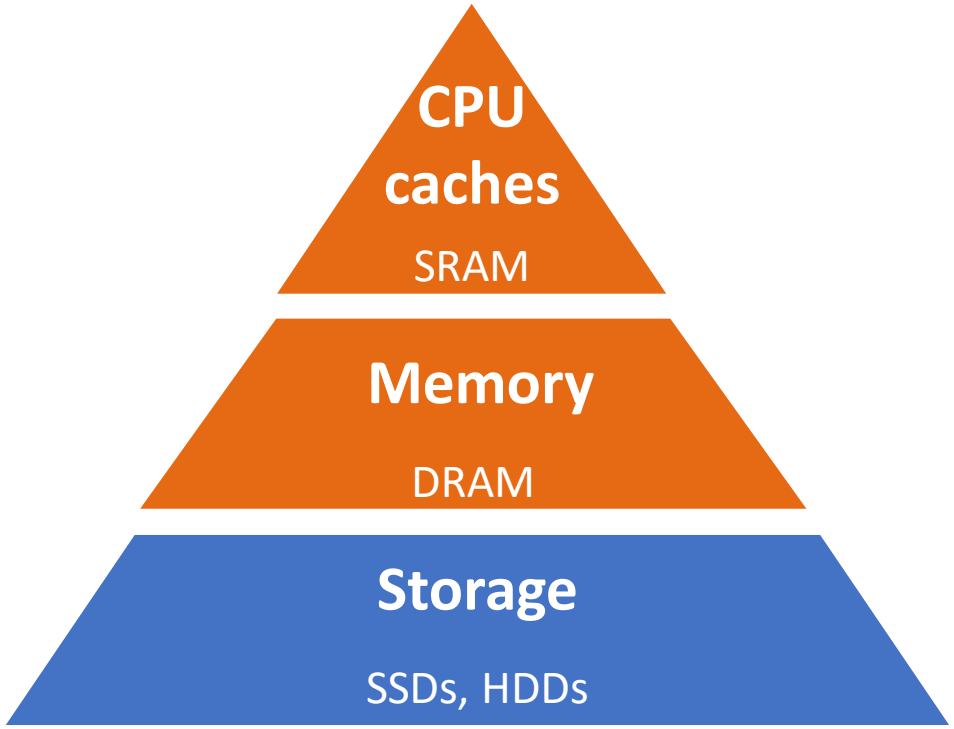
SSDs Striking Back: The Storage Jungle and Its Implications on Persistent Indexes

Kaisong Huang Darien Imai **Tianzheng Wang** Dong Xie

SFU SIMON FRASER UNIVERSITY

PennState

The storage hierarchy as we knew it



Layers with clear boundaries
Memory: fast but volatile
Storage: slower than memory but persistent

Caching stores hugely successful

- Hot (index) pages in buffer pool (DRAM)
- Persist to SSDs
- Cost-effective

…is being disrupted, by two trends

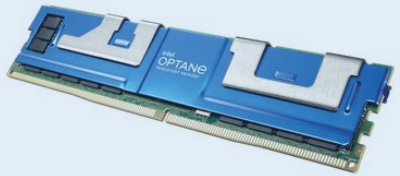
Disruptor 1: Persistent Memory (PM)

