ENDURANCE BENEFITS WITH PERSISTENT MEMORY

Sergey Vinogradov

Software Development Engineer
Intel

Storage Performance Development Kit (SPDK)
Persistent Memory Development Kit (PMDK)
Intel® VTune™ Profiler

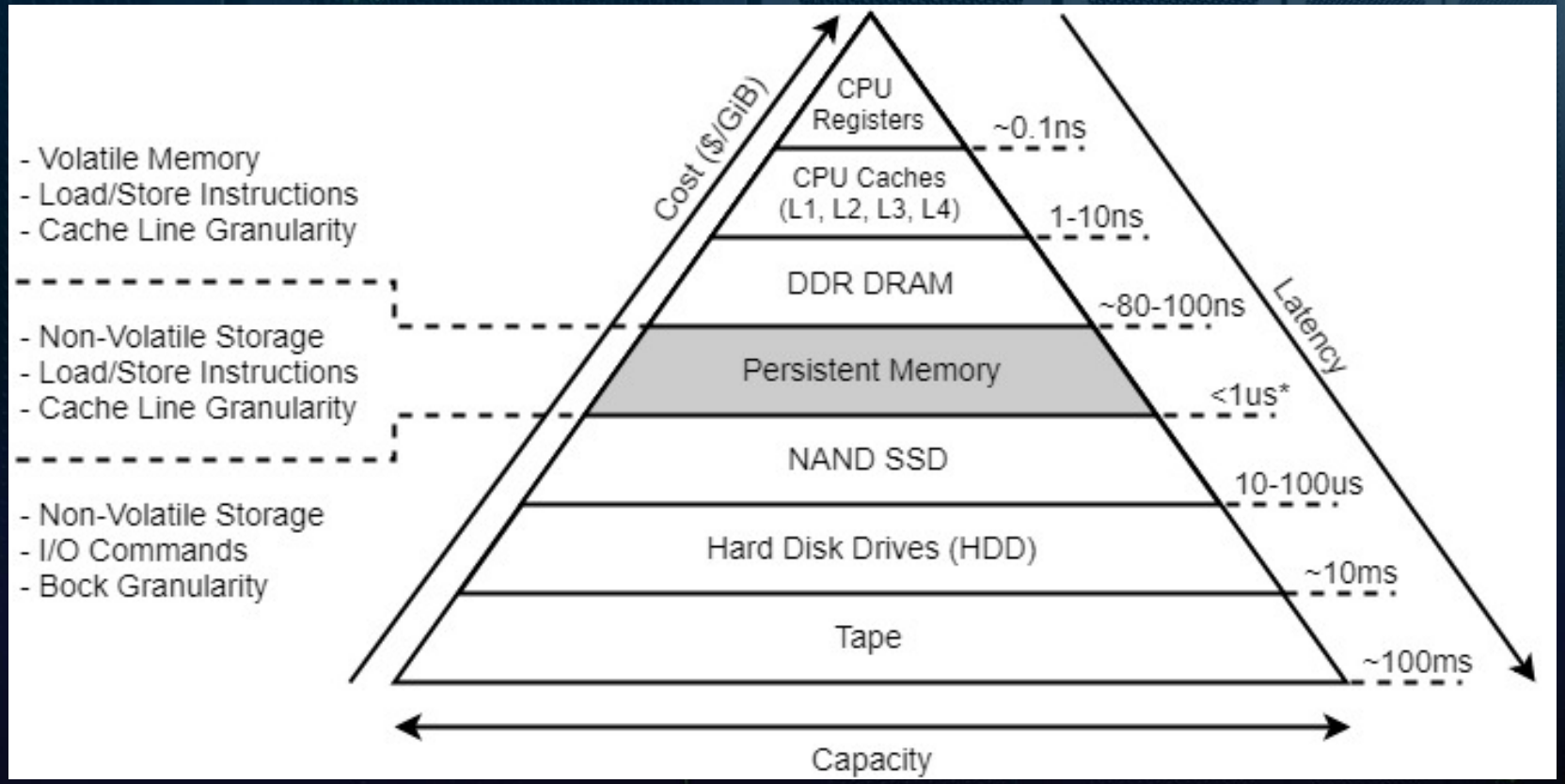
Virtual Forum

AGENDA

1. Introduction
2. Endurance characteristics
3. Write Amplification in software
   Software modernization with Persistent memory
4. Summary

