An Empirical Guide to Scalable Persistent Memory

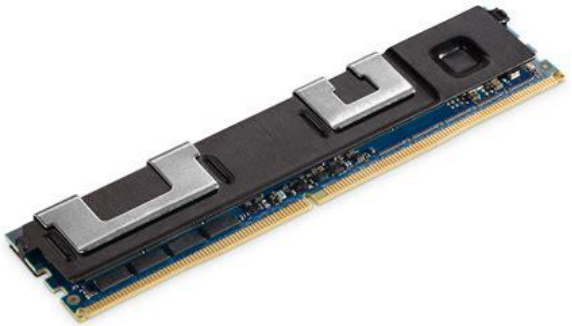
Joseph Izraelevitz, Jian Yang, Lu Zhang, Juno Kim, Xiao Liu, Amirsaman Memaripour, Yun Joon Soh, Zixuan Wang, Yi Xu, Subramanya R. Dulloor, Jishen Zhao, Steven Swanson

Non-Volatile Systems Laboratory
Department of Computer Science & Engineering
University of California, San Diego

Department of Electrical, Computer & Energy Engineering
University of Colorado, Boulder

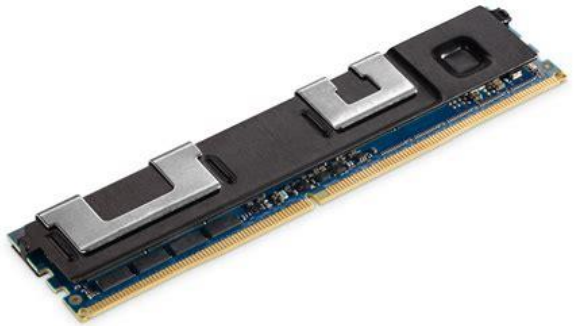
NVSL

Optane DIMMs are Here!



Optane DIMMs are Here!

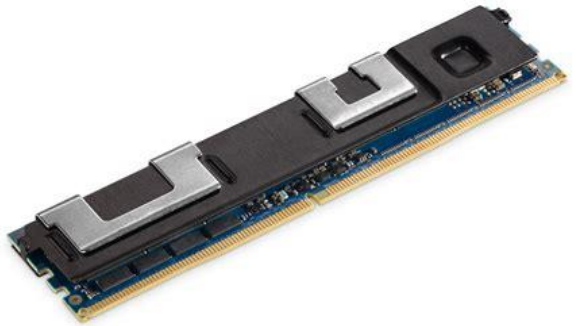
- Not Just Slow Dense DRAM™



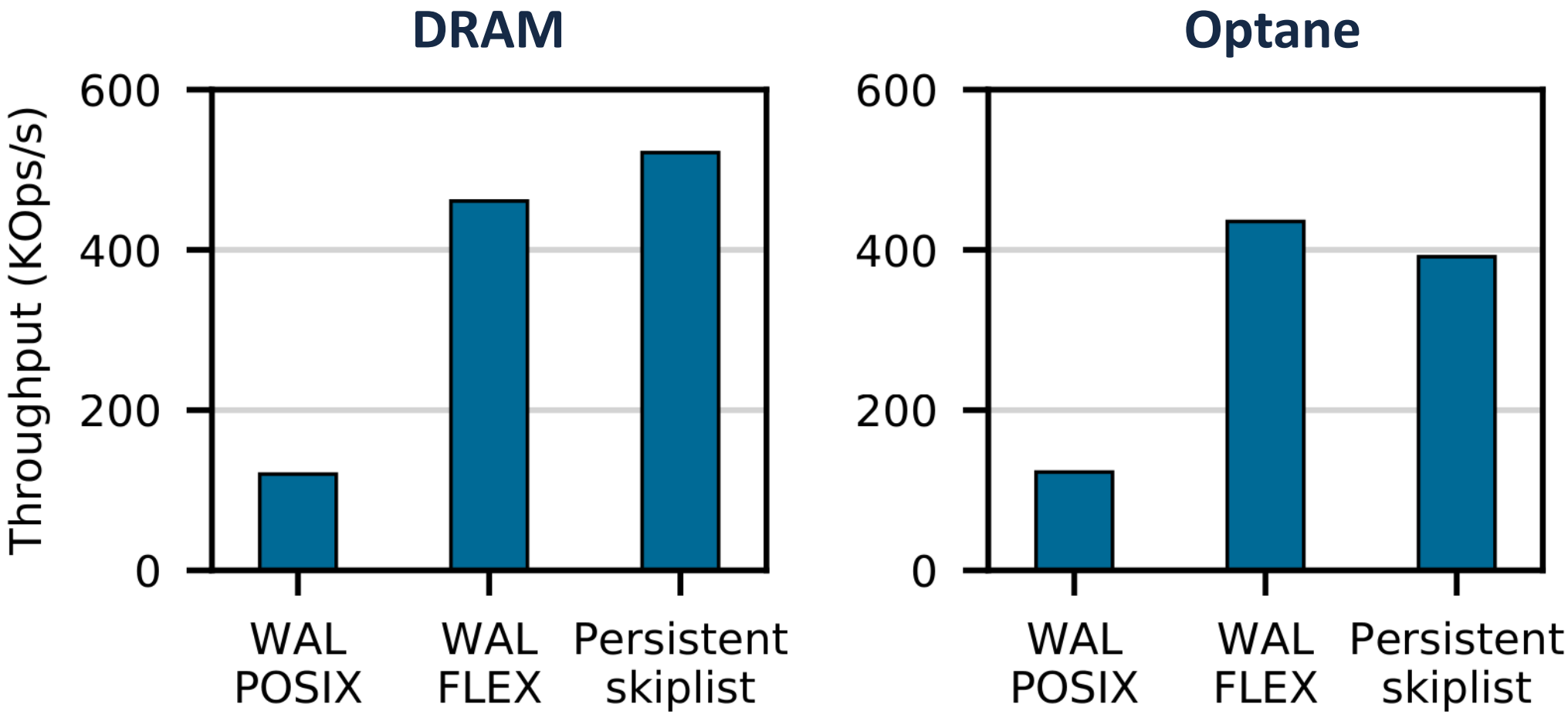
Optane DIMMs are Here!

- Not Just Slow Dense DRAM™

- Slower, denser, media
  - -> More complex architecture
    - -> Second-order performance anomalies
      - -> Fundamentally different

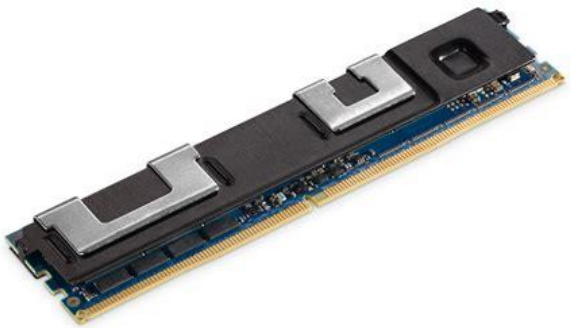


Basics: Emulation is misleading



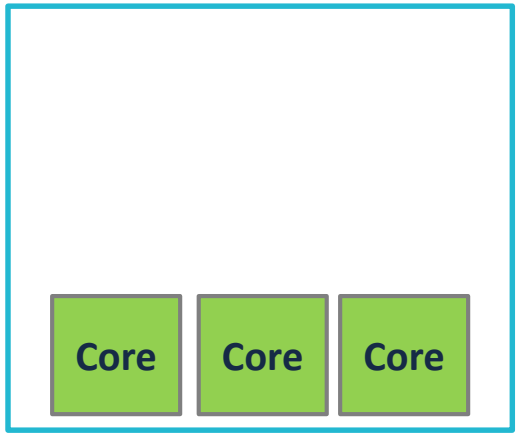
Outline

- Background
- Basics: Optane DIMM Performance
- Lessons: Optane DIMM Best Practices
- Conclusion