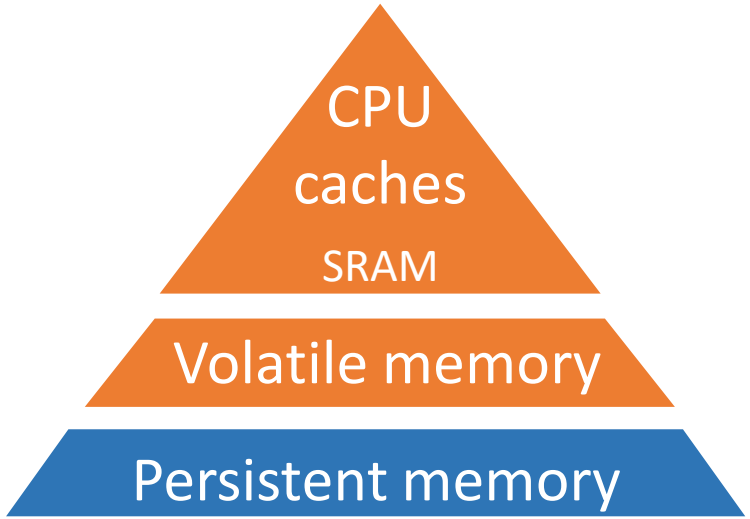
Persistent memory, generally speaking

- Byte addressable
- Persistence
- Large capacity
- Cheaper than DRAM
- Load/store instructions to access data

Intel Optane DCPMM 200 based on 3D XPoint

- Peak read: 7.4 GB/s per DIMM
- Peak write: 2.3 GB/s per DIMM
- Capacity: 128/256/512 GB per DIMM





Memory != volatile

“PM camp”
(a lot of attention)

Buffer pool + SSD

Single-level index/store

“SSDs no more, cheaper than DRAM – all in!”

Disruptor 2: Modern SSDs are Fast

3D V-NAND Flash or 3D XPoint

- New interconnect
  - NVMe Gen4
- New software stack
  - SPDK, io_uring

→ > 10M IOPS on one core!

Intel Optane DC SSD P5800X

- Peak read: 7.4 GB/s
- Peak write: 7.4 GB/s
- Capacity: 400/800/1600GB x # drives

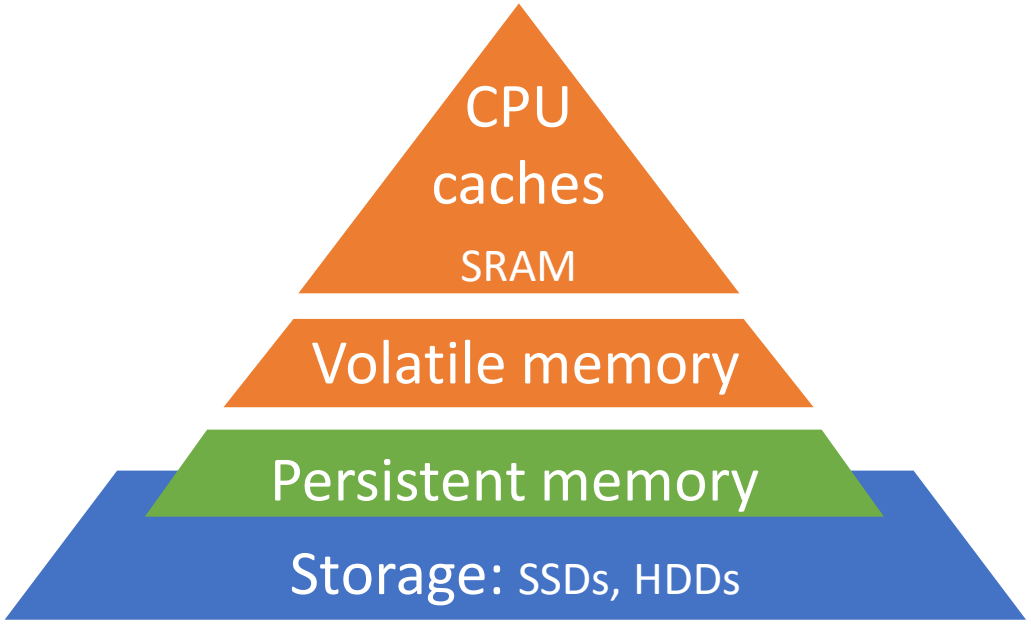
vs.