INTRODUCTION

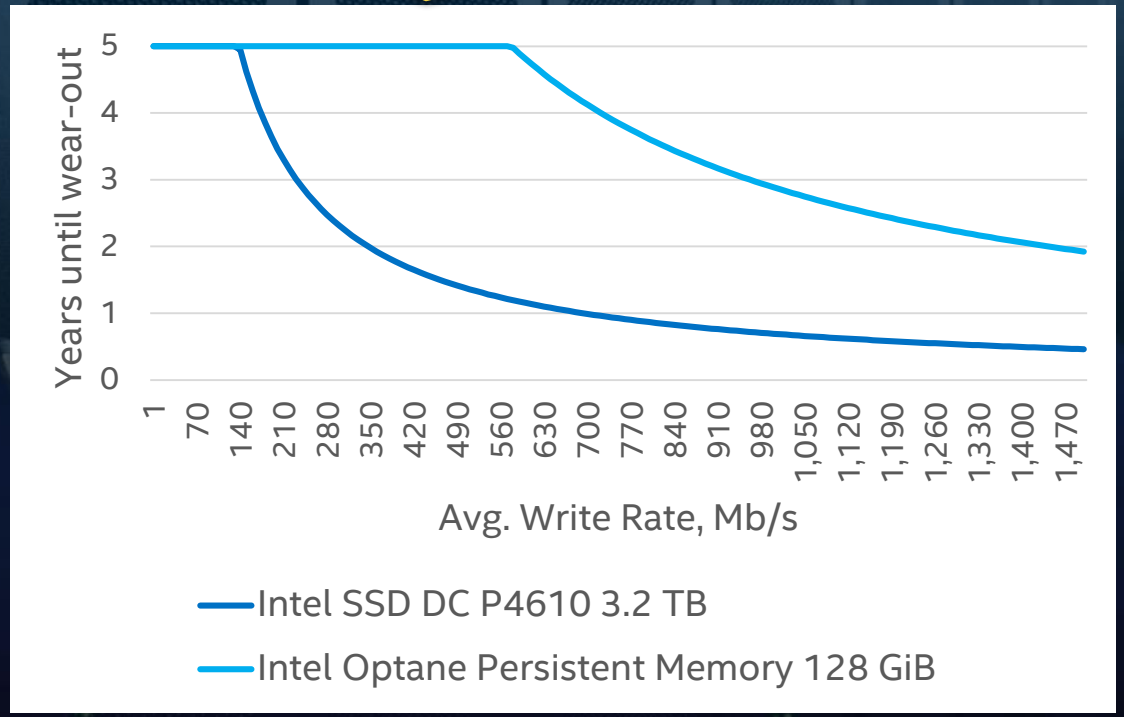
- Intel® Optane™ Persistent Memory has better performance characteristics comparing to NAND SDD. But… **What about endurance?**



ENDURANCE CHARACTERISTICS OF PERSISTENT MEMORY

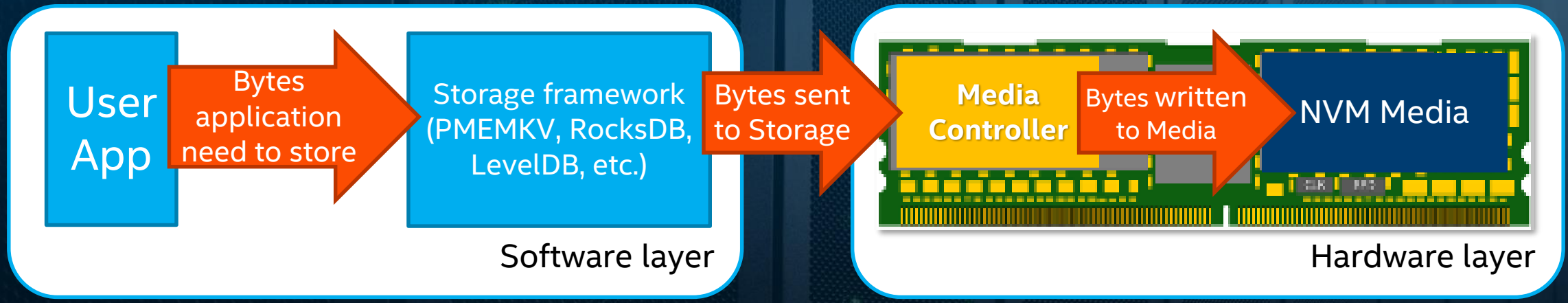
Persistent Memory survives better on write-intensive workloads.