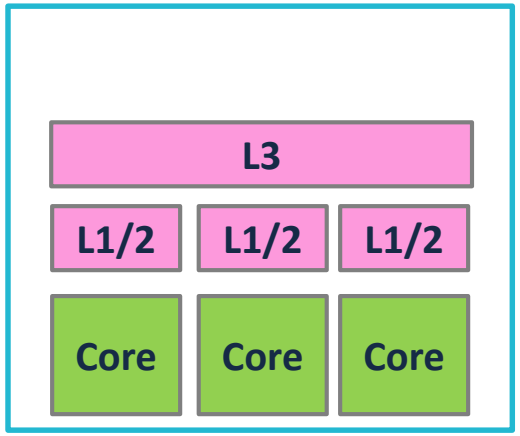


Background

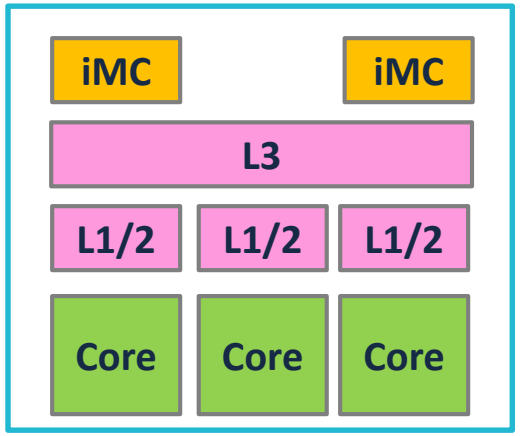
Background: Optane in the Machine



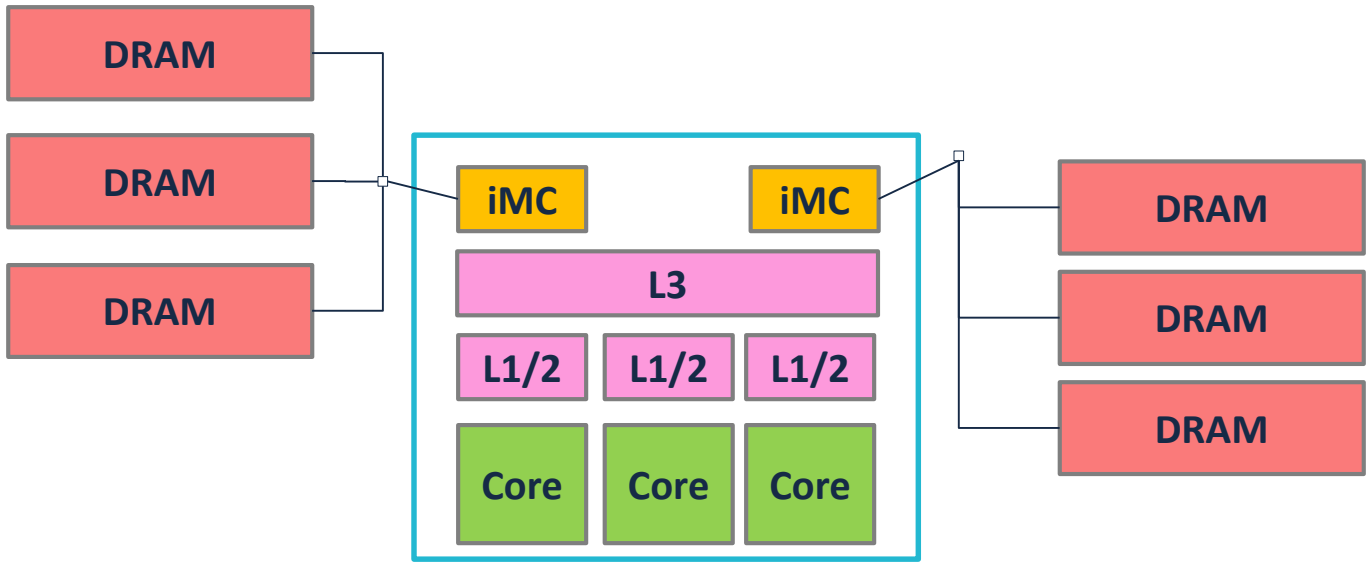
Background: Optane in the Machine



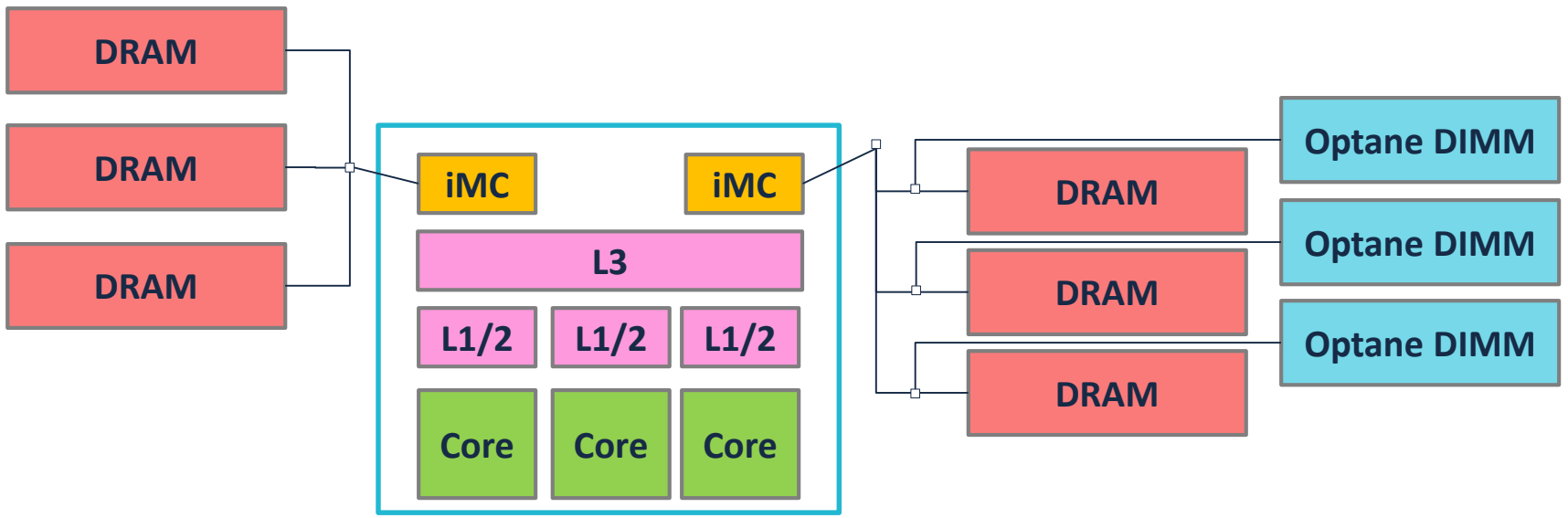
Background: Optane in the Machine



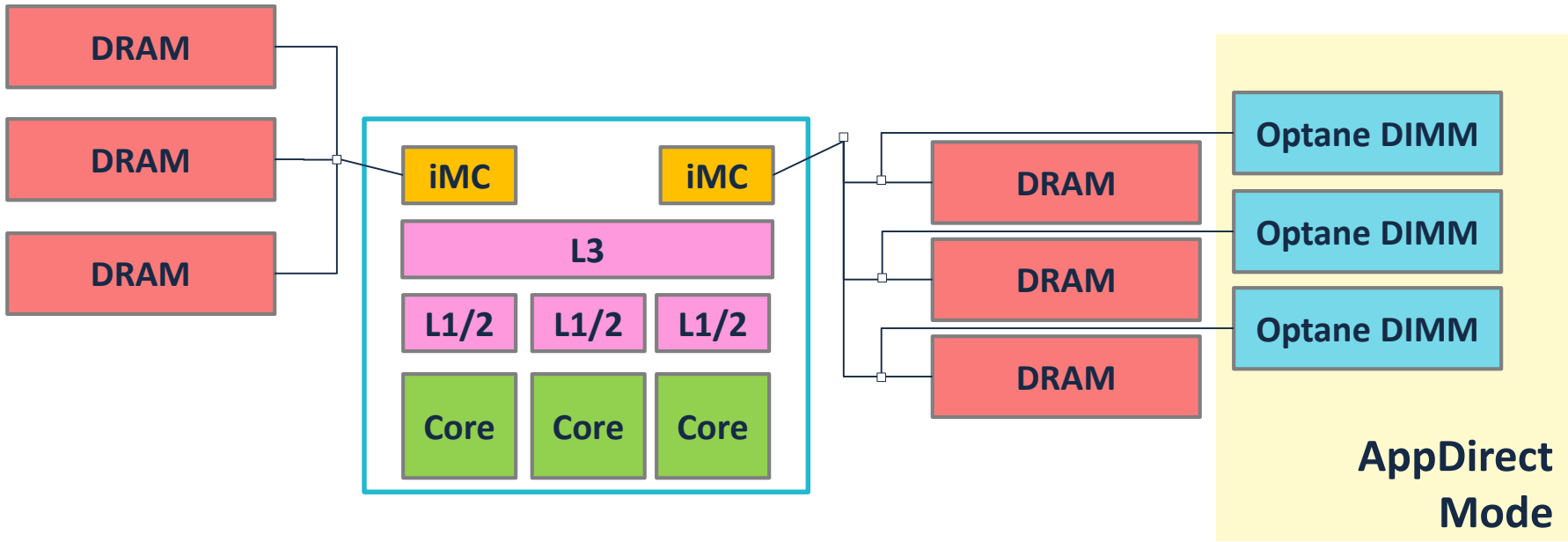
Background: Optane in the Machine



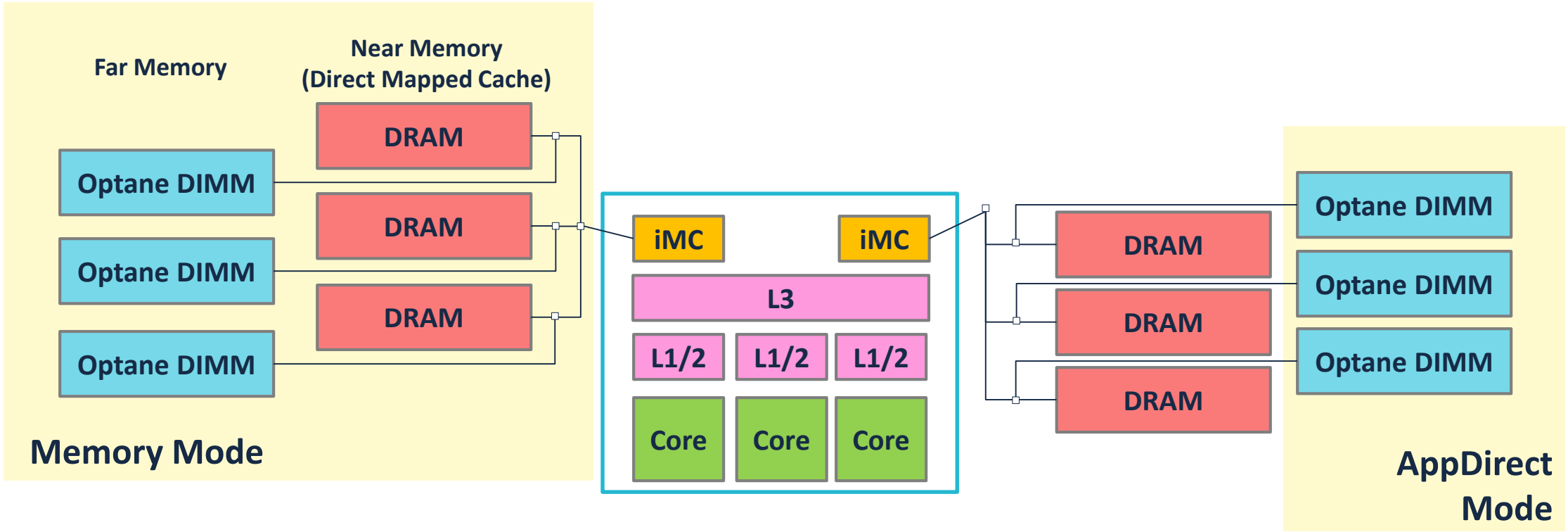
Background: Optane in the Machine



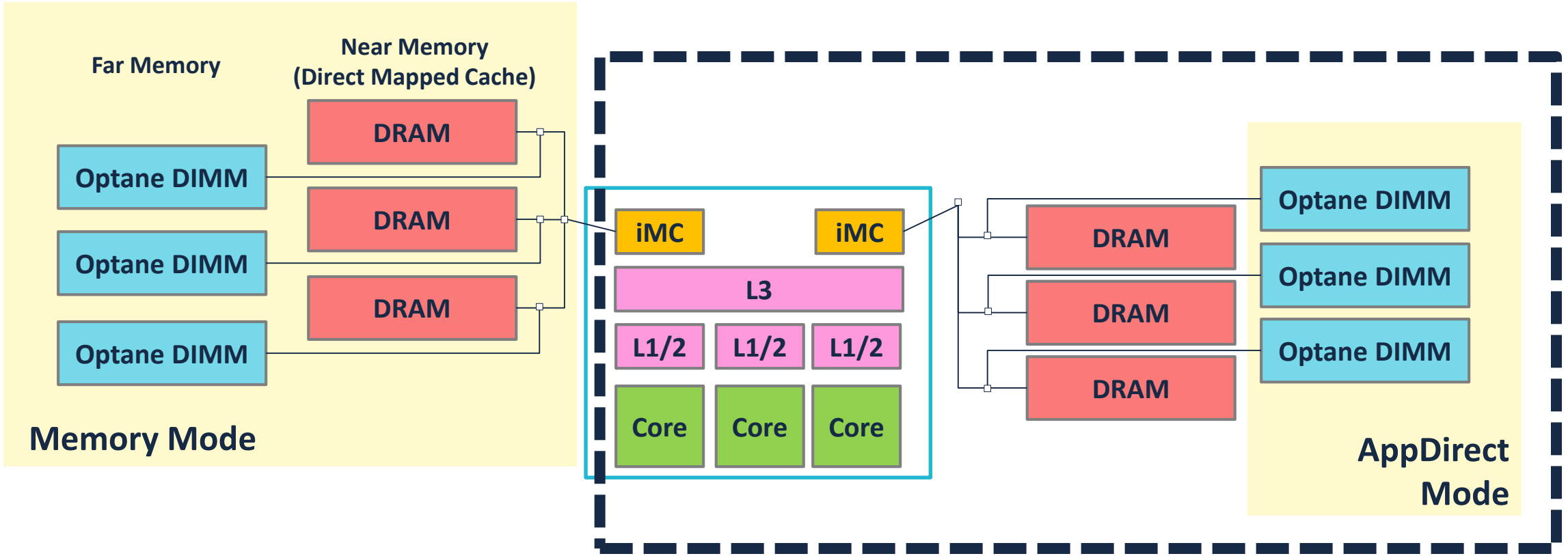
Background: Optane in the Machine



