Exadata with Persistent Memory: An Epic Journey

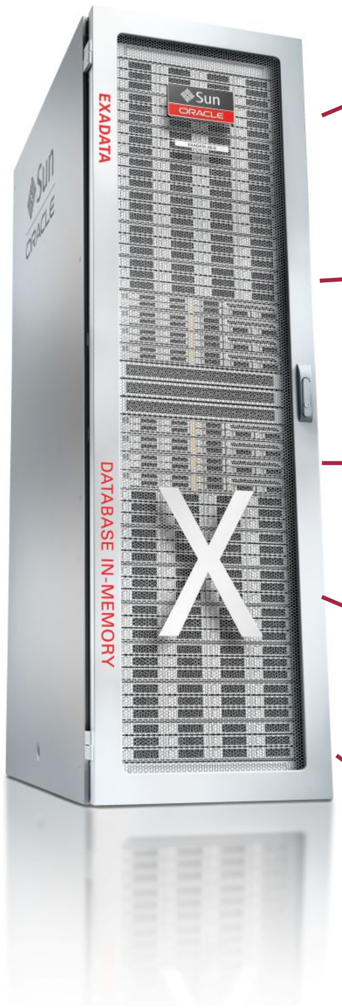
Zuoyu Tao
Oracle

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

What is the Exadata Database Machine?

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

Best Platform for the Oracle Database

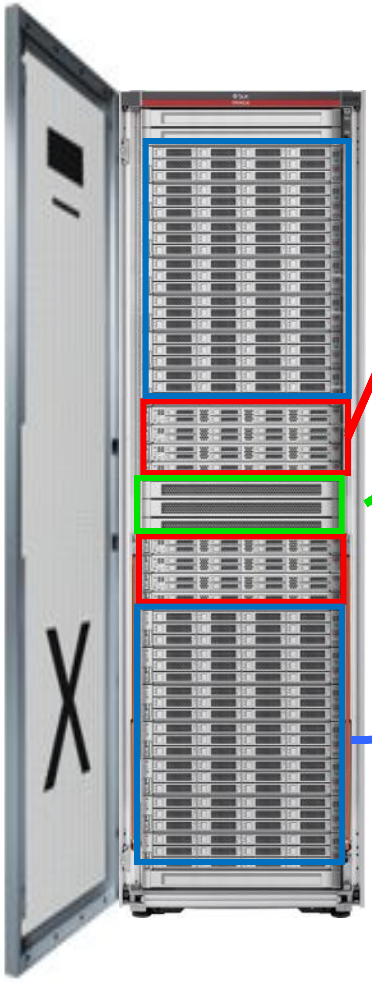


1. State-of-the-art enterprise-grade hardware
2. Optimized for Oracle Database workloads
3. Intelligent storage servers
4. Smart database protocols and optimizations from servers to network to storage
5. One vendor responsible for everything

Exadata Unique Intellectual Property

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

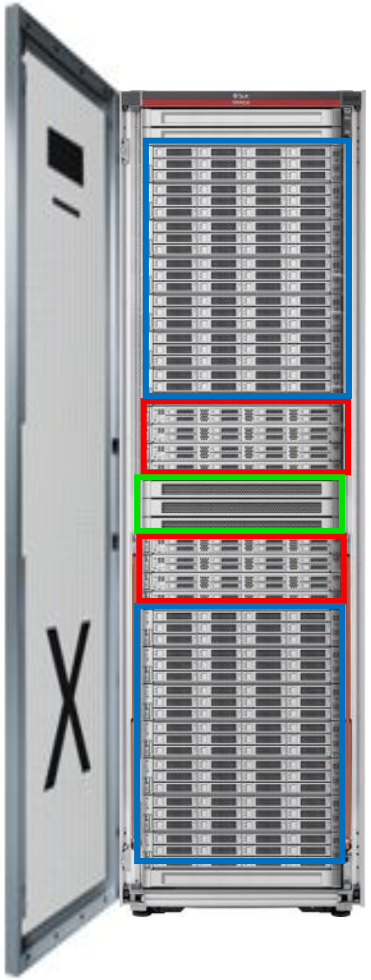
Exadata Components



- Database Servers
- Ultra-Fast Internal Fabric
- Storage Servers

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

Exadata Components



- **Scale-Out 2-Socket or 8-Socket** Database Servers
  - Latest 24 core Intel Xeon
  - Latest 25 GigE client connectivity
- **Ultra-Fast Internal Fabric**
- **Scale-Out <u>Intelligent</u>** Storage Servers
  - Latest Intel CPUs offload database processing
  - Latest disk technology - 14TB Disk Drives (x 12)
  - Latest NVMe PCIe Flash - 6.4 TB Hot Swappable (x 4)
  - Intel Optane DC Persistent Memory

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

Exadata Uniquely Accelerates OLTP

Completely Automatic, No Management Required

- **Exadata uniquely deals with traditional OLTP (online transaction processing) bottleneck: Random I/O**
- Unique scale-out storage, ultra-fast NVMe Flash, ultra-fast iDB delivers:
  - Over 3.5 Million Database 8K reads or **writes** per rack;
  - 250us I/O latency at 2M IOPs

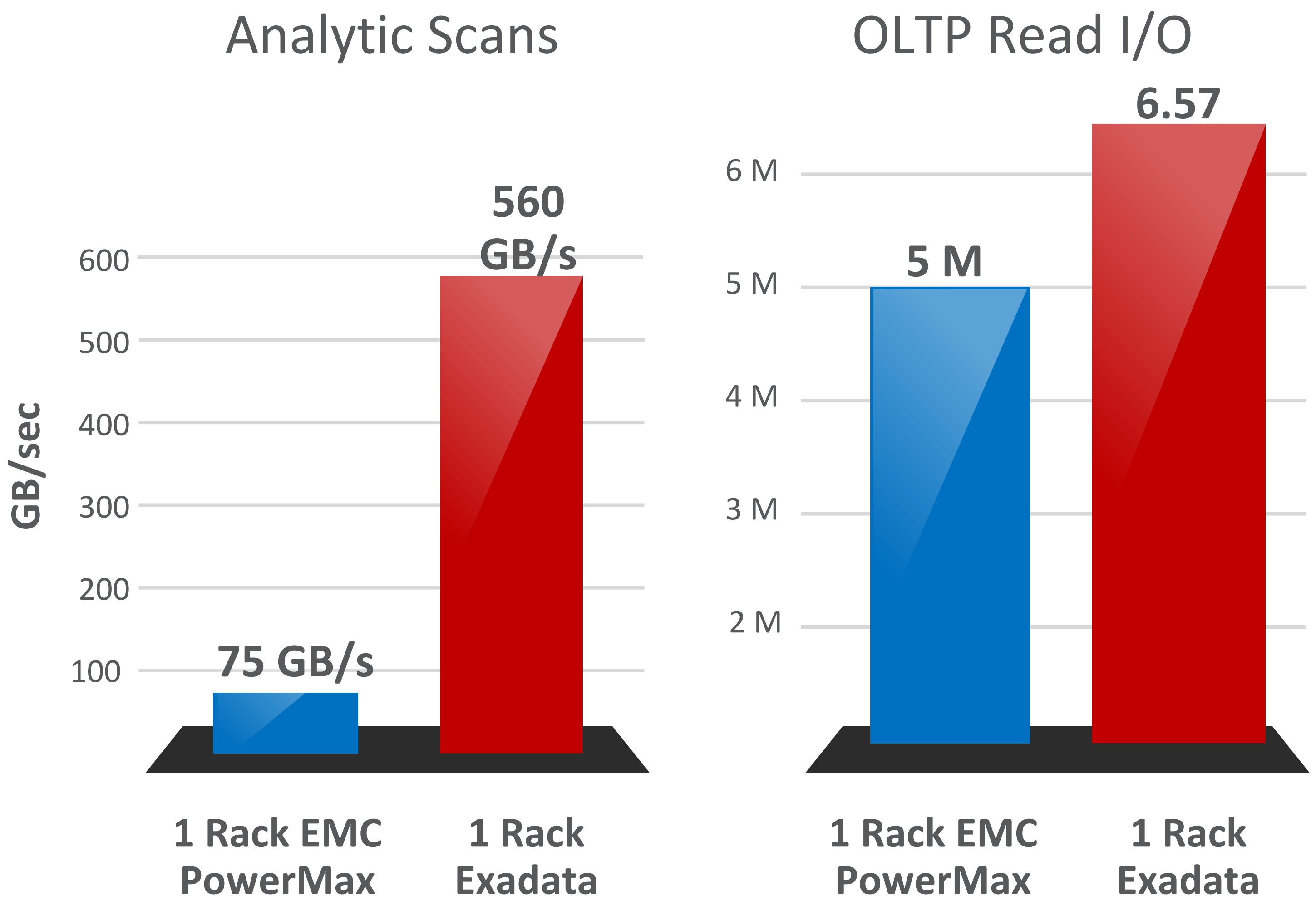
Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