Optane DCPMM 200 (128GB DIMM)

- Peak read: 7.4 GB/s
- Peak write: 2.3 GB/s
- Capacity: 128GB x # memory channels

SSD approaches (persistent) memory

The Storage Jungle



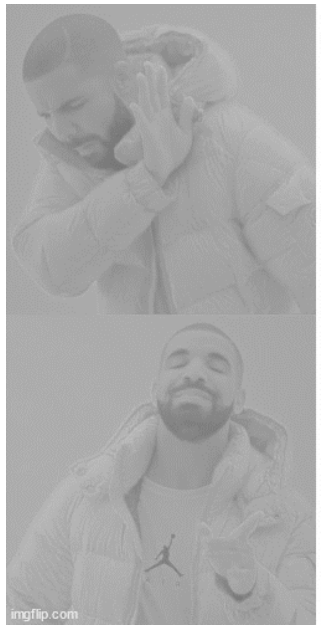
Layers with overlapping properties

Memory not necessarily volatile

Storage not necessarily slower than memory

Want to build an OLTP index?

“PM camp”
(a lot of attention)



Buffer pool + SSD

Single-level index/store

“How much $$$?”

“SSD camp”
(relatively quieter)

Single-level index/store

~In-memory performance atop SSD

“With fast SSDs, match or outperform PM index?”

PM vs. SSD Servers: What (costs) to consider

$$\frac{\texttt{Cost}}{\text{GB}} = \frac{\texttt{CPU + DRAM + Storage}}{\text{Capacity (GB)}}$$

Rigid installation requirements

- Strict population rules
  - >= 1 DRAM DIMM per controller

→ Overprovisioning

- Clock down frequency

→ Affect overall memory performance

Non-trivial CPU cost

- Synchronous `load/store`

→ High-end CPU cores wasted

Flexible installation requirements

- DRAM requirement decoupled
- Few population rules (e.g., RAID)