Background: Optane in the Machine

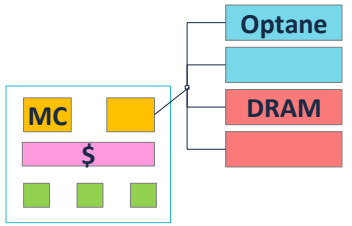


Background: Optane in the Machine

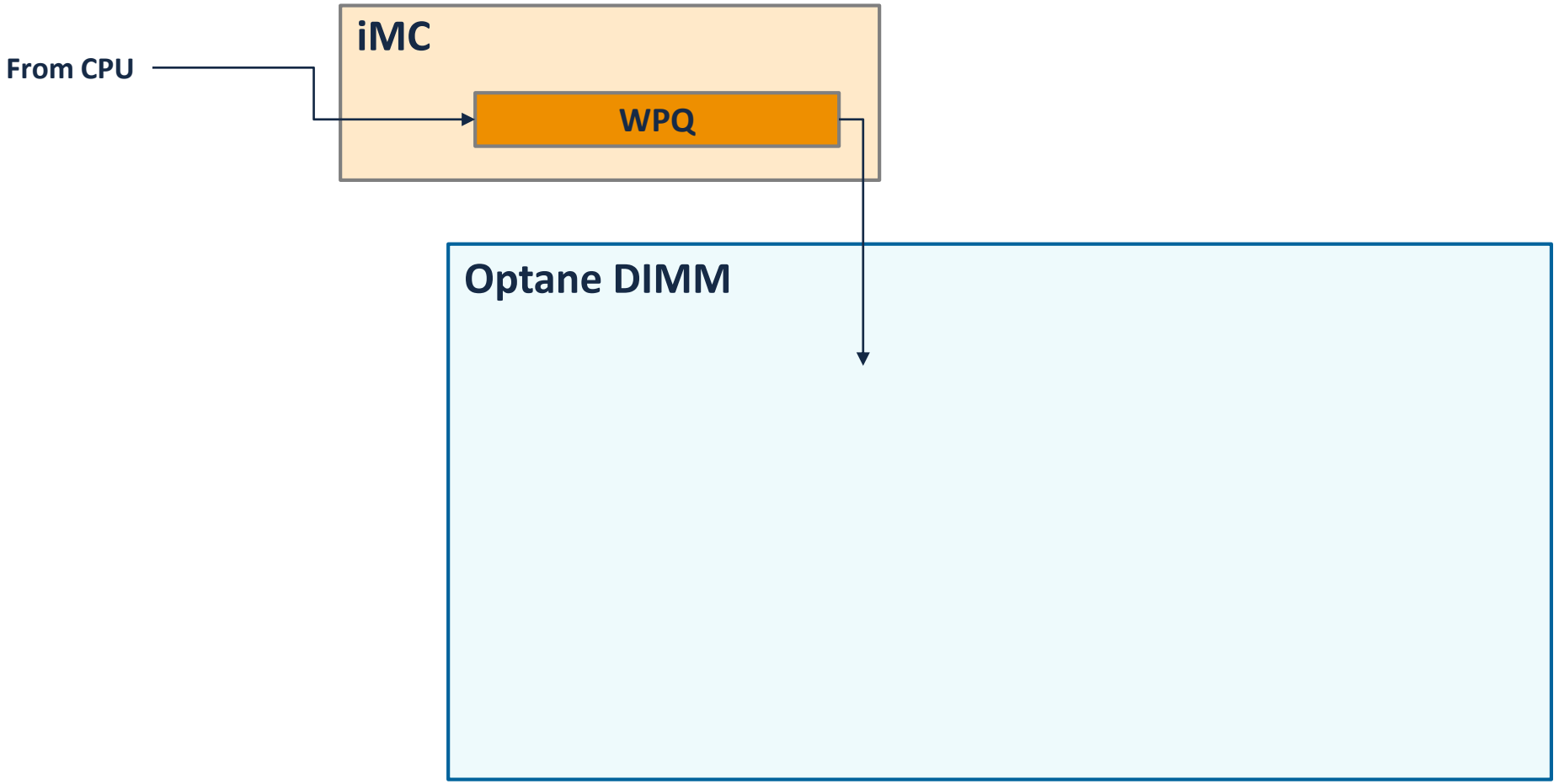


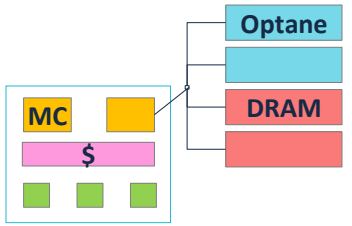
Background: Optane Internals

Optane DIMM

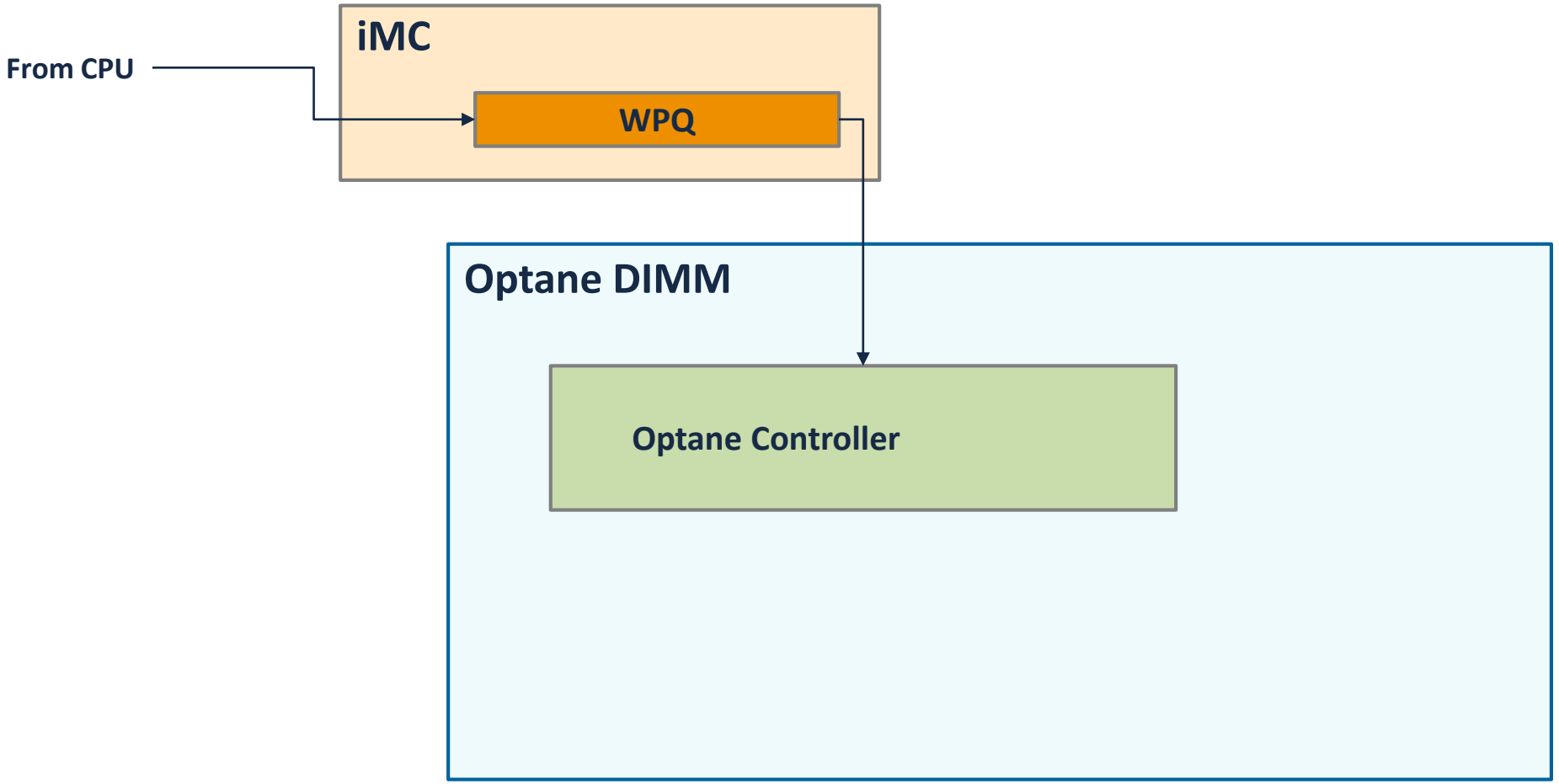


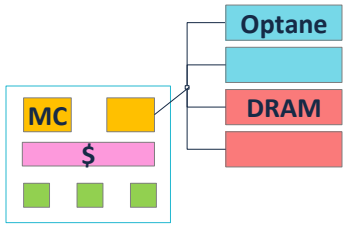
Background: Optane Internals



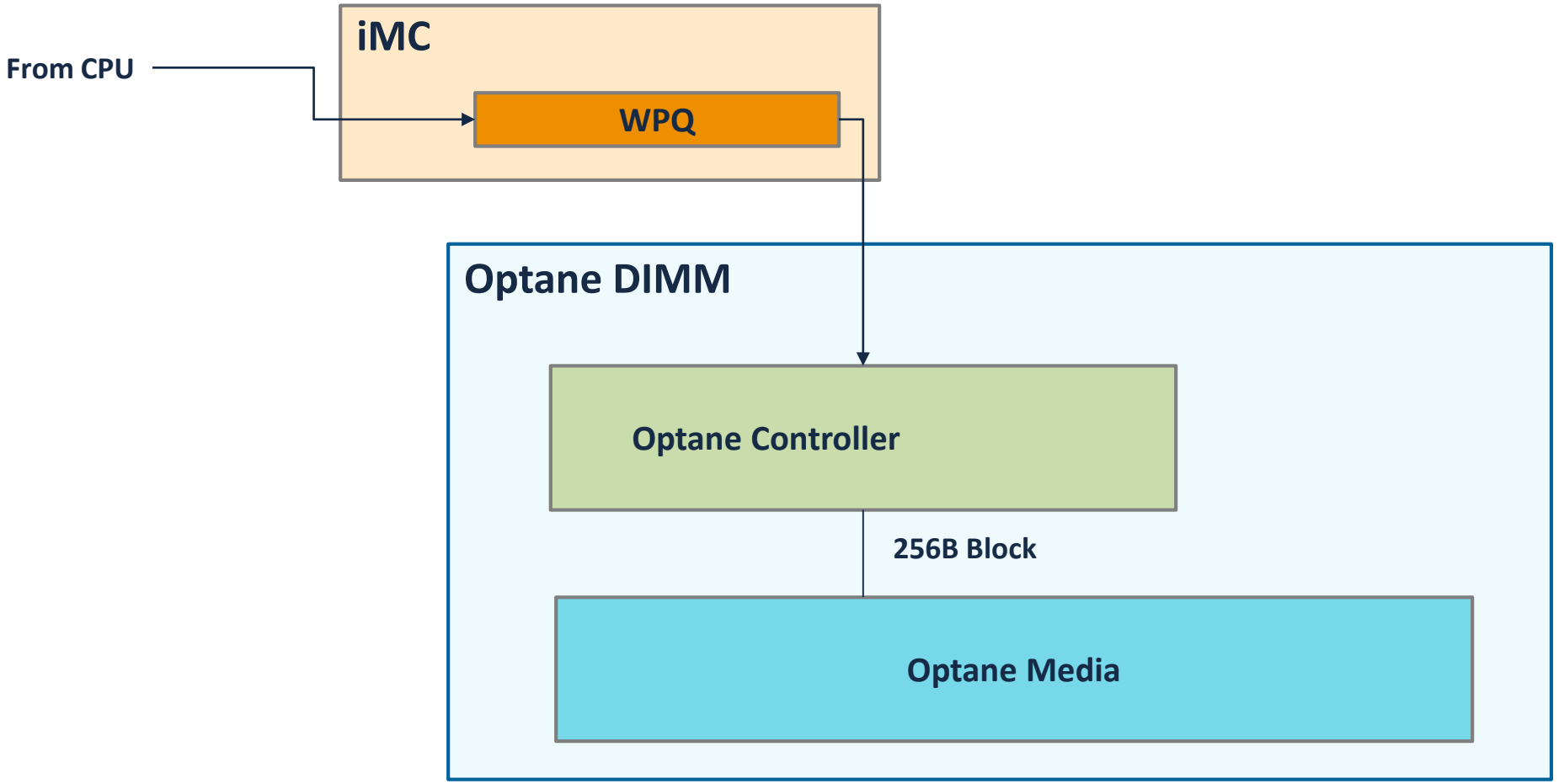


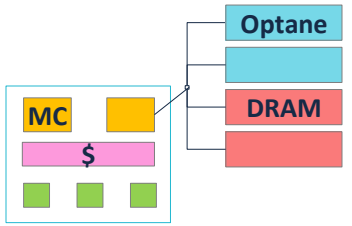
Background: Optane Internals



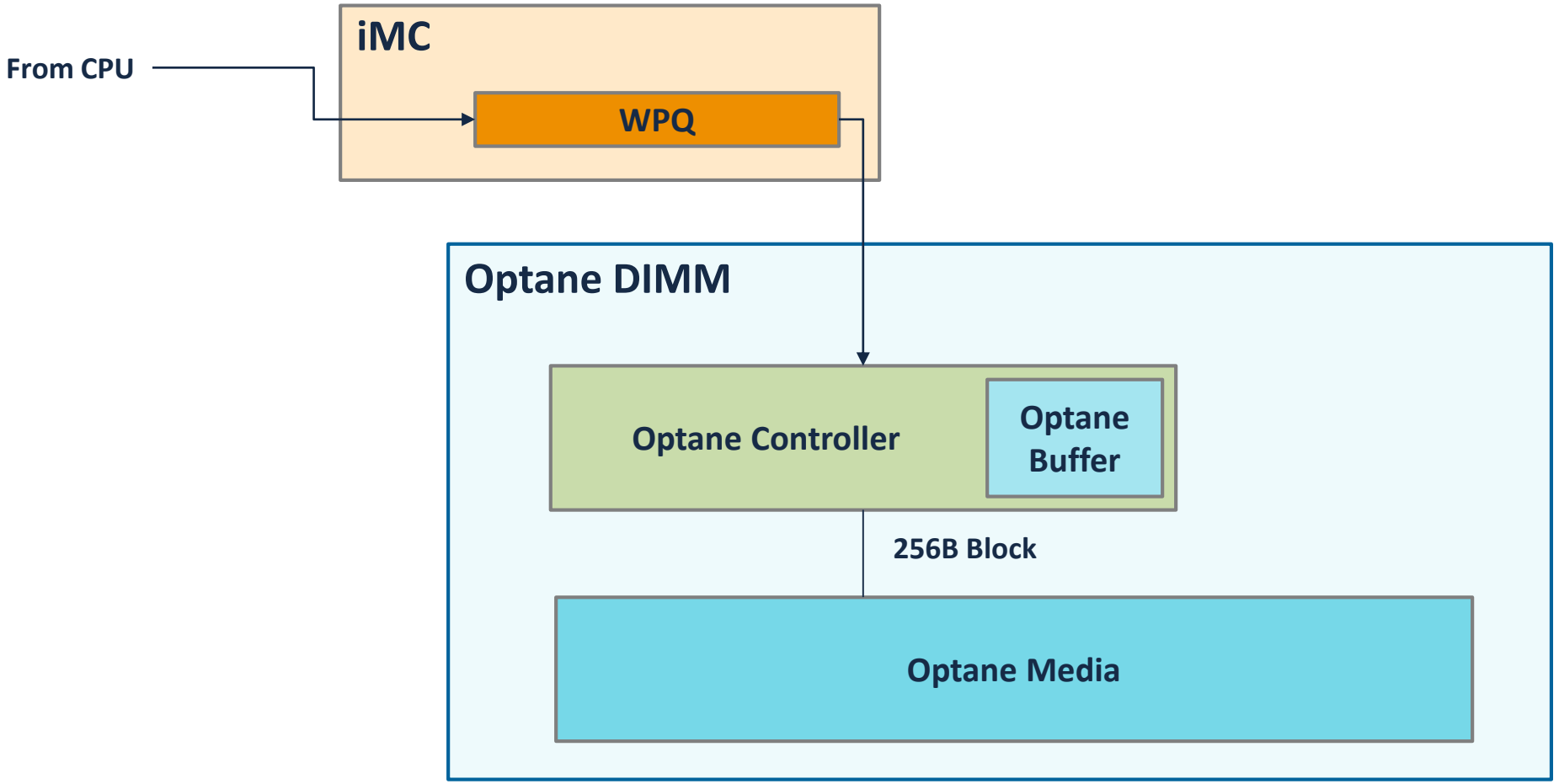