Exadata X8 I/O is Dramatically Faster than All-Flash EMC

Single rack Exadata beats the fastest EMC PowerMax **all-Flash** array in performance metrics:

- **7.4X more throughput**
- **2X faster OLTP I/O latency**
- **1.5 Million more IOPS**

Exadata performance scales linearly as more racks are added



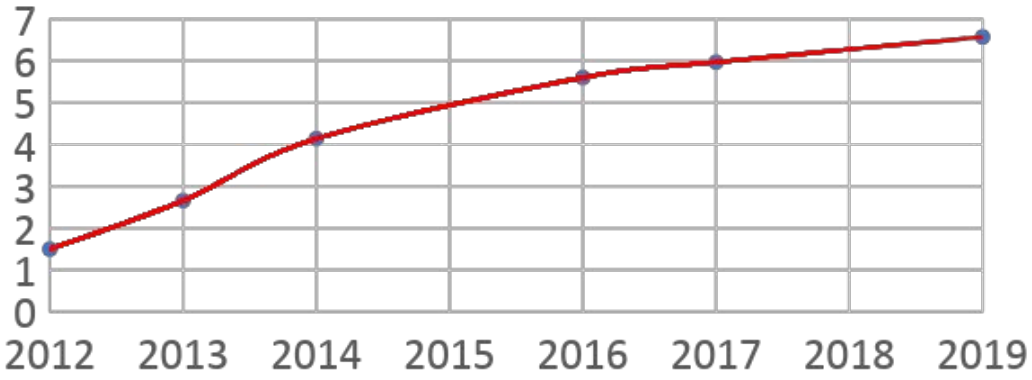
Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

Database Platform Leadership Since 2008

	V1	V2	X2	X3	X4	X5	X6	X7	X8
	Sep 2008	Sep 2009	Sep 2010	Sep 2012	Nov 2013	Dec 2014	Apr 2016	Oct 2017	Apr 2019
Storage (TB)	168	336	504	504	672	1344	1344	1.68	2.35 PB
Flash Cache (TB)	0	5.3	5.3	22.4	44.8	89.6	179.2	358	358 TB
CPU (cores)	64	64	96	128	192	288	352	384	384 cores
Max Mem (GB)	256	576	1152	2048	4096	6144	12288	12288	12 TB
Ethernet (Gb/s)	8	24	184	400	400	400	400	800	800 Gb/s
Scan Rate (GB/s)	14	50	75	100	100	263	301	350	560 GB/s
Read IOPS (M)	.05	1	1.5	1.5	2.66	4.14	5.6	5.97	6.57 M

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

Improving Performance



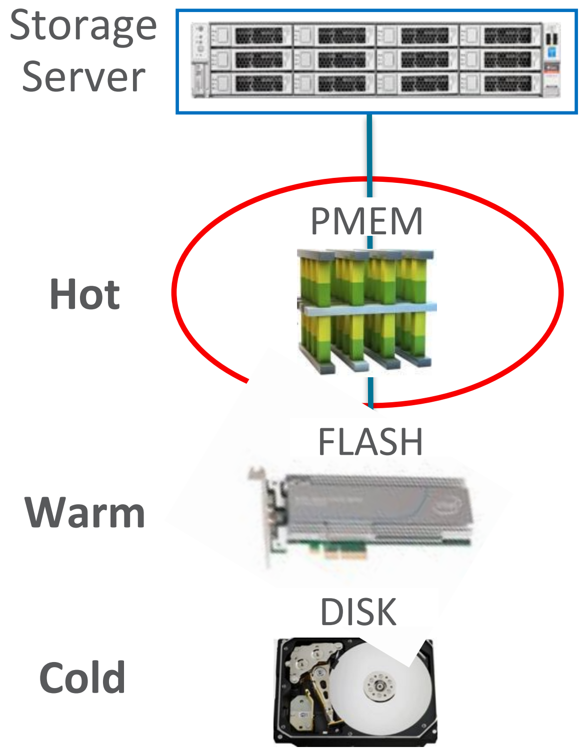
- Flash performance (Read IOPS) plateauing
- Total bandwidth increasingly bottlenecked by network limitations
- Latency can be improved
- New disruptive technology: Persistent Memory

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

PMEM on Exadata

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

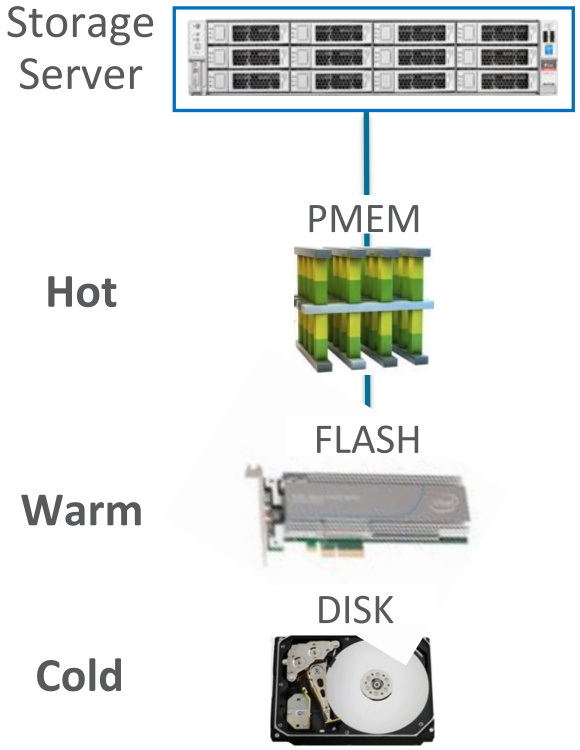
Use **Persistent Memory** (PMEM) to make IOs faster



- Added in front of flash memory
  - Cache hierarchy with hottest data cached in PMEM
- **Inclusive**, shared cache
- Mirrored
- Adapted existing Flash Cache Code for PMEM Cache

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

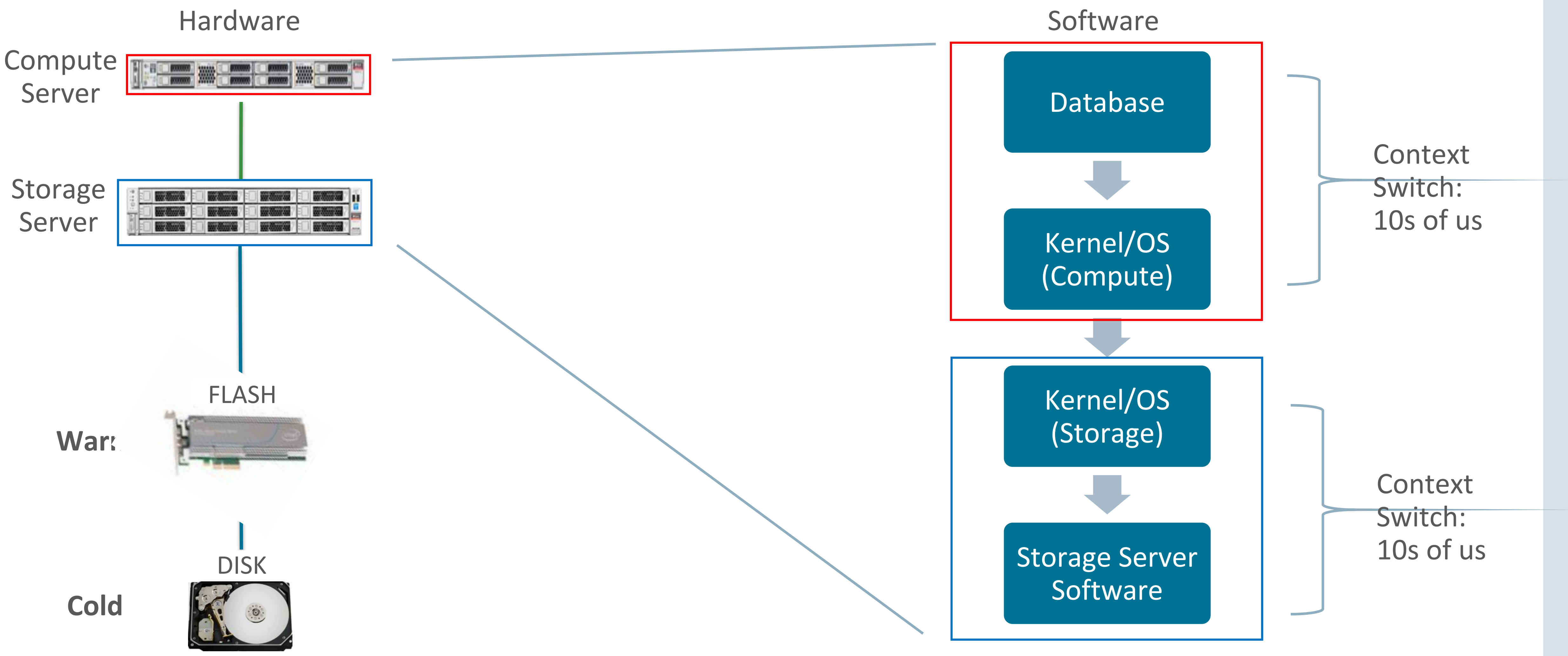
Use **Persistent Memory** to make IOs faster