Params	SSD DC P4610 3.2 Tb	Optane Persistent Memory 128 Gib	
size	3.2 Tb	128 Gb	
Warranty	5 years	5 years	
Endurance, 100% Writes	21.85	64B	256B
		91 PBW	292 PBW
DWPD	3.74	389.55	1250
Max Avg. Write Rate	138.5 MB/sec	577 MB/sec	1851 MB/sec



* For Intel Optane Persistent Memory we take the worst case Endurance

DEFINITION OF WRITE AMPLIFICATION (WA)



- Software write amplification caused by SW
  - Algorithm may duplicates data writes
  - Write granularity may cause Write Amplification
- Hardware write amplification caused by HW/FW

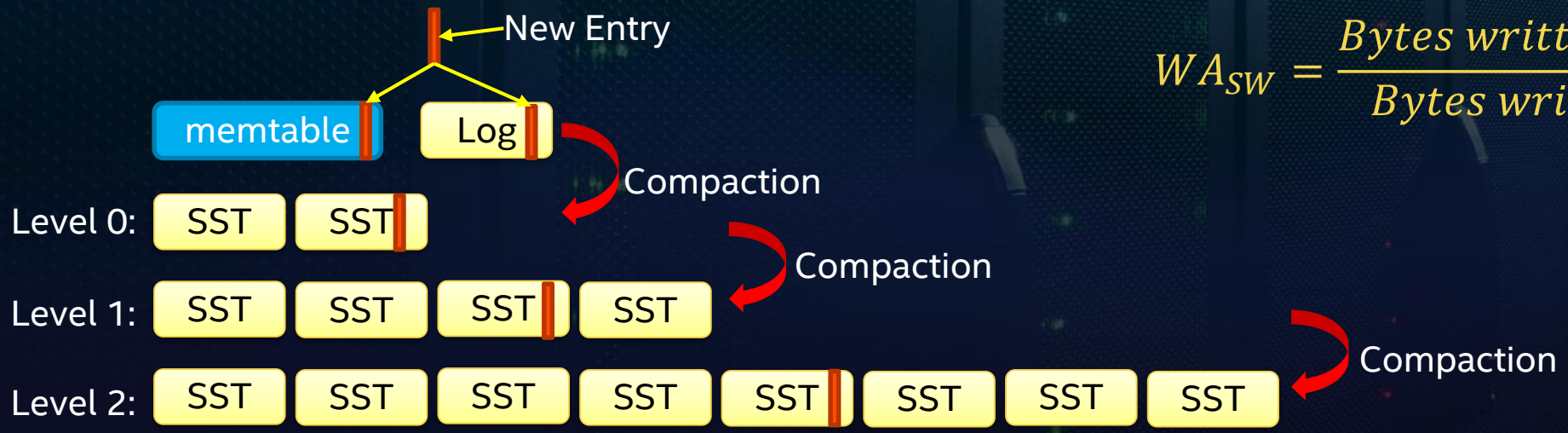
$$WA_{SW} = \frac{Bytes\ sent\ to\ storage}{Bytes\ app\ want\ to\ store}$$

$$WA_{HW} = \frac{Bytes\ written\ to\ media}{Bytes\ sent\ to\ storage}$$

$$WA = WA_{SW} \times WA_{HW}$$

SOFTWARE WRITE AMPLIFICATION IN LSM TREE

- Log Structured Merge (LSM) tree is a key/value storage algorithm
  - Designed for block storage device
- Compaction causes Software Write Amplification