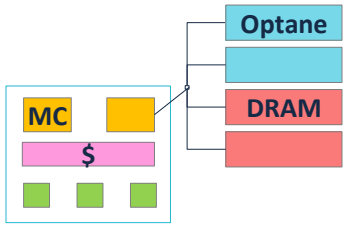
Background: Optane Internals



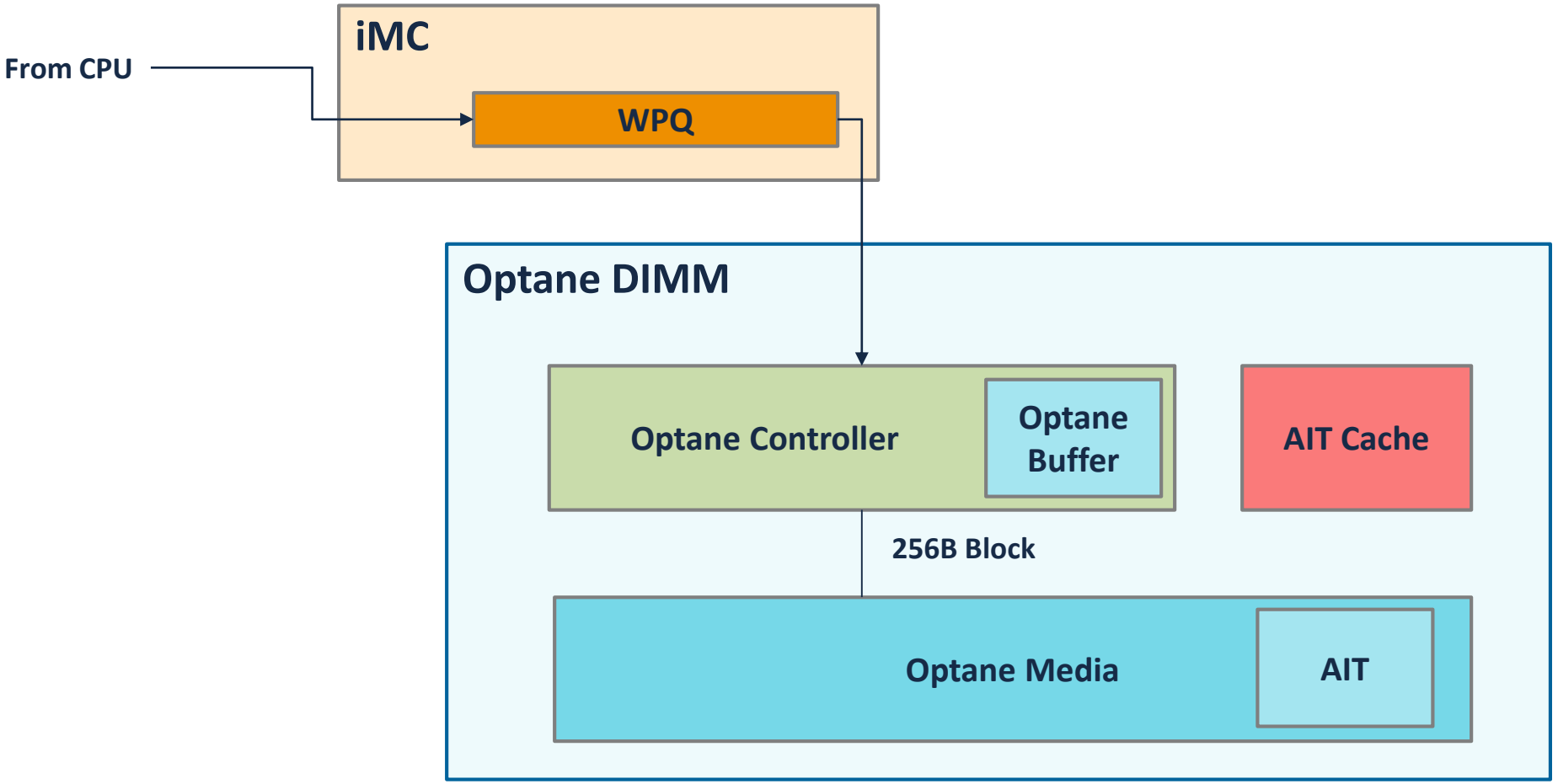


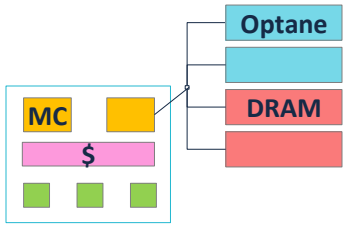
Background: Optane Internals



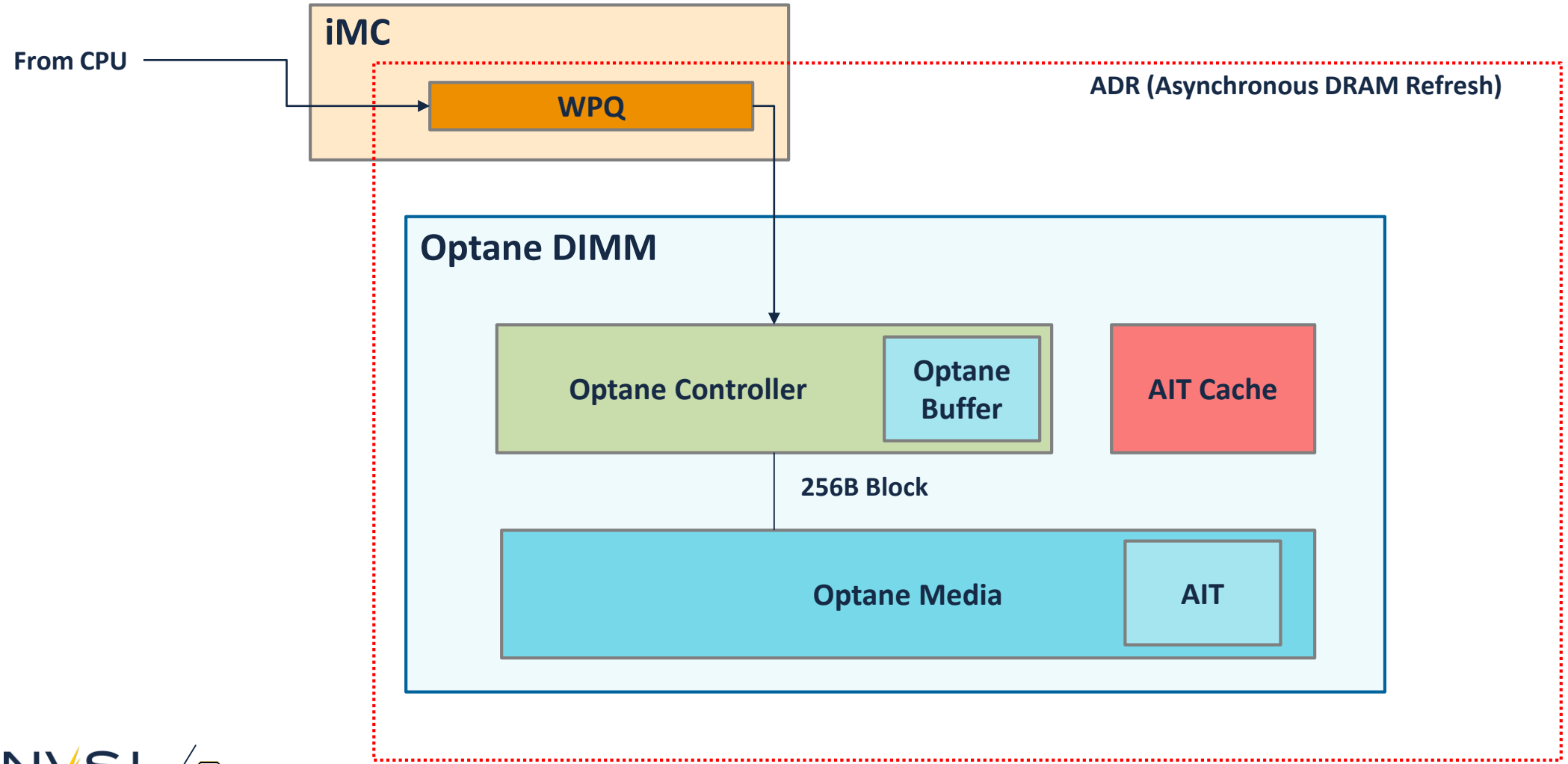


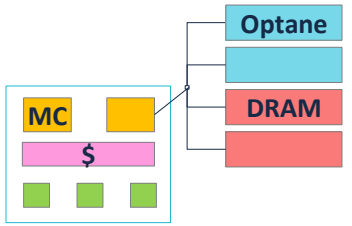
Background: Optane Internals





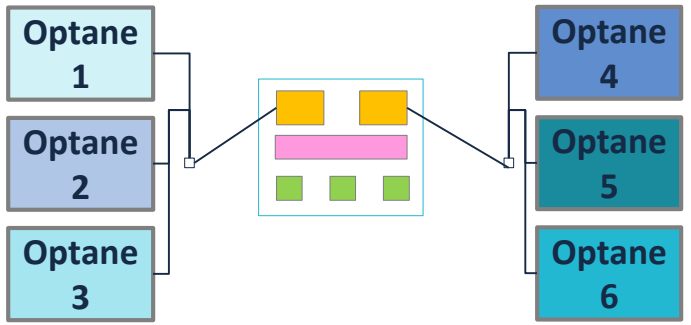
Background: Optane Internals





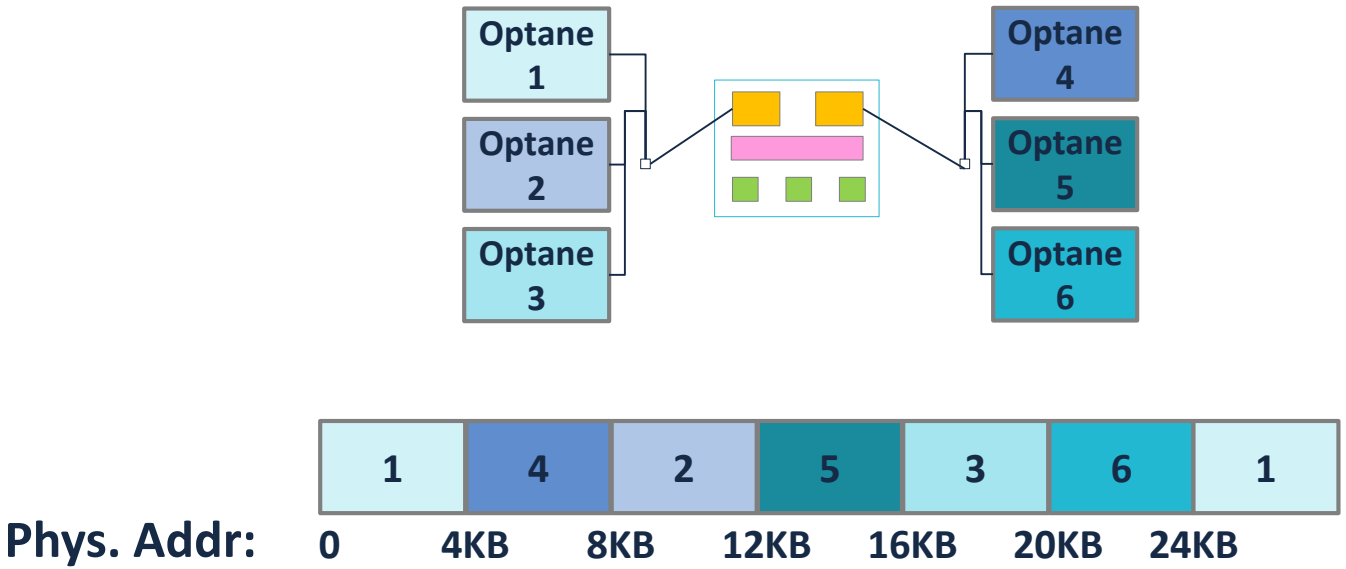
Background: Optane Interleaving

- Optane is interleaved across NVDIMMs at 4KB granularity.