- Can be Writethrough
  - PMEM used as larger volatile RAM
- Can be Writeback
  - Challenges and opportunities of persistence
- Improves throughput, latency
- But can achieve still lower latencies…

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

Current I/O Latencies in Exadata

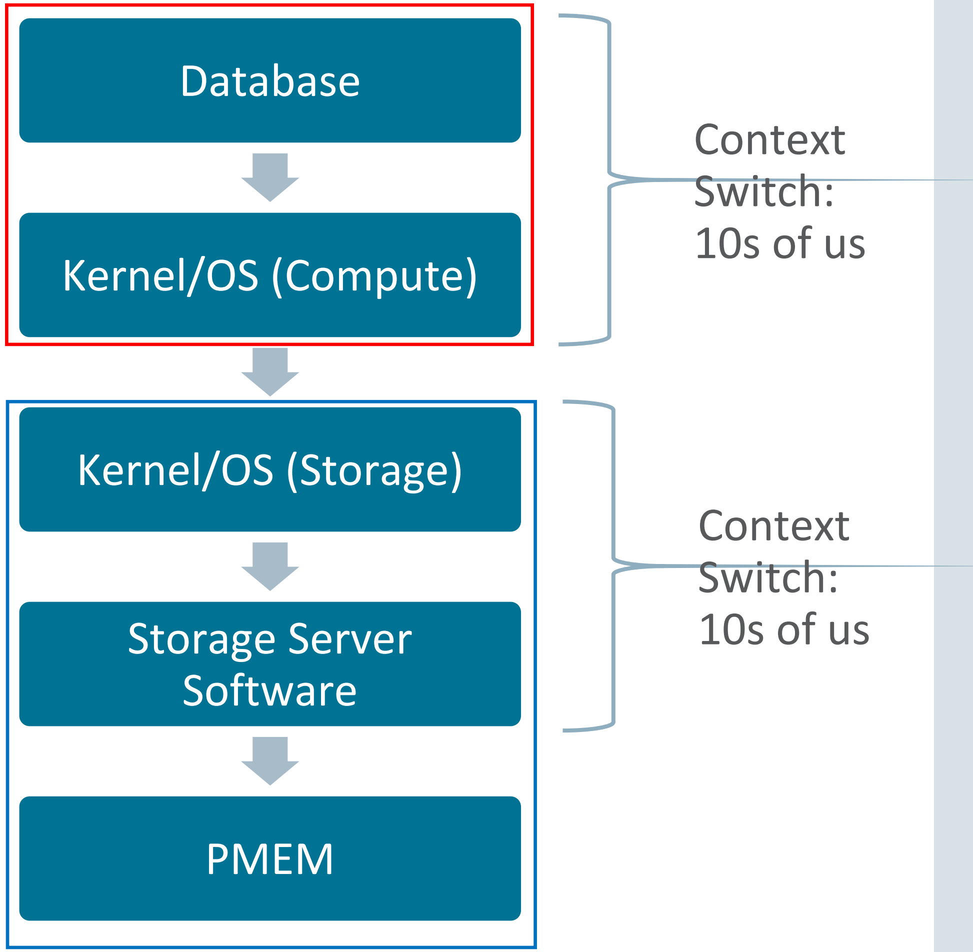


Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

PMEM Latency

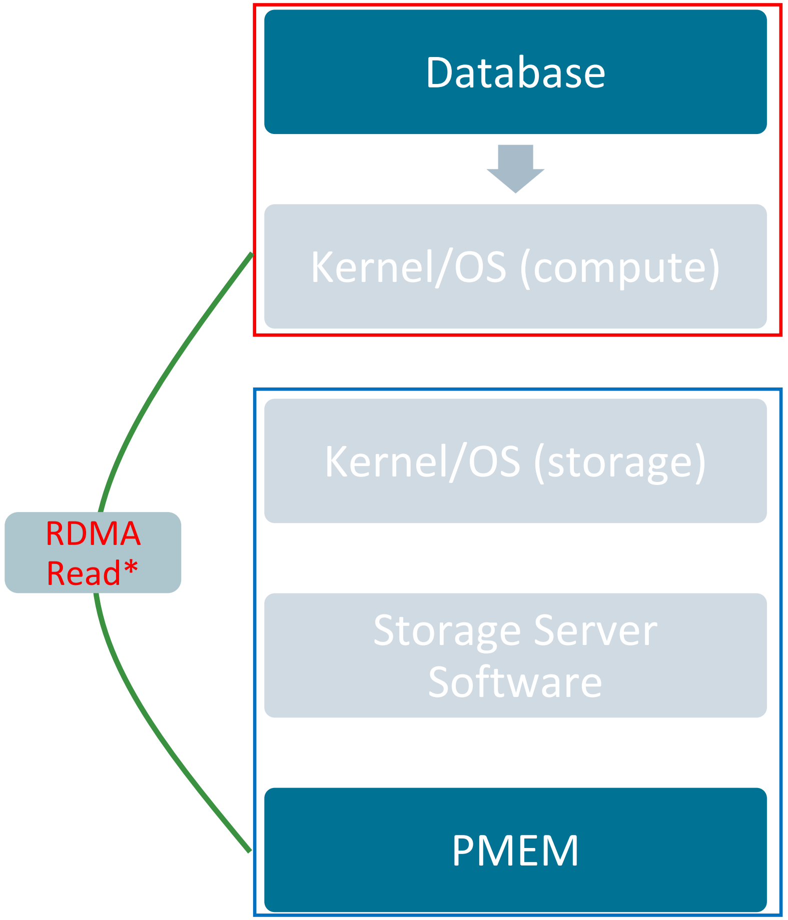
- PMEM IO is Super Fast
  - 8K local memory read @ <1 usec
- IO Software Stack Overhead
  - Context Switch Cost of Storage Software: **10s of us**
  - Context Switch Cost of Database: Additional **10s of us**

Over 90% of the Time *Wasted*. Yikes!!!



Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

RDMA makes access to persistent memory faster



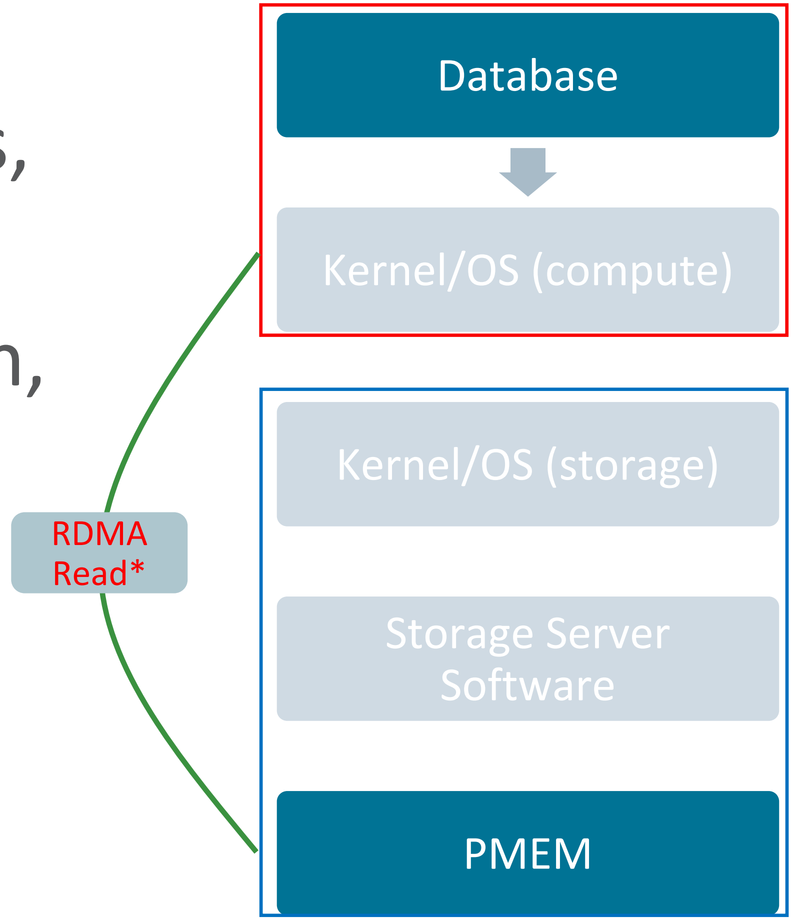
- Bypass the software stack
- 10X+ faster access latency