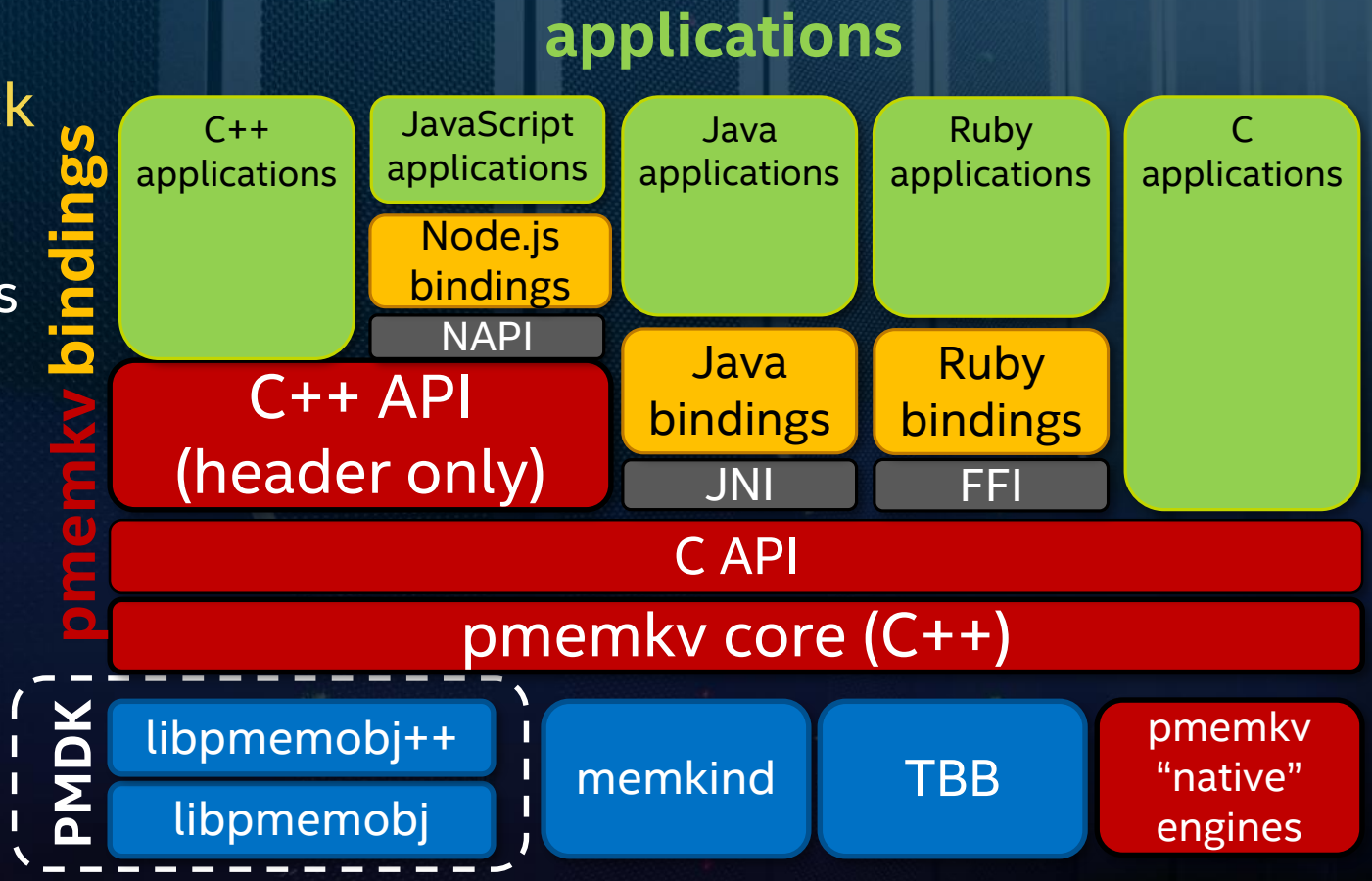


$$WA_{SW} = \frac{Bytes\ written\ to\ the\ disk(SST)}{Bytes\ written\ to\ MemTable}$$

CAN WE OPTIMIZE SOFTWARE FOR PERSISTENT MEMORY?