Background: Optane Interleaving

- Optane is interleaved across NVDIMMs at 4KB granularity.



Basics: How does Optane DC perform?

Basics: Our Approach

- Microbenchmark sweeps across state space
  - Access Patterns (random, sequential)
  - Operations (read, ntstore, clflush, etc.)
  - Access/Stride Size
  - Power Budget
  - NUMA Configuration
  - Address Space Interleaving
- Targeted experiments

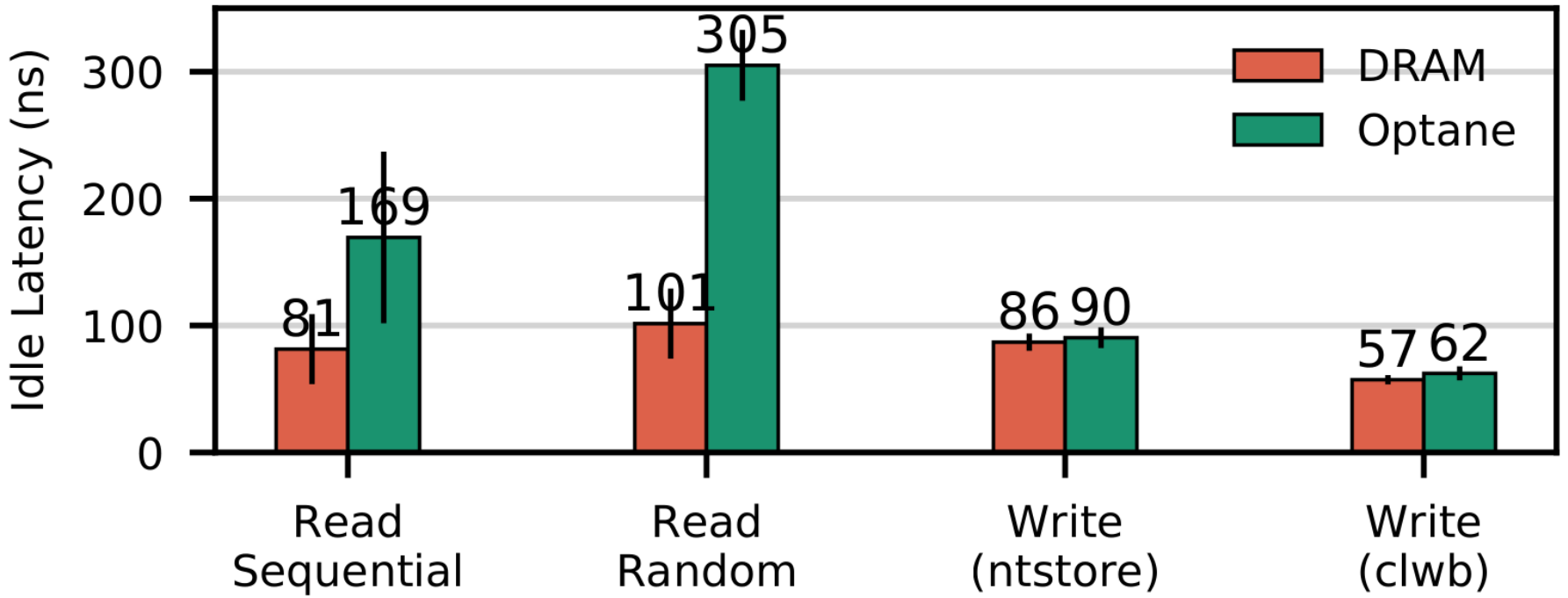
- Total: 10,000 experiments

Test Platform

- CPU
  - Intel Cascade Lake, 24 cores at 2.2 GHz in 2 sockets
  - Hyperthreading off
- DRAM
  - 32 GB Micron DDR4 2666 MHz
  - 384 GB across 2 sockets w/ 6 channels
- Optane
  - 256 GB Intel Optane 2666 MHz QS
  - 3 TB across 2 sockets w/ 6 channels
- OS
  - Fedora 27, 4.13.0