* Preview

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

Ultra Fast Path: One-sided Read

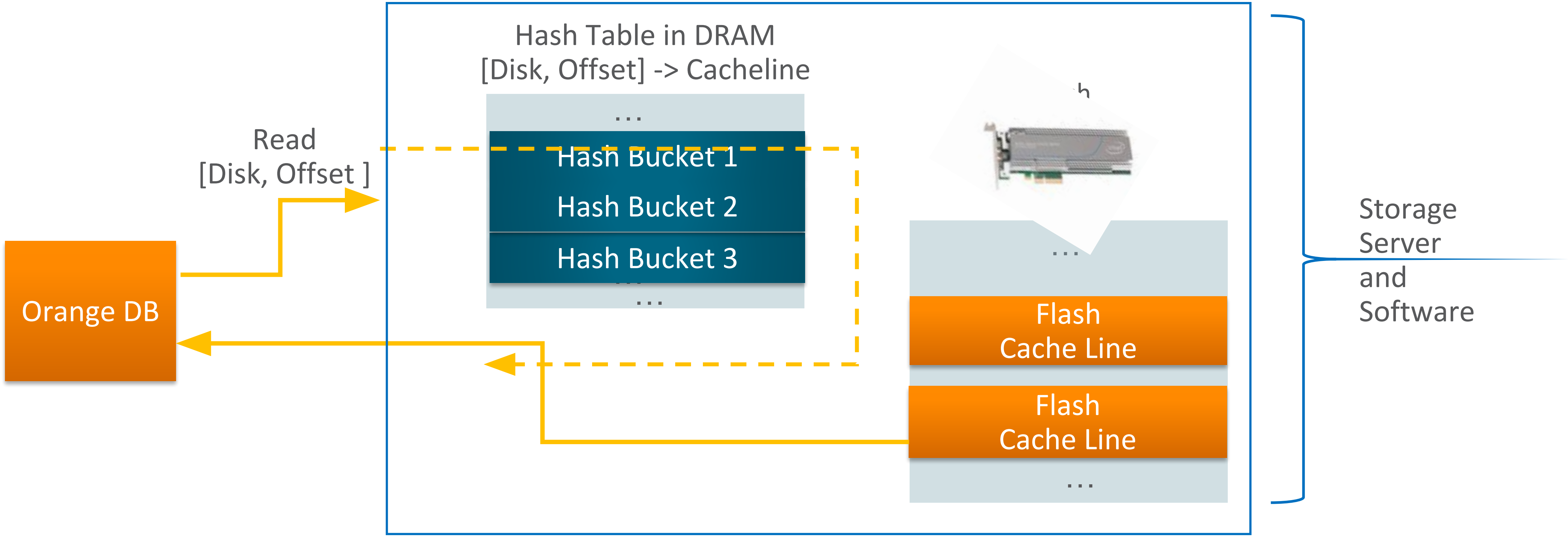
- Direct RDMA to PMEM on storage servers, Eliminating Software Cost
- Database Sync Polls for RDMA Completion, Eliminating Database Context Switch
  - Faster (i.e., lower latency) and Cheaper (i.e., using less CPU)
- For the future: one-sided <u>writes</u>



Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

One sided reads – Where is My Data on PMEM?

- Traditionally storage software looks up hash table to redirect hits to FC



Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

One sided reads – Where is My Data on PMEM?

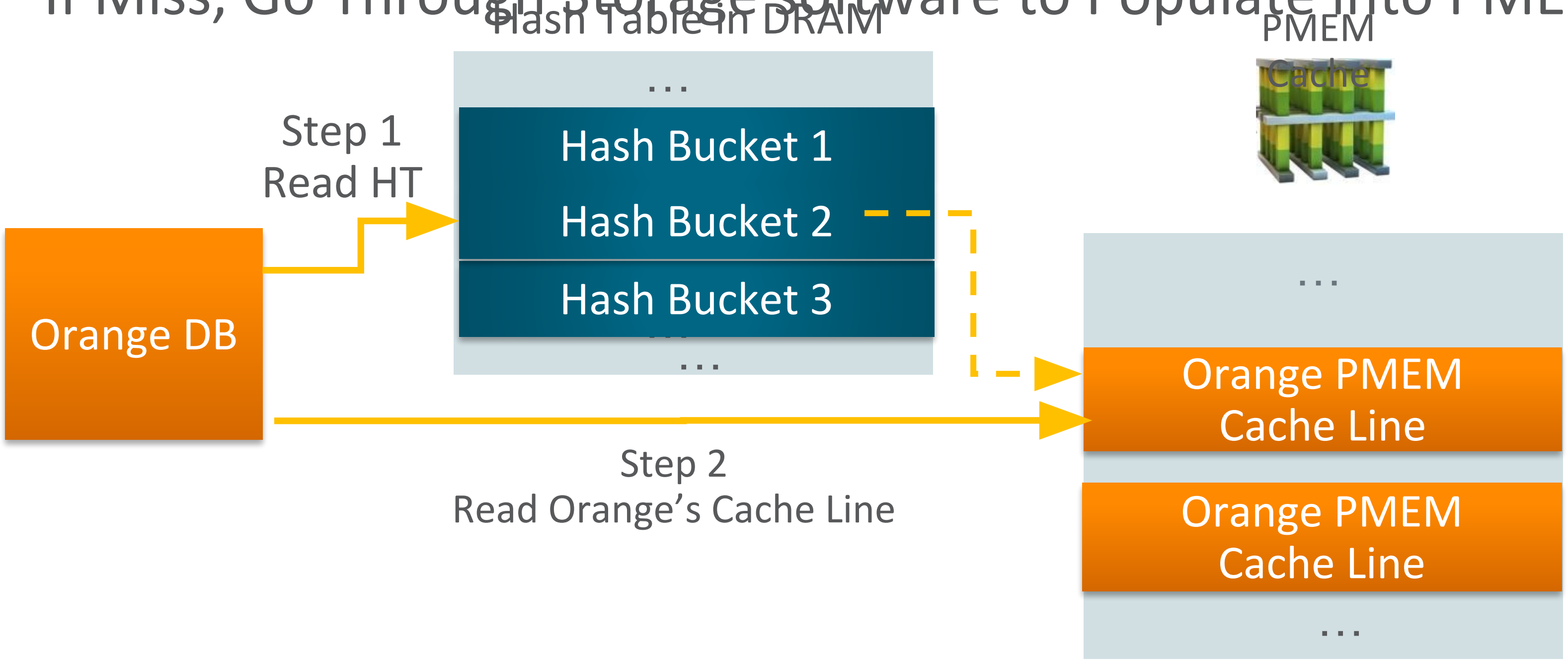
- Solution: DB Probes Hash Table via RDMA Read
  - Find bucket based on hash of desired [disk, offset] - do RDMA read of that bucket
- No DB Side Translation Caching or Invalidation
  - Scales better

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

PMEM Cache – DB Read Flow

1. RDMA Read of DRAM Hash Table (HT) to Look up Translation (~hundred bytes)
2. If Hit, RDMA Read of PMEM Cache Line (few kilobytes)

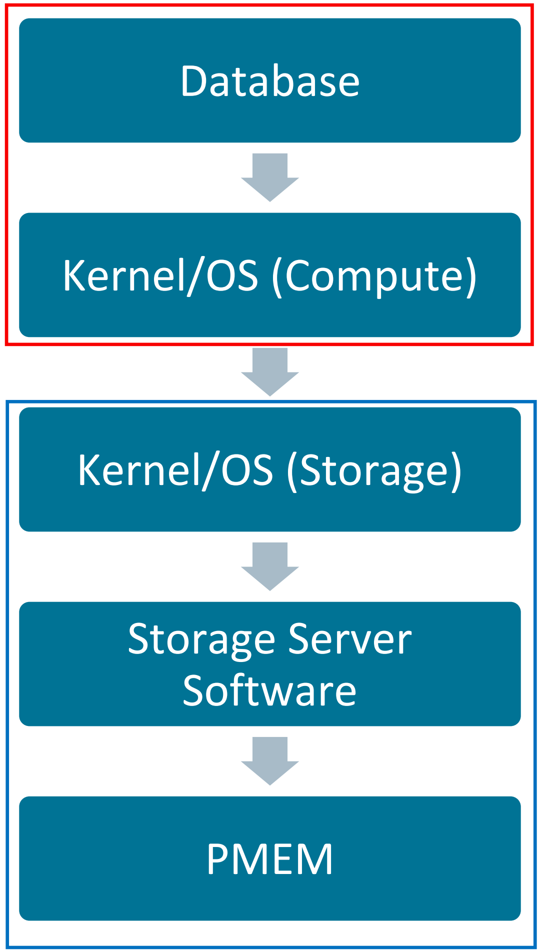
   If Miss, Go Through Storage software to Populate into PMEM Cache



Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

Two-sided Read/Write Path

- Go through storage server software
- Why?
  - Cache population
  - Prevents torn blocks