We can compare PMEMKV with popular LSM-tree based frameworks

- LSM-tree designed for a block storage devices
  - Write optimized but the cost is write amplification
  - RocksDB and LevelDB are popular implementations
- **PMEMKV** is a key-value storage developed for Persistent Memory



SOFTWARE WA: PMEMKV VS ROCKSDB VS LEVELDB

- PMEMKV, RocksDB and LevelDB use the same benchmarking tool - db_bench
  - RocksDB and LevelDB use Persistent memory as a block storage device (Storage over AppDirect)
  - PMEMKV works in AppDirect mode (avoids page cache)
- Selected **3 write-intensive benchmarks**
  - **fillseq –** insert entries to the empty storage in sequential order
  - **fillrandom –** insert entries to the empty storage in random order
  - **overwrite –** overwrite existing entries in prefilled storage in random order
- Measured Write Amplification for **3 different data set sizes**
  - 10M entries, 20M entries, 30M entries

METHODOLOGY TO MEASURE SOFTWARE WA

- db_bench reports the **Raw Size**, which is cumulative size of entries inserted by the benchmark
- PMU uncore counters allow to monitor writes to Persistent Memory
  - **UNC_M2M_IMC_WRITES.TO_PMM**
  - **UNC_M_PMM_WPQ_INSERTS**
  - Counts number of cache lines sent to PMEM
- **EMON** is used to collect PMU events
  - Part of Intel® VTune™ Profiler
  - Perf can be used alternatively

$$WA_{sw} = \frac{Written\ to\ PMEM}{Raw\ Size}$$

Written to PMEM – measured by Uncore counter

Raw Size $= \#entries * sizeof(entry)$ reported by benchmarks

SOFTWARE WA: PMEMKV VS ROCKSDB VS LEVELDB

Bench	PMEMKV (csmap)		RocksDB		LevelDB	
	Written to PMEM, Mb	**WA**	Written to PMEM, Mb	**WA**	Written to PMEM, Mb	**WA**
10 000 000 entries, Raw Size = 7782 Mb						
fillseq	32 378	**4.16**	35 068	**4.51**	35 932	**4.62**
overwrite	20 143	**2.59**	1 853 610	**238.19**	271 942	**34.95**
fillrandom	27 800	**3.57**	92 446	**11.88**	243 860	**31.34**
20 000 000 entries, Raw Size = 15564 Mb						
fillseq	64 294	**4.13**	70 439	**4.53**	71 523	**4.60**
overwrite	40 285	**2.59**	2 191 380	**140.80**	599 998	**38.55**
fillrandom	55 637	**3.57**	219 580	**14.11**	551 925	**35.46**
40 000 000 entries, Raw Size = 31128 Mb						
fillseq	128 196	**4.12**	141 099	**4.53**	140 611	**4.52**
overwrite	80 575	**2.59**	2 478 590	**79.63**	1 307 440	**42.00**
fillrandom	111 377	**3.58**	588 285	**18.90**	1 221 960	**39.26**