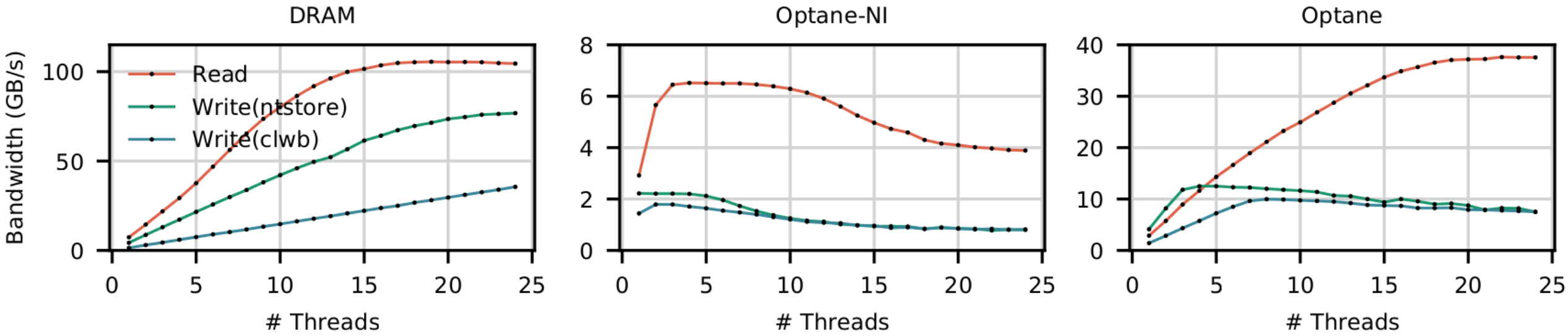
Basics: Latency

- 2x -3x as slow as DRAM
- Write latency masked by ADR



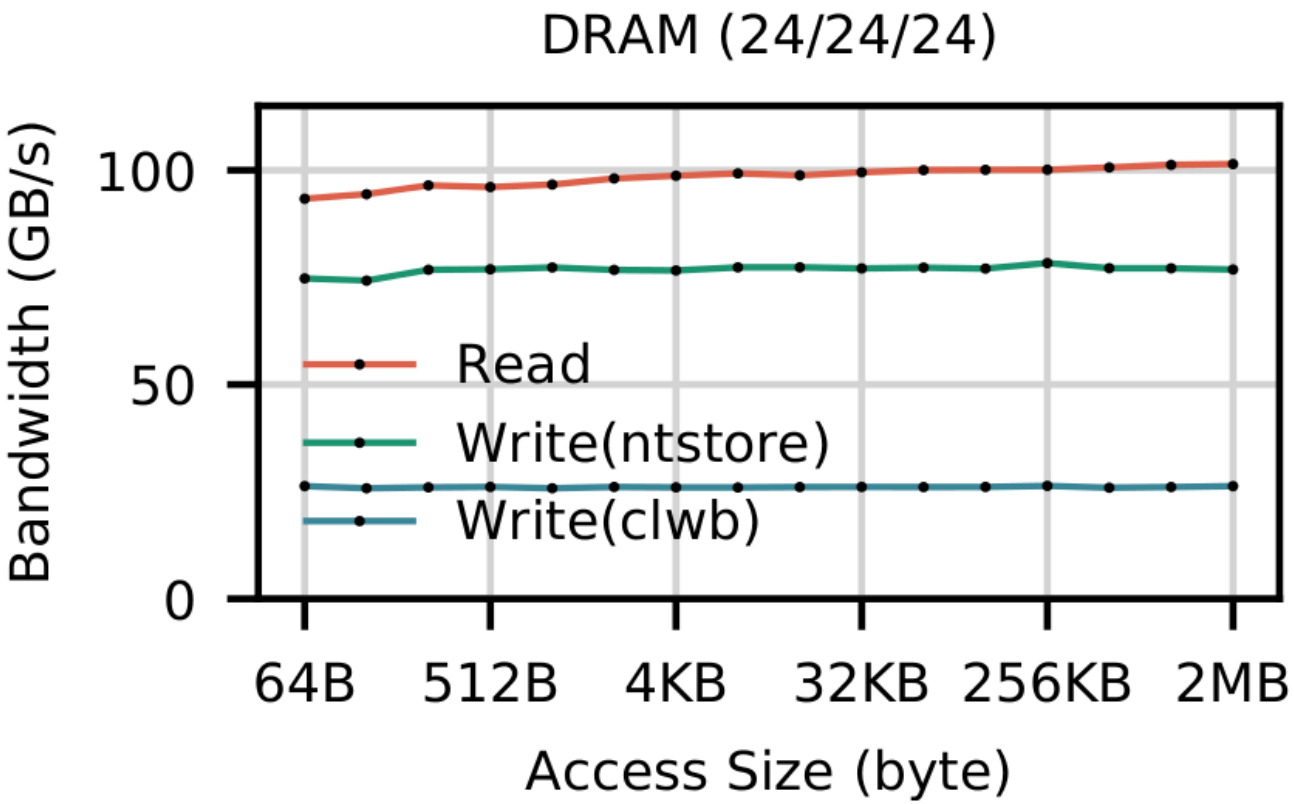
Basics: Bandwidth

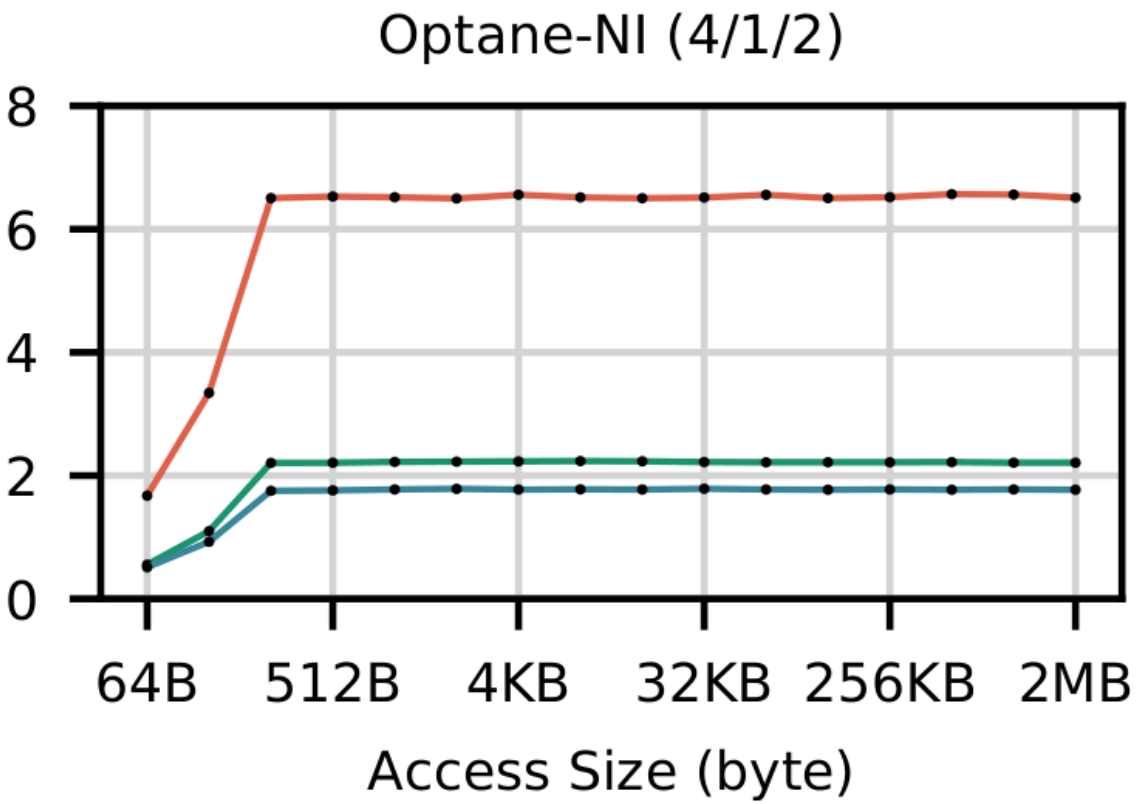
- ~Scalable reads, Non-scalable writes



Basics: Bandwidth

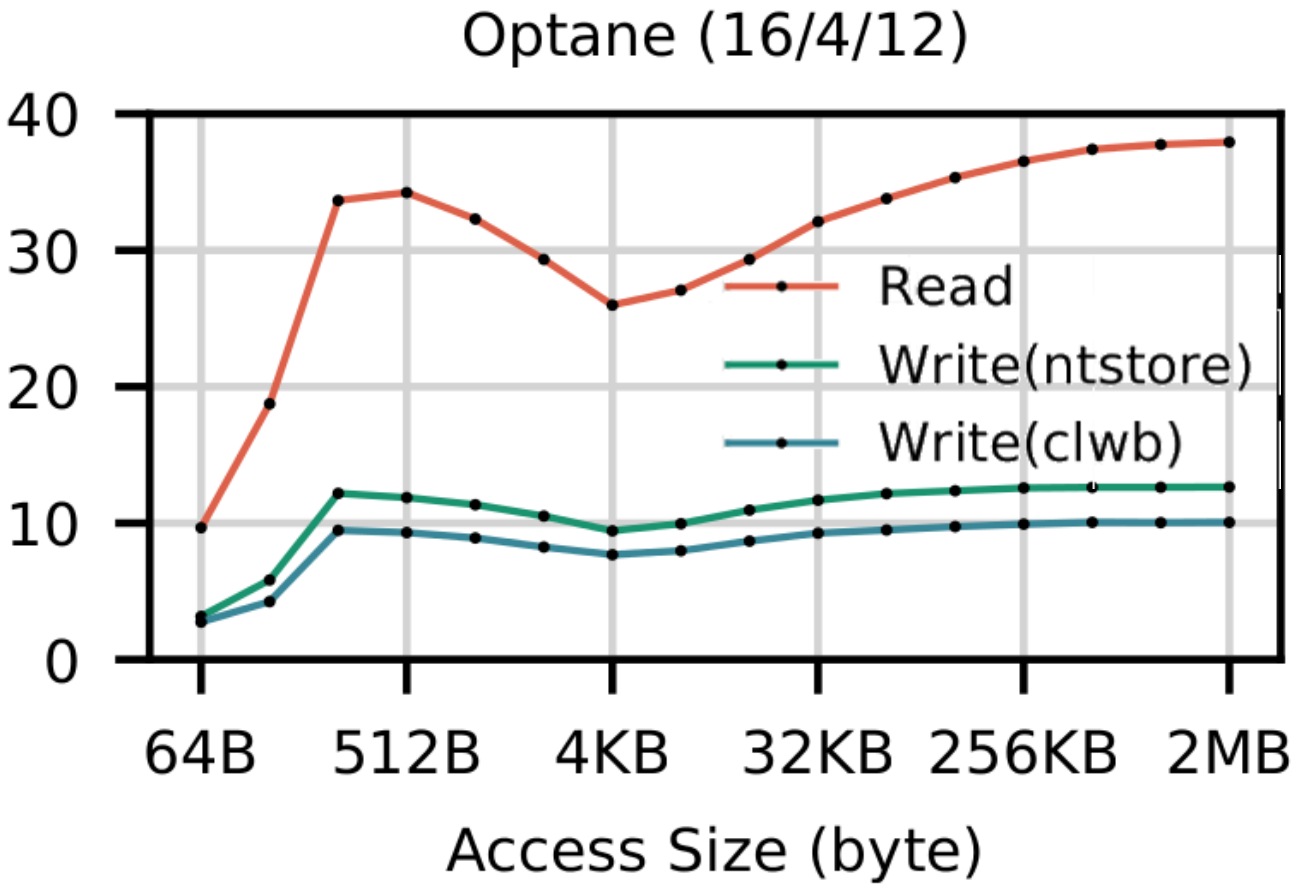
- Access size matters





Basics: Bandwidth

- A mystery!



*answer in ~10 minutes

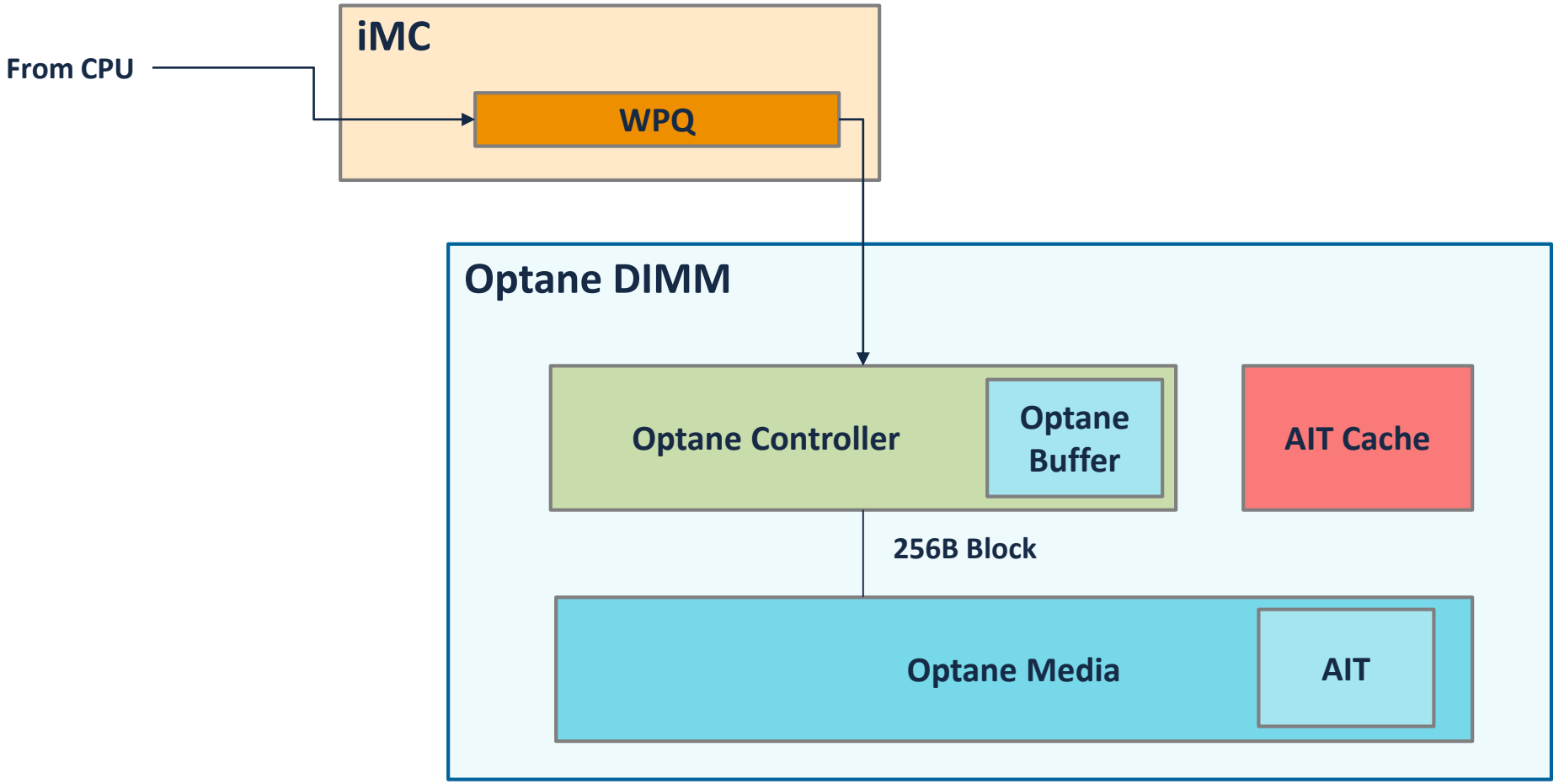
Lessons: What are Optane Best Practices?

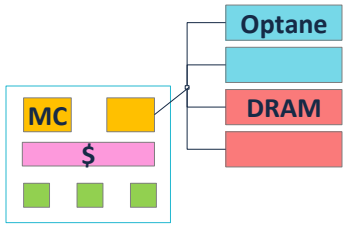
Lessons: What are Optane Best Practices?

- Avoid small random accesses
- Use ntstores for large writes
- Limit threads accessing one NVDIMM