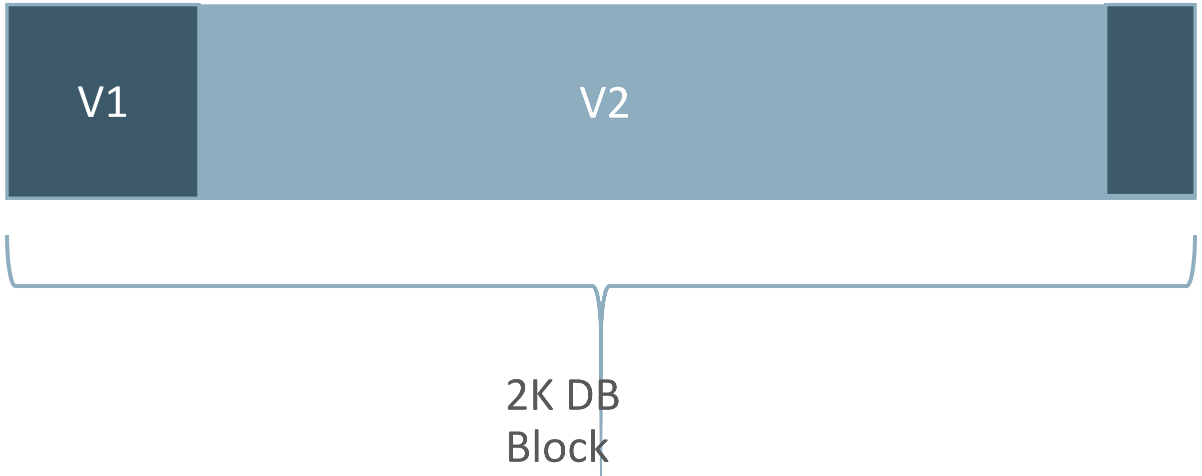
Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

Torn Block Prevention on PMEM

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

Problem: torn blocks

- PMEM has 8-byte Atomic Write Guarantee
  - Not Good Enough for Database Blocks (>= 2K)
  - Database checks beginning and end of block -- does not catch torn block
  - Torn blocks = Database corruption!



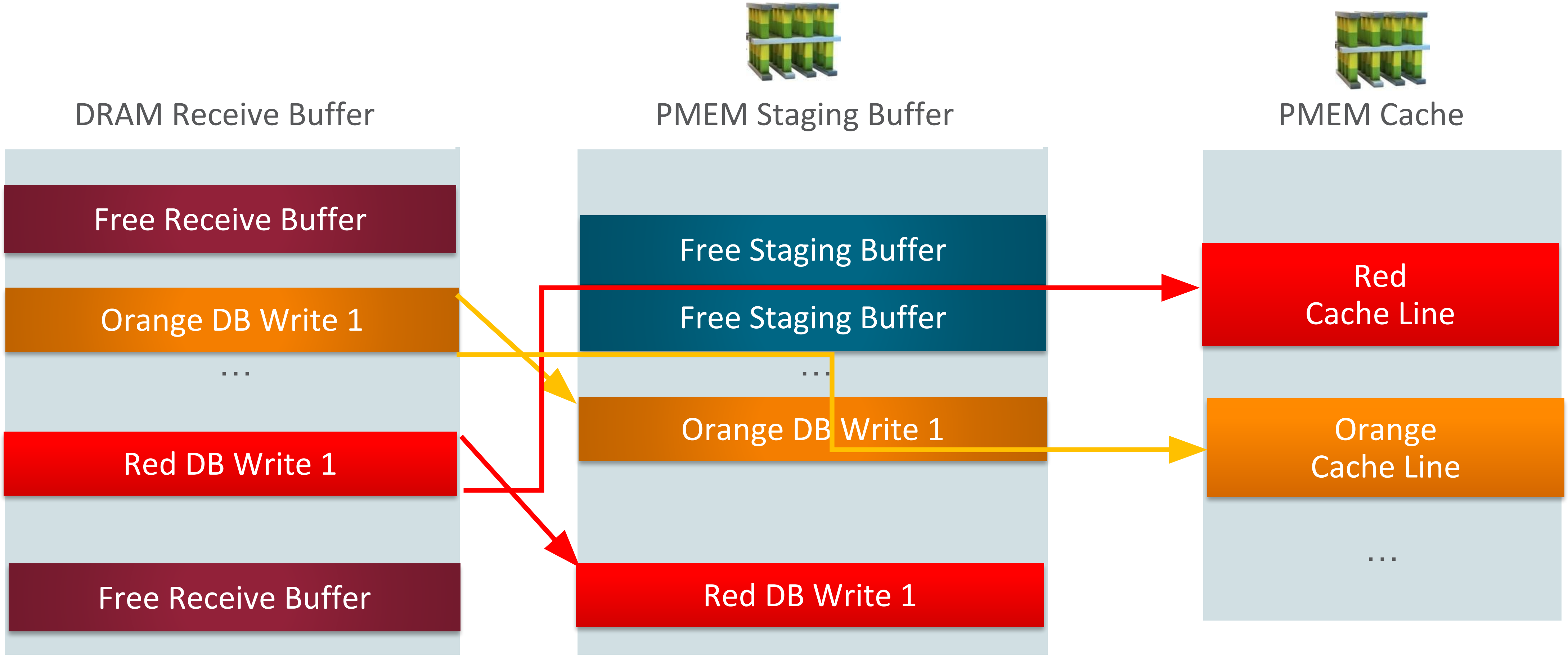
Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

Solution: Send Writes to Storage Server Software

- Database does 2-sided Writes
  - No RDMA
  - Not in the Critical Path of Application Performance
- Storage server software guarantees Atomic Database Block Writes to PMEM

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

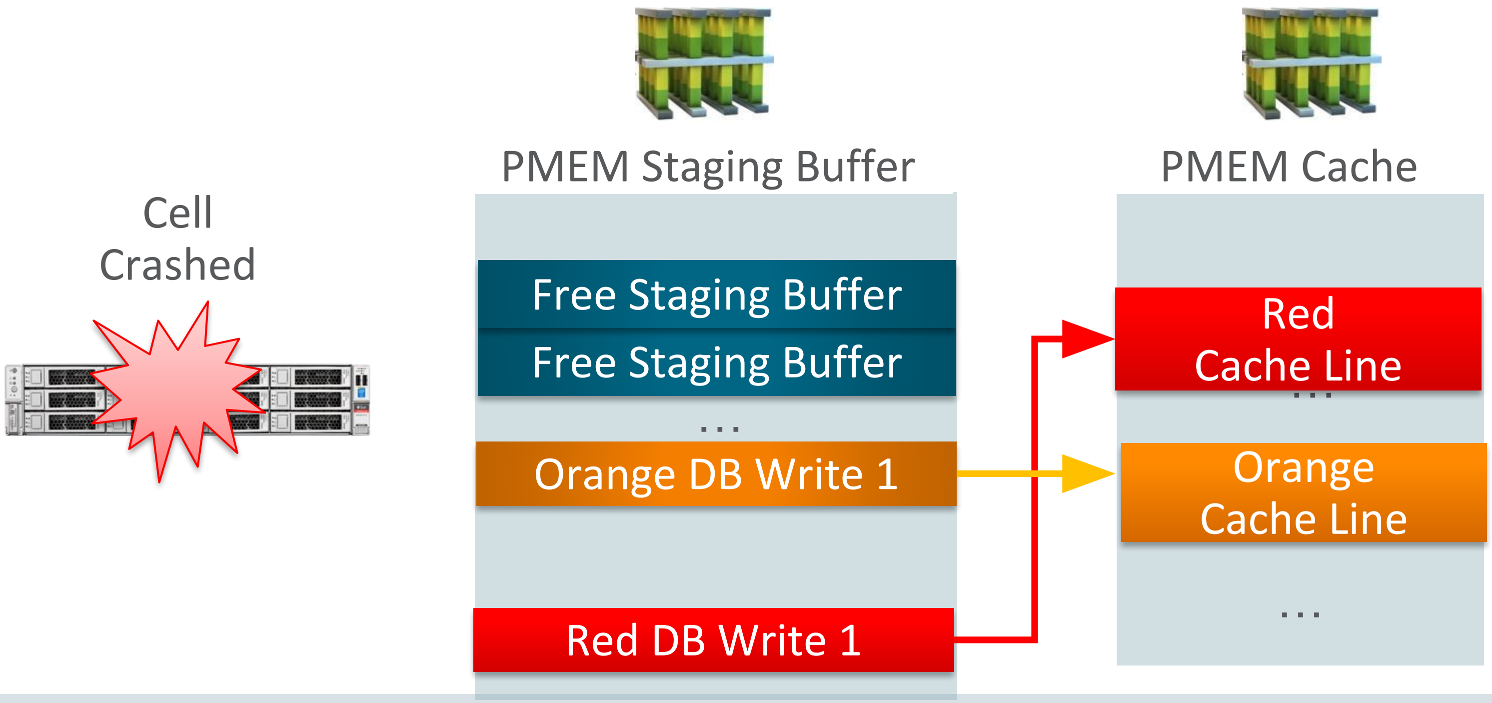
Staging Buffers Guarantee Atomicity



Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

Storage software recovers DB Writes After Crash

- Upon Startup, storage software scans PMEM Staging Buffers and Applies to PMEM Cache
- No Client I/O Accepted Until Recovery is Completed