→ Nothing overprovisioned

Low CPU cost

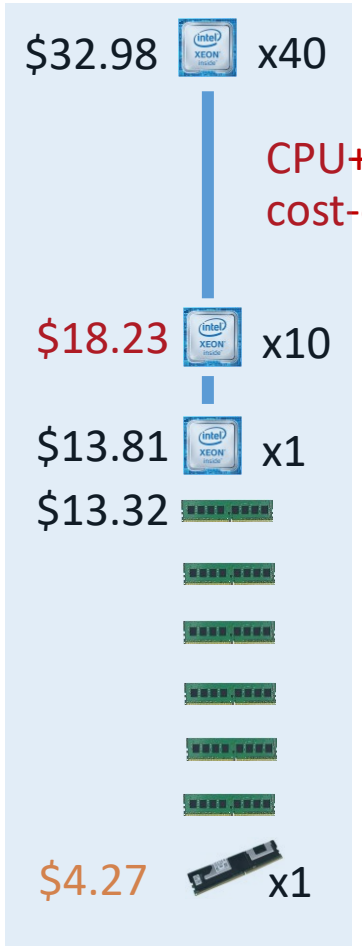
- Asynchronous DMA

→ Overlap I/O and compute

PM vs. SSD Servers: How the costs stack up

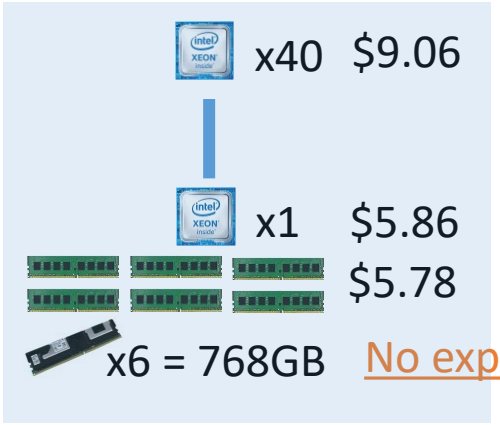
$$ Per GB

PM1: $4,221.69



More DCPMMs
→ lower $/GB

PM6: $6955.44



Same material, but more expensive than SSD

$$Cost = CPU + DRAM + Storage$$

CPU	DRAM	Storage
Xeon Gold 6242R	6x32GB DDR4	128GB DCPMM 100 *or* 375GB P4800X

P4800X1: $4,673.94

Very expandable + linear price scalability

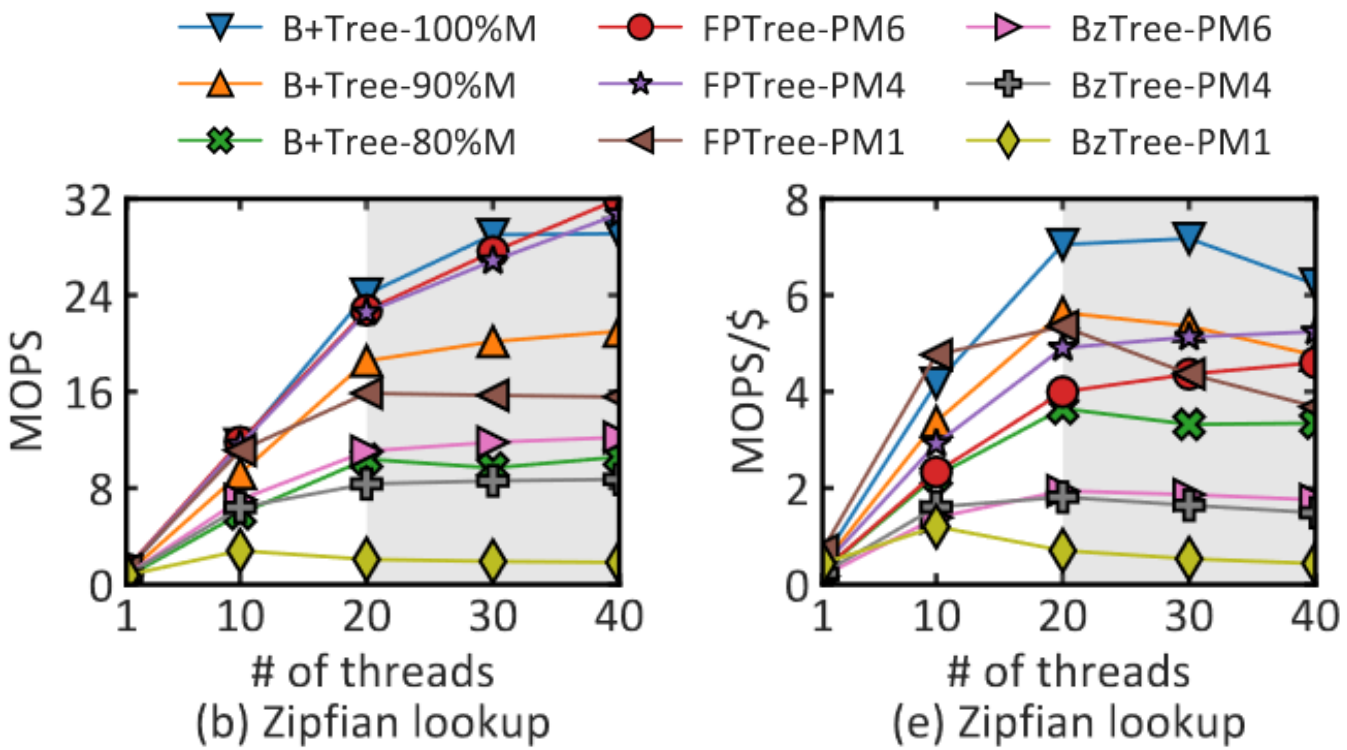
Performance/$

FPTree/BzTree:

Tailor-made, optimized for PM

B+Tree:

Coursework-grade (!) atop P4800X



- Memory-resident? Use SSD + buffer pool
- P4800X1 competitive with PM1
  - Future work: interleaving both SSD and PM

(more details in paper)

Implications and Outlook

- PM hardware still too expensive; PM software stack also “expensive”
  - High-end CPU cores for “I/O” + DRAM costs
  - Learn to program it
  - Wishes: lower $/GB + lift population restrictions

- (Modern) SSDs are great, use them!
  - Before “all-in” PM
  - Usually more cost effective
    - Even with suboptimal implementation
  - Many exciting directions with new data structure designs (SPDK, ZNS, etc.)
  - Unless your application needs PM (seems niche)

Thank you!