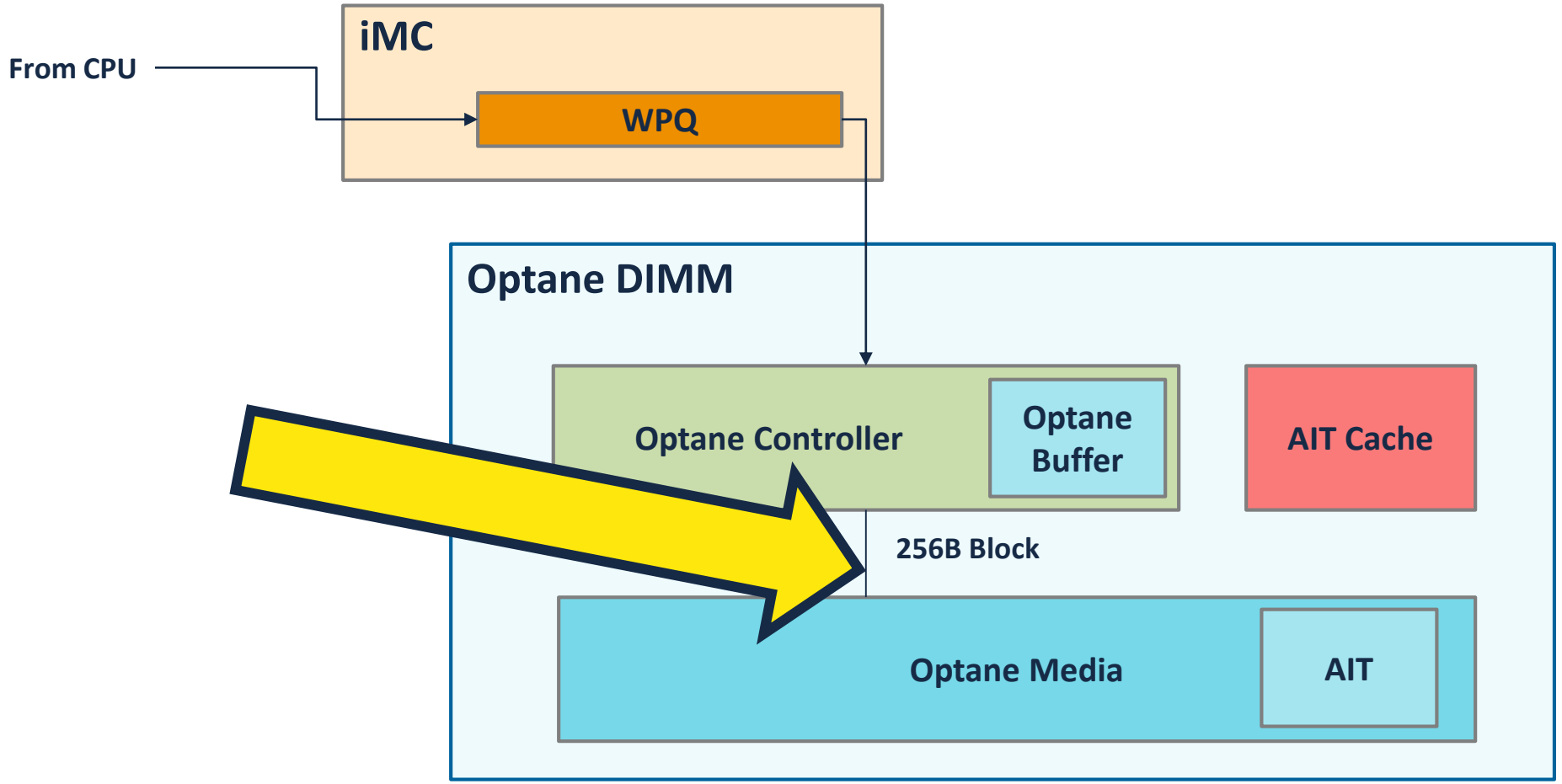
Lesson #1: Avoid small random accesses

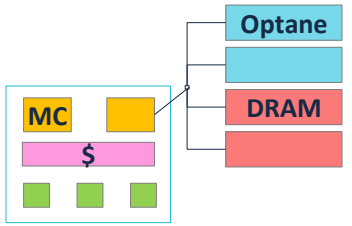
Lesson #1: Avoid small random accesses



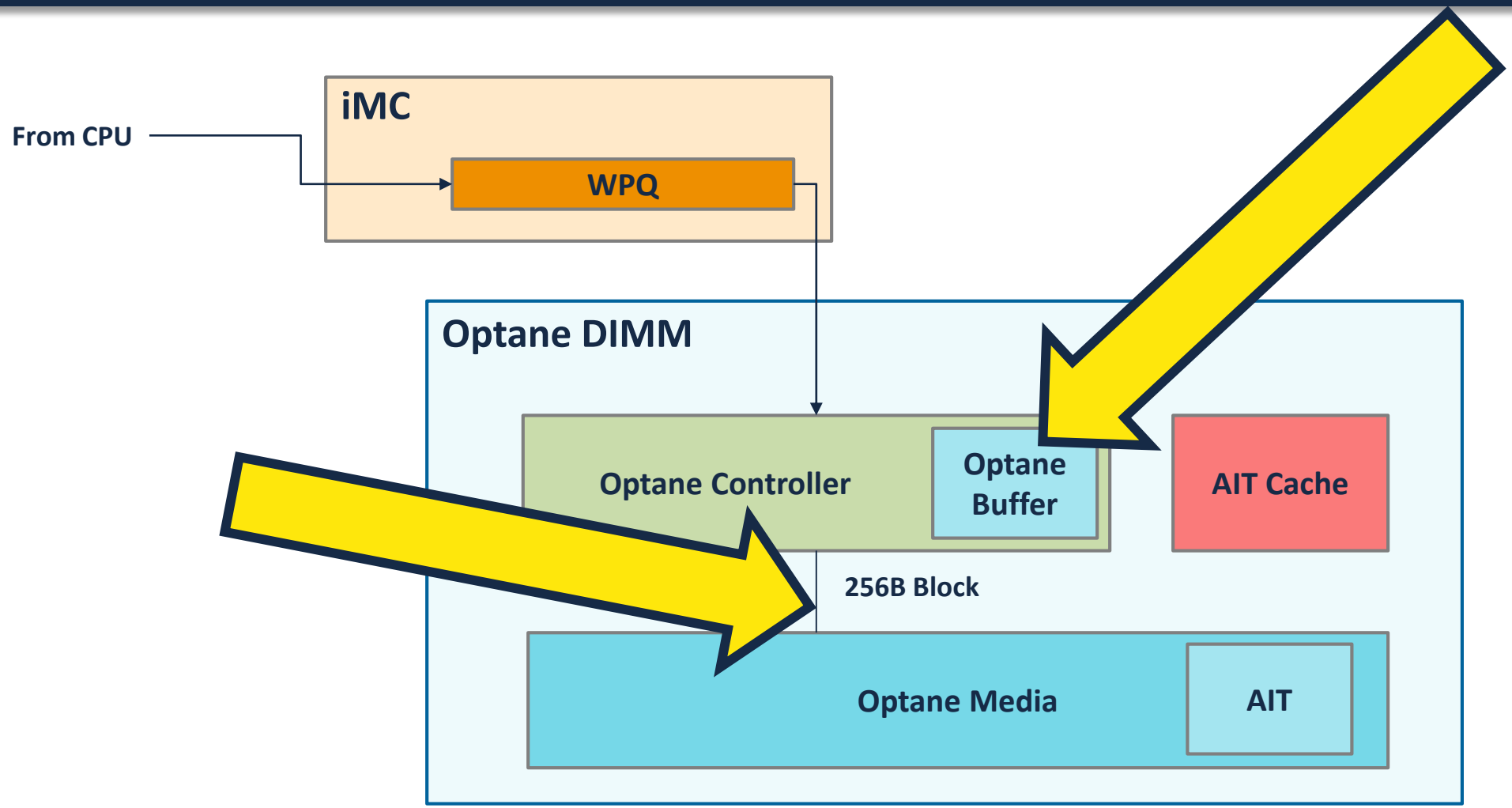


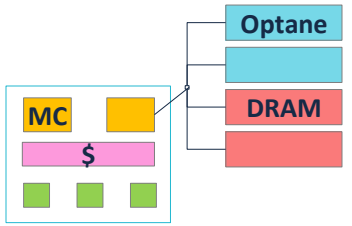
Lesson #1: Avoid small random accesses





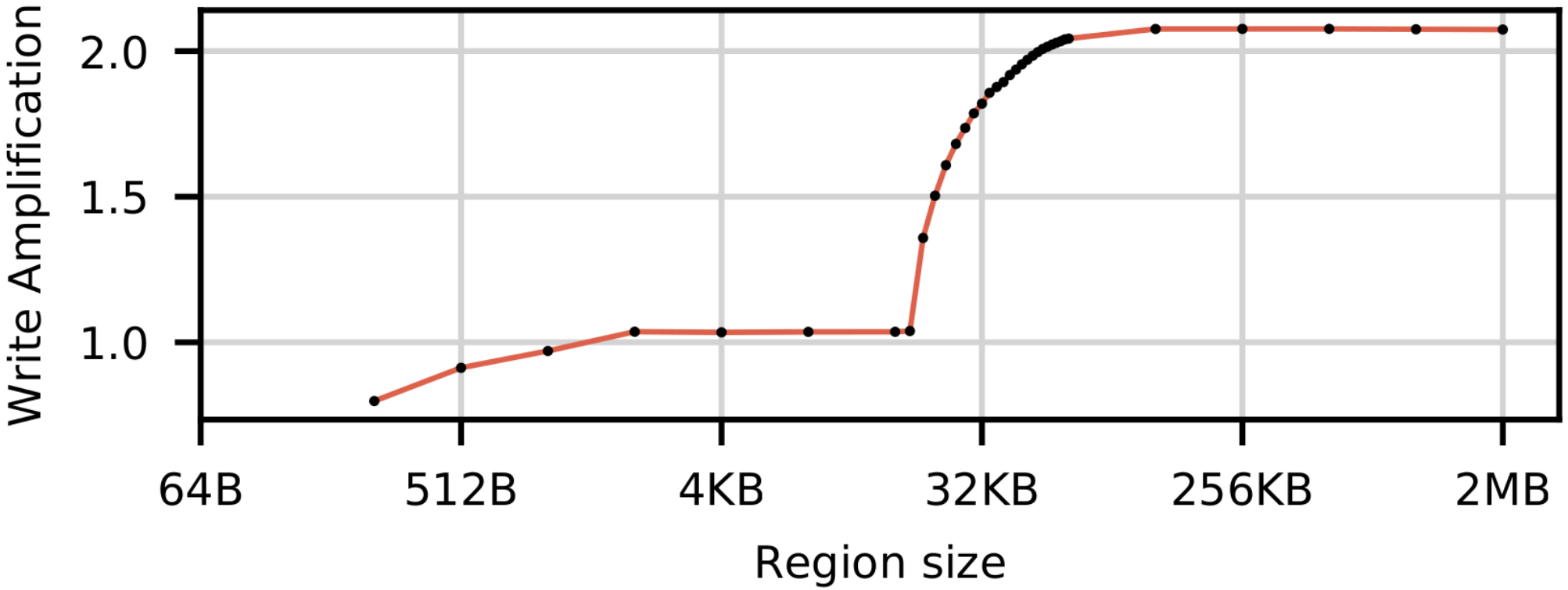
Lesson #1: Avoid small random accesses





Lessons: Optane Buffer Size

- Write amplification if working set is larger than Optane Buffer



Lesson #1: Avoid small random accesses

- Bad bandwidth with:
  - Small random writes (<256B)
  - Not tiny working set / NVDIMM (>16KB)
- Good bandwidth with
  - Sequential accesses

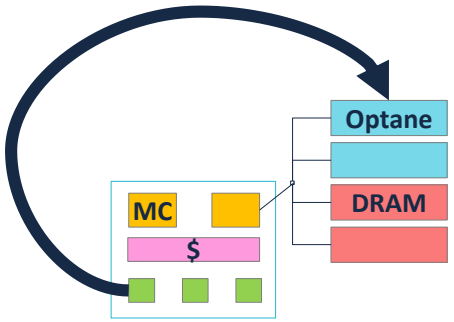
Lesson #2: Use ntstores for large writes

Lesson #2: Use ntstores for large writes

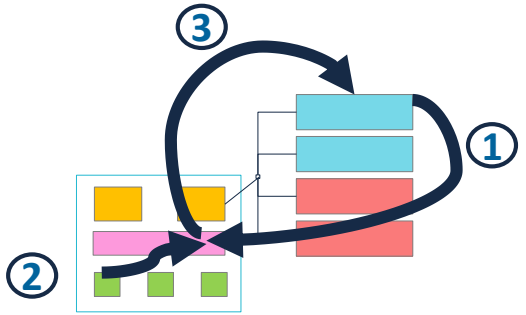
- Non-temporal stores bypass the cache

Lessons: Store instructions

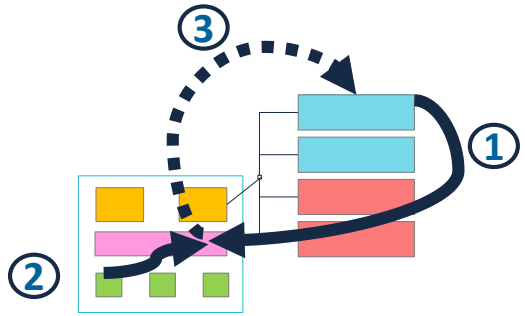
ntstore



store + clwb

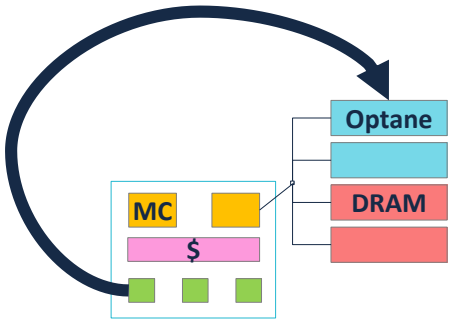


store

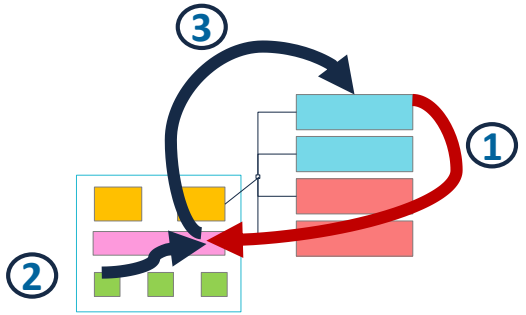


Lessons: Store instructions

ntstore

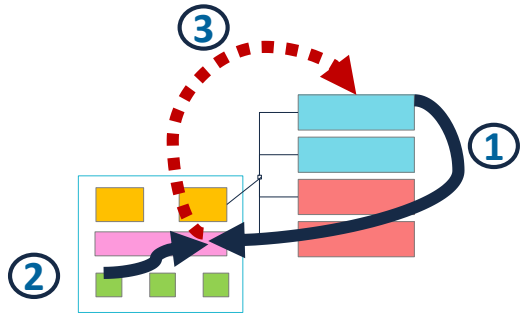


store + clwb



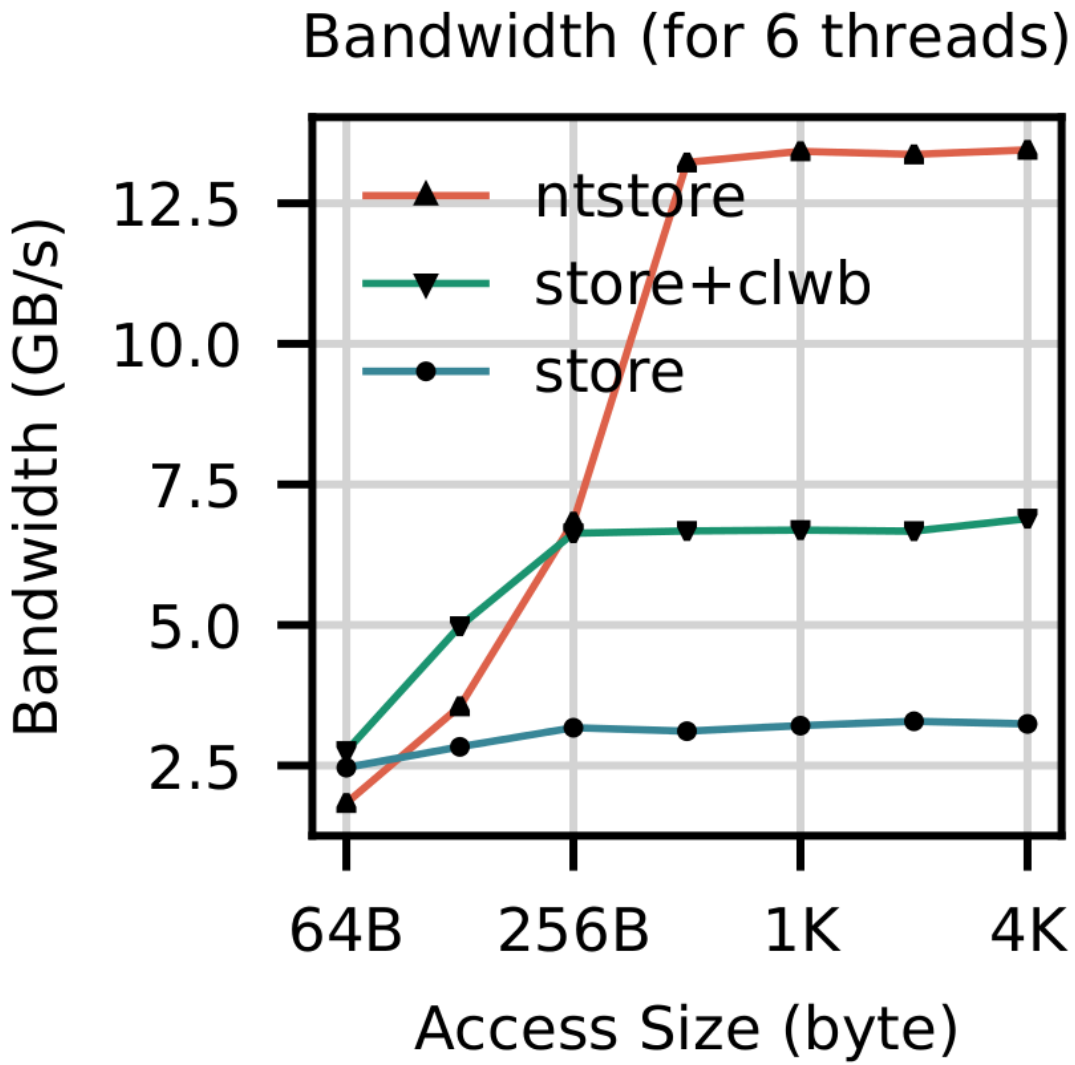
*lost bandwidth

store



*lost locality

Lesson #2: Use ntstores for large writes



Lesson #2: Use ntstores for large writes

- Non-temporal stores bypass the cache
  - Avoid cache-line read
  - Maintain locality