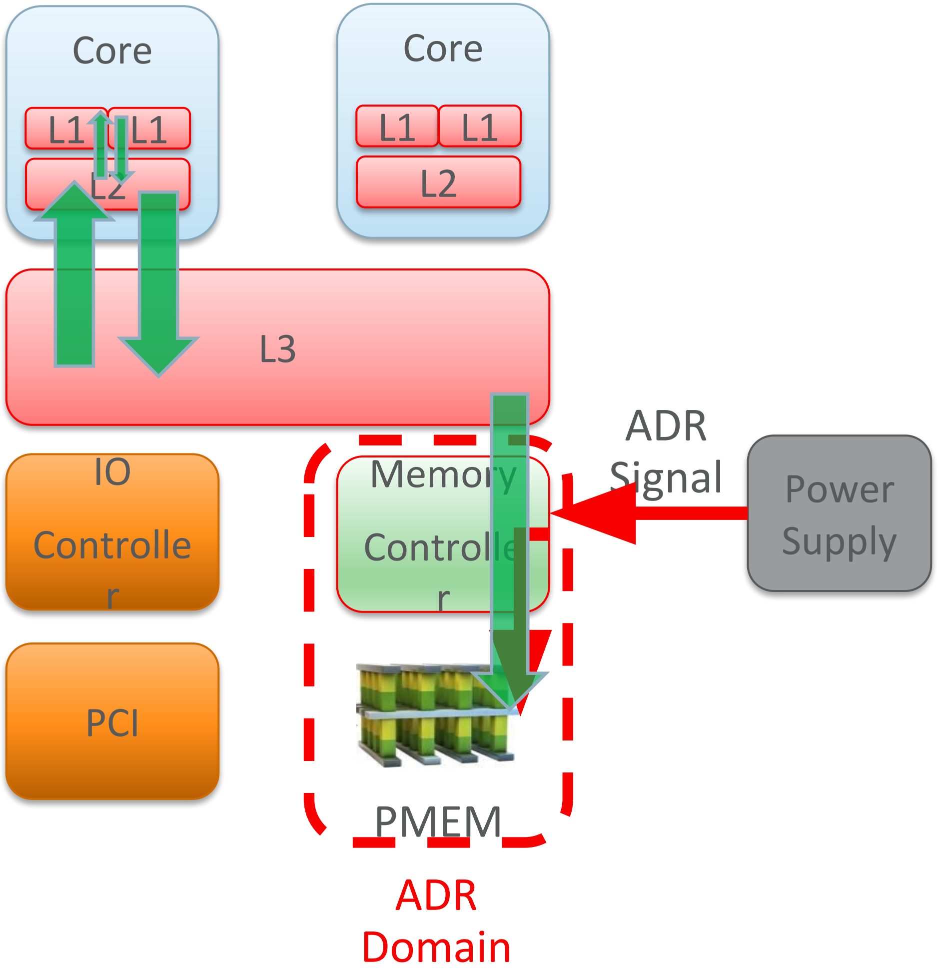


Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

Achieving Persistence

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

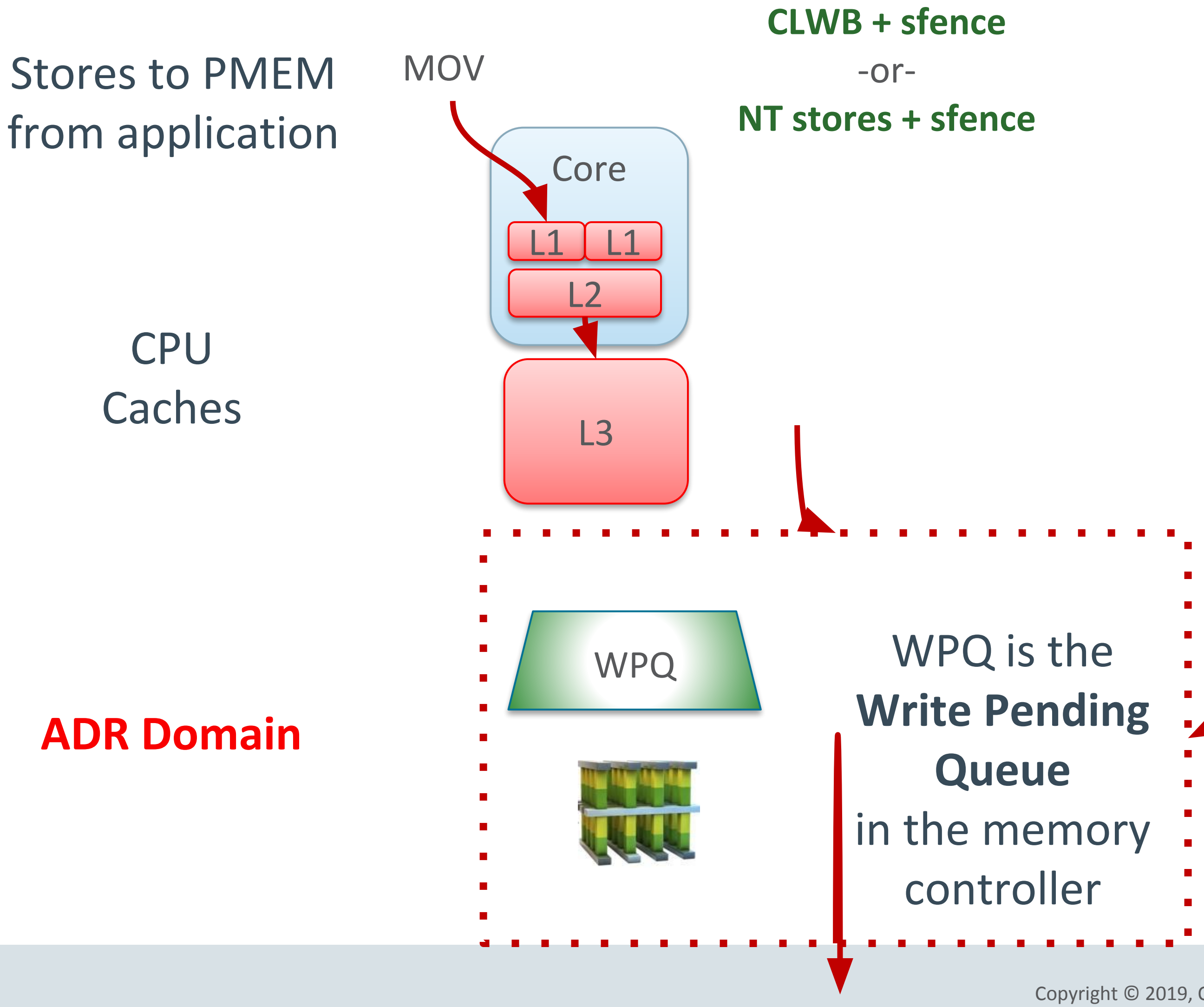
PMEM Data Persistence: ADR



- Asynchronous DRAM Refresh (ADR)
  - Saves Updates upon Power Fail
  - Only Data Inside ADR Domain is Protected Against Power Fail

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

PMEM Data Persistence: CPU Cache



- Stores in CPU Cache are NOT Persistent
  - Use Non-Temporal Stores to Bypass CPU Cache
    - _mm512_stream_si512/_mm_stream_si64 + sfence (for ordering)
  - Use CLWB to Flush into ADR Domain
    - CLWB -> flush from cacheline
      - _mm_clwb_ + sfence

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

Current Solution: NT Store + sfence

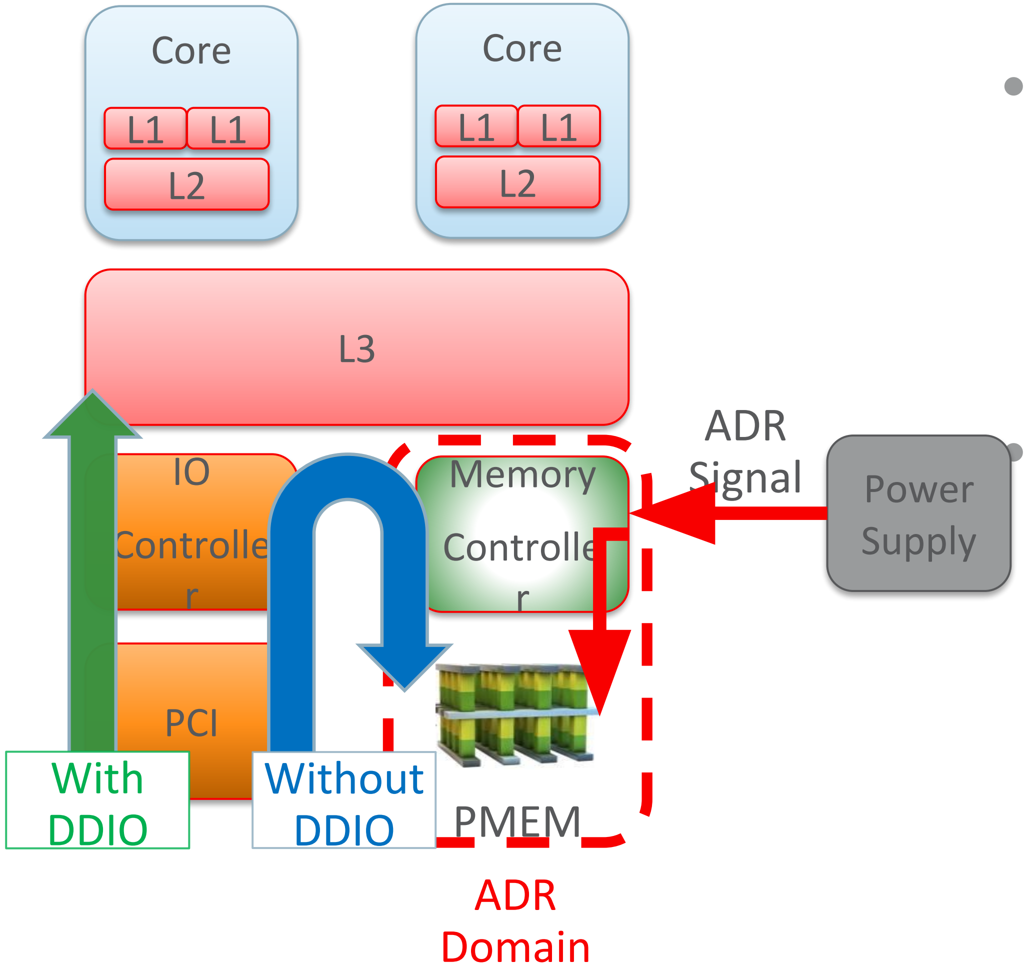
```
void do_PMEM_write(…)
{
    for (…)
    {
        do_non_temporal_store(…); // write 8/64 bytes at a time
    }

    sfence(); // BUG!
}
```

- sfence should be inside the loop, otherwise stores appear out of order
- Although example is trivial, memory ordering not easy for most developers.
- **Future enhancement to ADR** – all stores to persistent memory will be ordered and durable.

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

Persistence for RDMA writes – DDIO



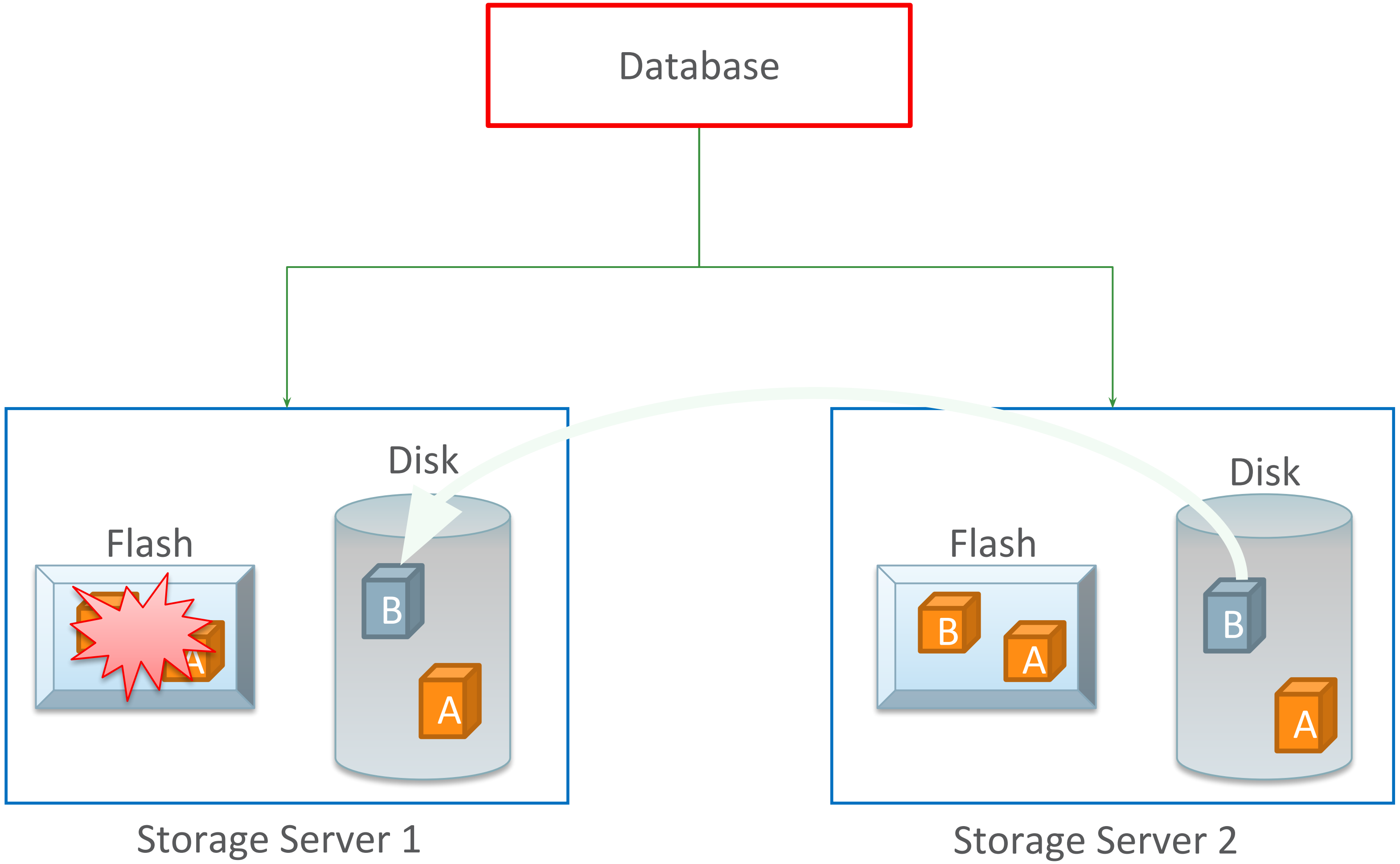
- Data Direct IO (DDIO)
  - Performs Write Allocate/Update into L3 Cache
  - Data outside ADR Domain
    - Not Durable upon Power Fail
- Turn off DDIO for network card PCI Slot in BIOS
  - Global Knob per PCI Device:
    - Affects DMA to both DRAM and PMEM
  - Impacts write performance due to L3 misses

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

High Availability on PMEM

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

HA Before PMEM



- On flash failure
  - Storage server software stack stays up
  - Software is notified of IO failure
- Resilver from mirror using in-memory data
  - Fine-grained

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

PMEM: IO Failures without RAS

- Writes to PMEM may fail, resulting in poison
- Poison Read Generates Uncorrectable Error (UE)
  - Machine Check Exception (MCE) -> kernel panic (similar to behavior upon DRAM error)
  - Lose all state and in-memory data on the storage server

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

Current PMEM RAS

- PMEM Address Range Scan (ARS) runs after bootup in the background
  - ARS logs bad blocks in sysfs
  - Storage server software monitors bad block list, to avoid using PMEM DIMMs with bad blocks
  - Data cached by failed PMEM DIMMs copied from mirror

Future Enhancement: PMEM RAS

- Software Gets SIGBUS on poison read
- Fine-Grained Resilver of poisoned data
  - Copy stale data from mirror
  - Similar to flash cache
- Currently testing, but challenges remain

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

RAS Kernel Panics

- Currently, poison can still cause kernel panics even with RAS
  - Only one MCE register per core
  - Concurrent read of poison from different HW threads leads to overflow and kernel panic
  - On-demand read and cache prefetch are different HW threads

- Only recovery method is to scan list of bad blocks upon restart, and identify and quarantine PMEM DIMM.