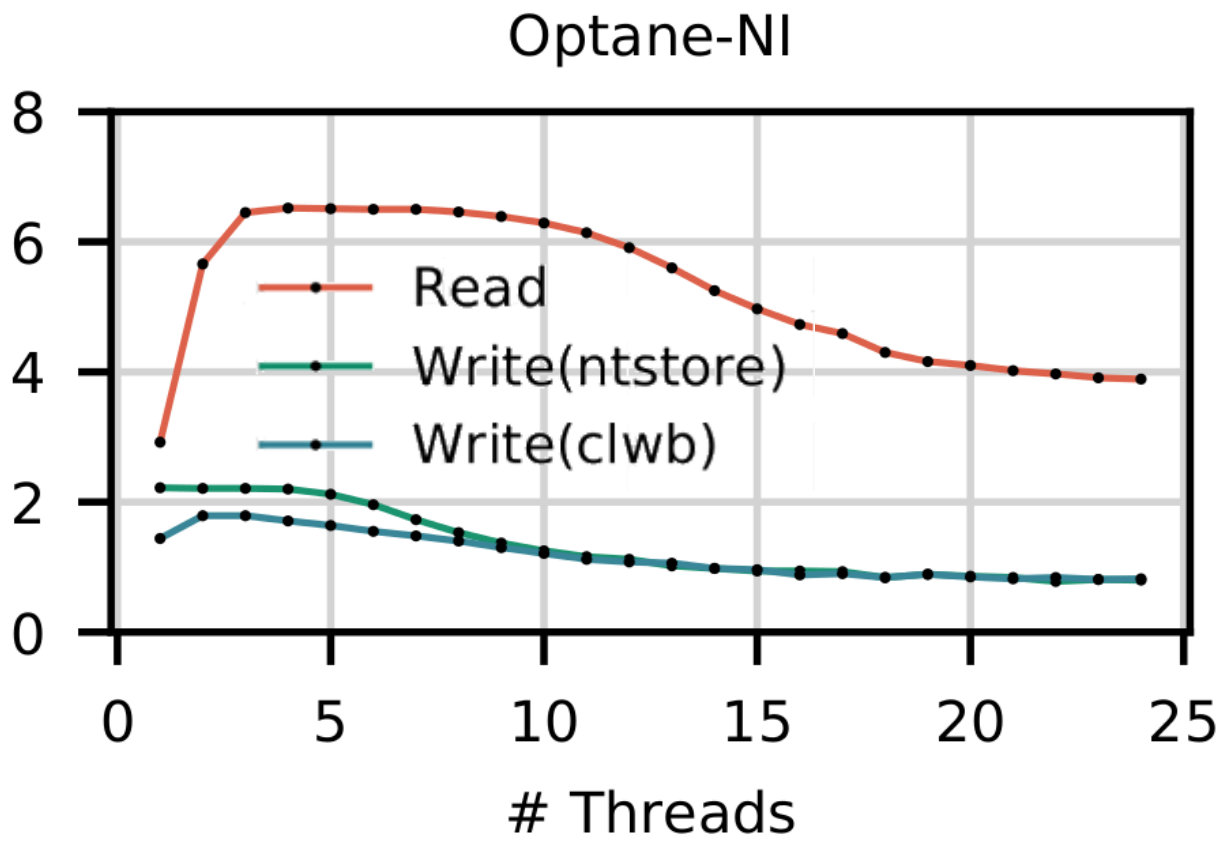
Lesson #3: Limit threads accessing one NVDIMM

Lesson #3: Limit threads accessing one NVDIMM

- Contention at Optane Buffer
- Contention at iMC

Lessons: Contention at Optane Buffer

- More threads = access amplification = lower bandwidth

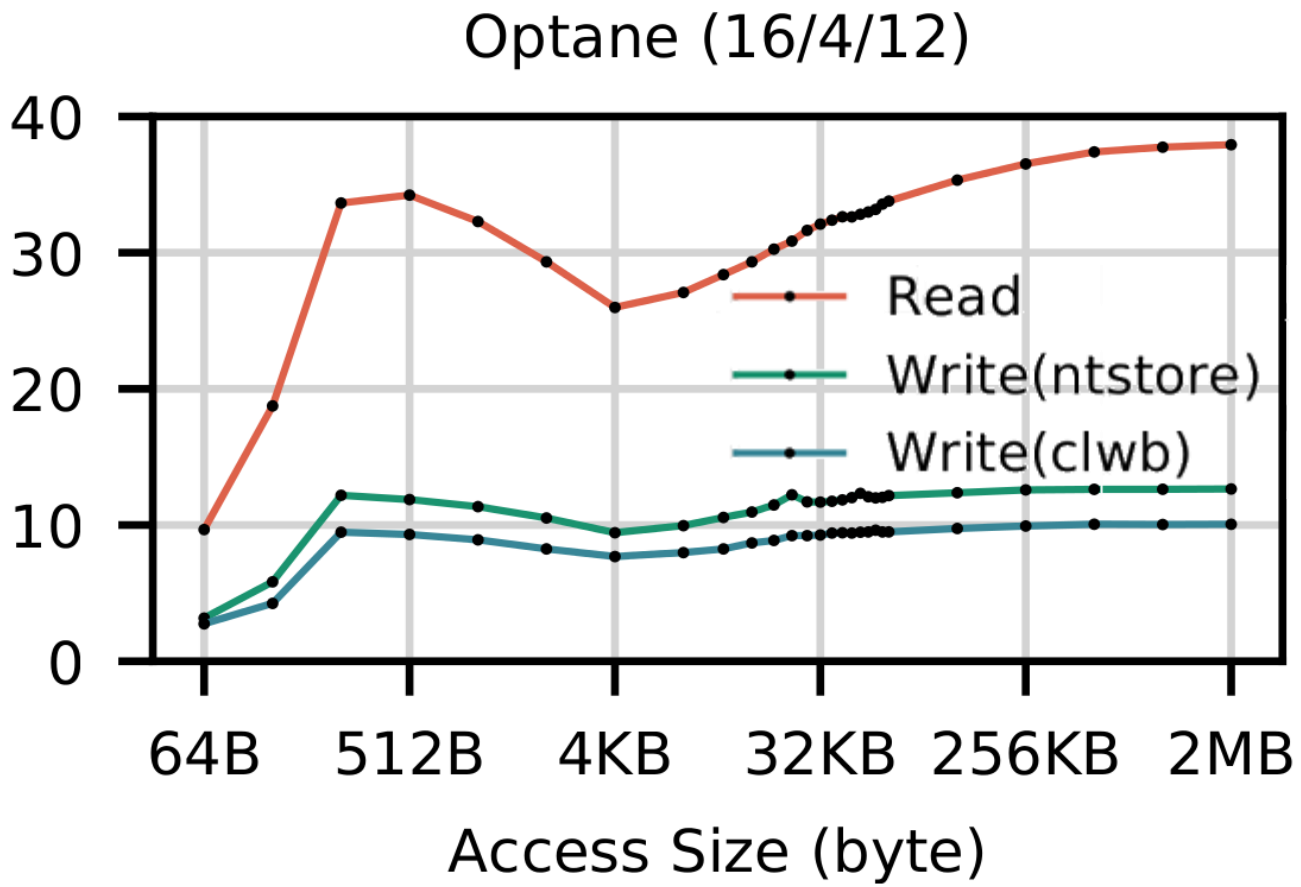


Lessons: Contention at media & iMC

- iMCs queues are small & media is slow
  - Short queues end up clogged

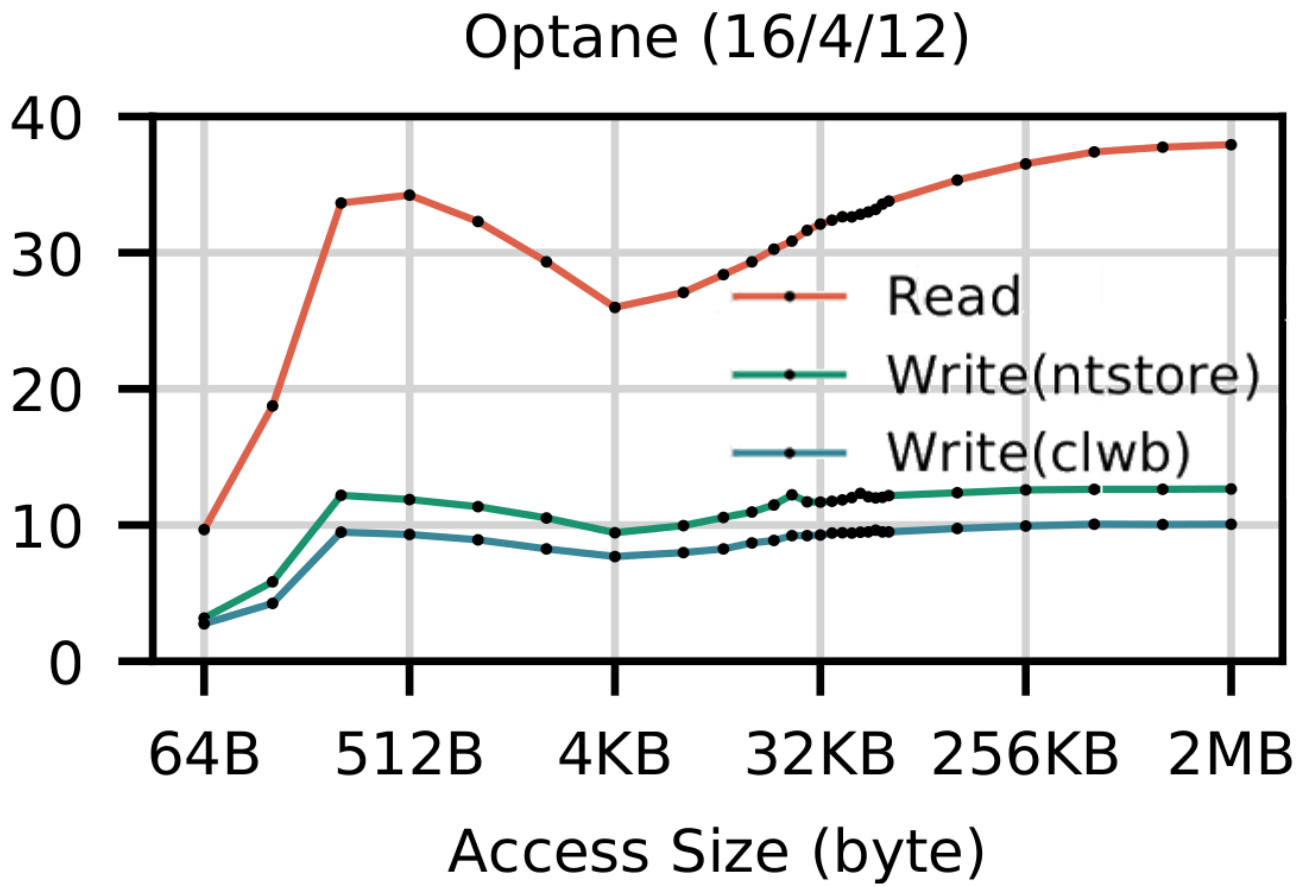
Lessons: Contention at media & iMC

- Contention is largest when random access size = interleave size (4KB)



Lessons: Contention at media & iMC

- Contention is largest when random access size = interleave size (4KB)
- Load is fairest when access size = #DIMMs x interleave size (24KB)



Lesson #3: Limit threads accessing one NVDIMM

- Contention at Optane Buffer
  - Increase access amplification
- Contention at media/iMC
  - Lose bandwidth through uneven NVDIMM load
  - Avoid interleave aligned random accesses

Conclusion

Conclusion

- Not Just Slow Dense DRAM
- Slower media
  -> More complex architecture
    -> Second-order performance anomalies
      -> Fundamentally different

- Max performance is tricky
  - Avoid small random accesses
  - Use ntstores for large writes
  - Limit threads accessing one NVDIMM