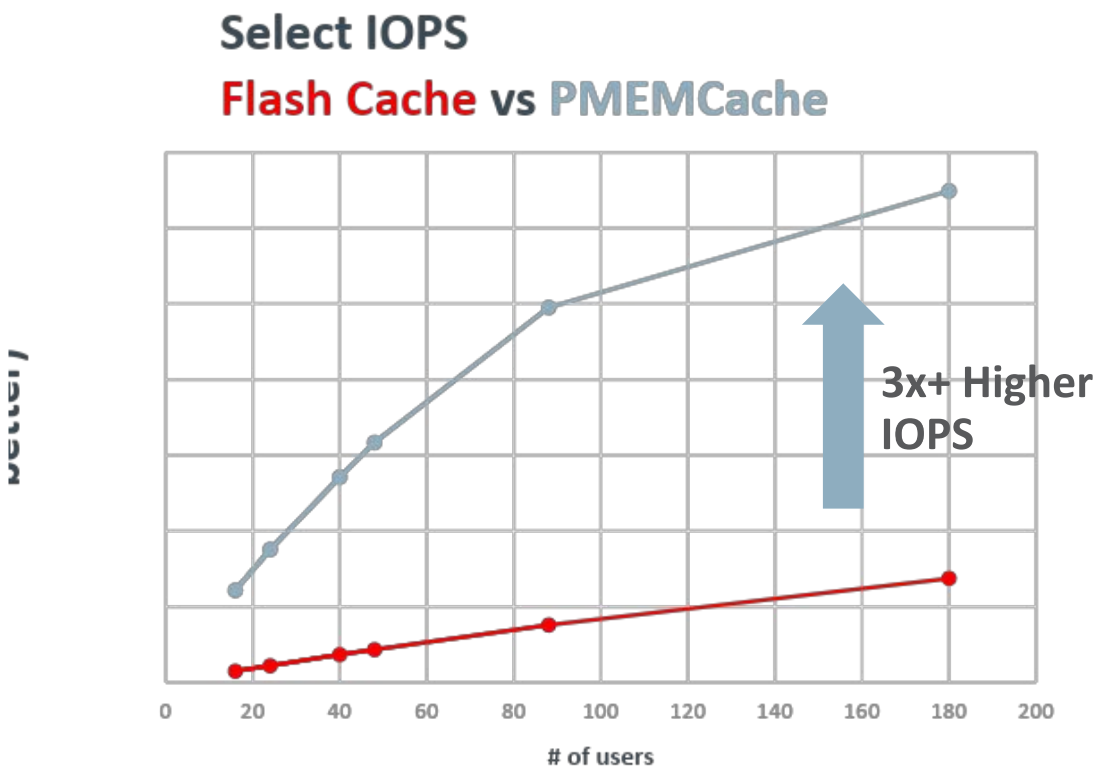
- Future HW improvement to fix overflow

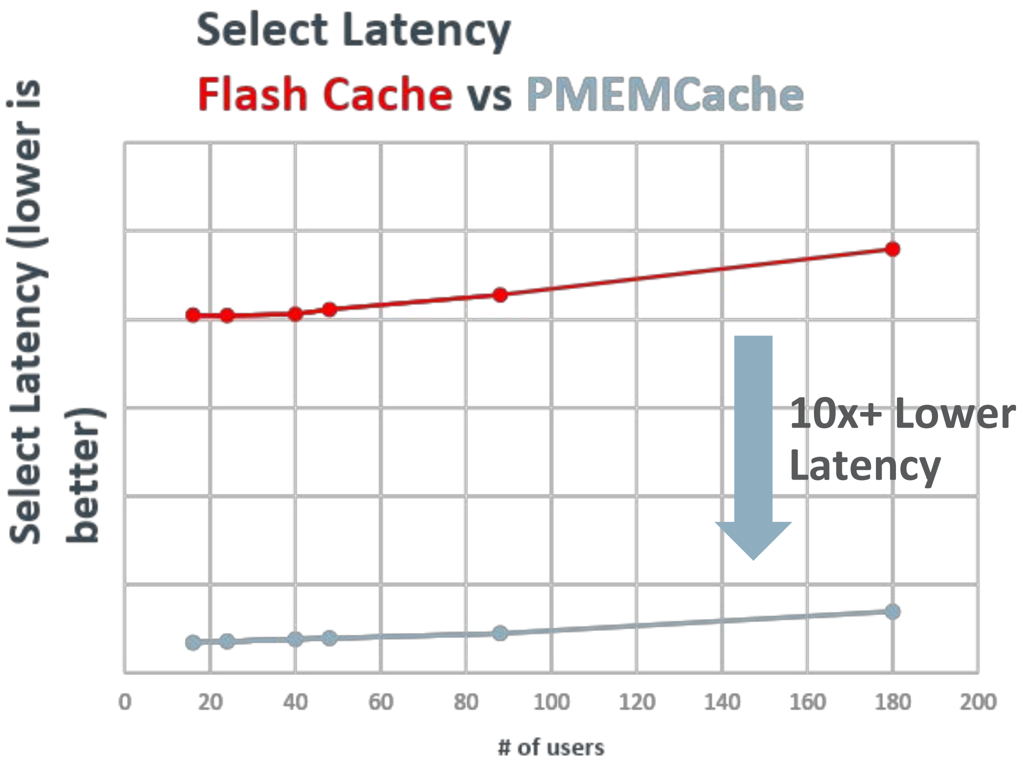
Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

Performance

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

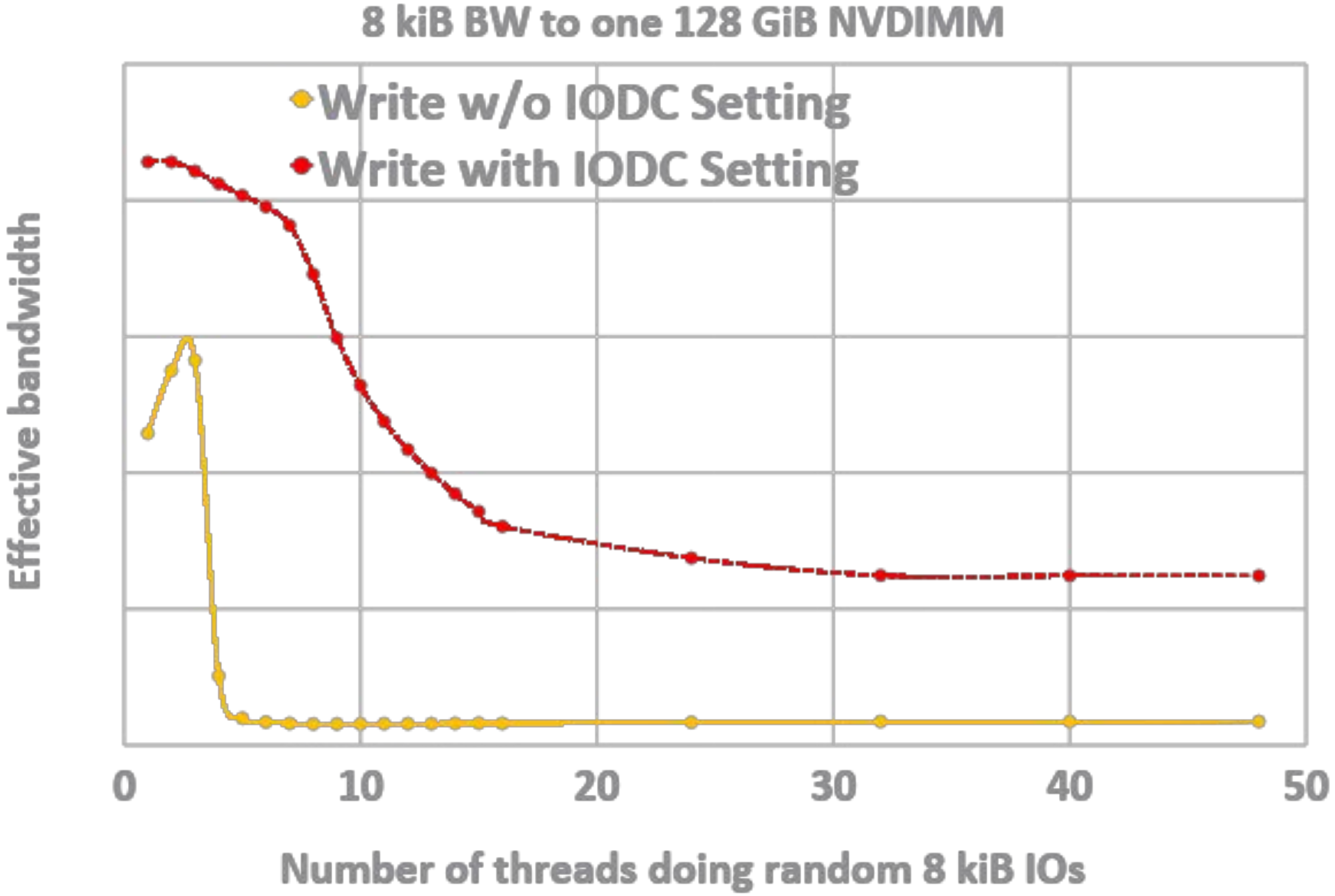
Amazing PMEMCache Read Performance – ¼ rack





Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

Future enhancement: write scaling performance



- Currently, write scales poorly with concurrent writes
    - Especially with writes from remote core
- IO Directory Cache contention
    - Specific IODC setting reduces contention
    - Still room for future improvements
- Each thread slows down writes from another thread
- Best performance with fewer threads doing IOs

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

Lessons Learned

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

What Worked with using PMEM and RDMA?

- Read performance – IOPS and Latency
- Persistence (excluding user bugs)
  - Ran many persistence tests, barely any issues

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

What can be enhanced?

- ADR
- RAS
- Threaded write performance

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |

Questions?

Copyright © 2019, Oracle and/or its affiliates. All rights reserved. |