Key size = 16 bytes.
Value size = 800 bytes

$$WA_{sw} = \frac{Written\ to\ PMEM}{Raw\ Size}$$

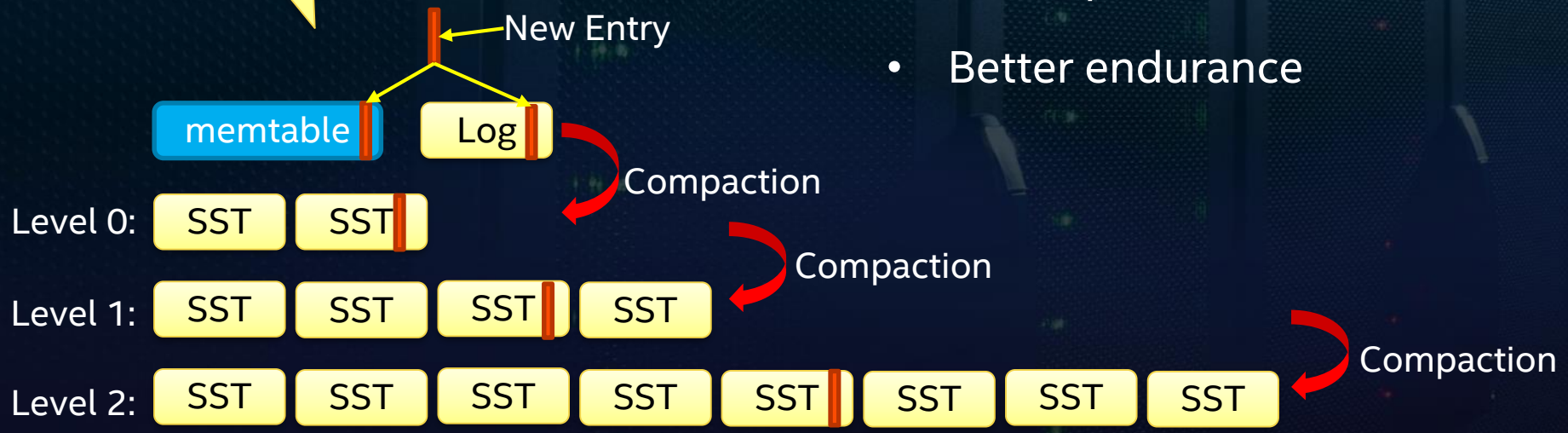
Written to PMEM – measured by Uncore counter

Raw Size = *#entries*sizeof(entry)* reported by benchmarks

POSSIBLE SOFTWARE MODERNIZATION (1/4)

No code changes.
Use existing software!

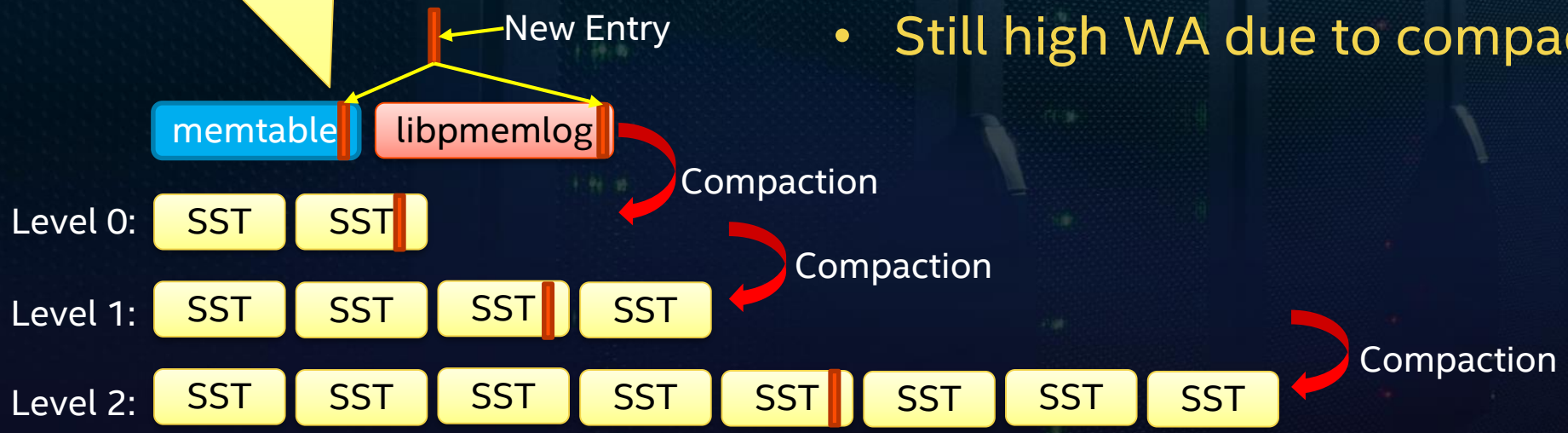
- Run existing software on Persistent Memory
  - Storage over AppDirect mode
- Intel Optane Persistent Memory vs SSDs
  - Better performance
  - Better endurance



POSSIBLE SOFTWARE MODERNIZATION (2/4)

Minor modifications.
Use Libpmemlog as a memtable log.

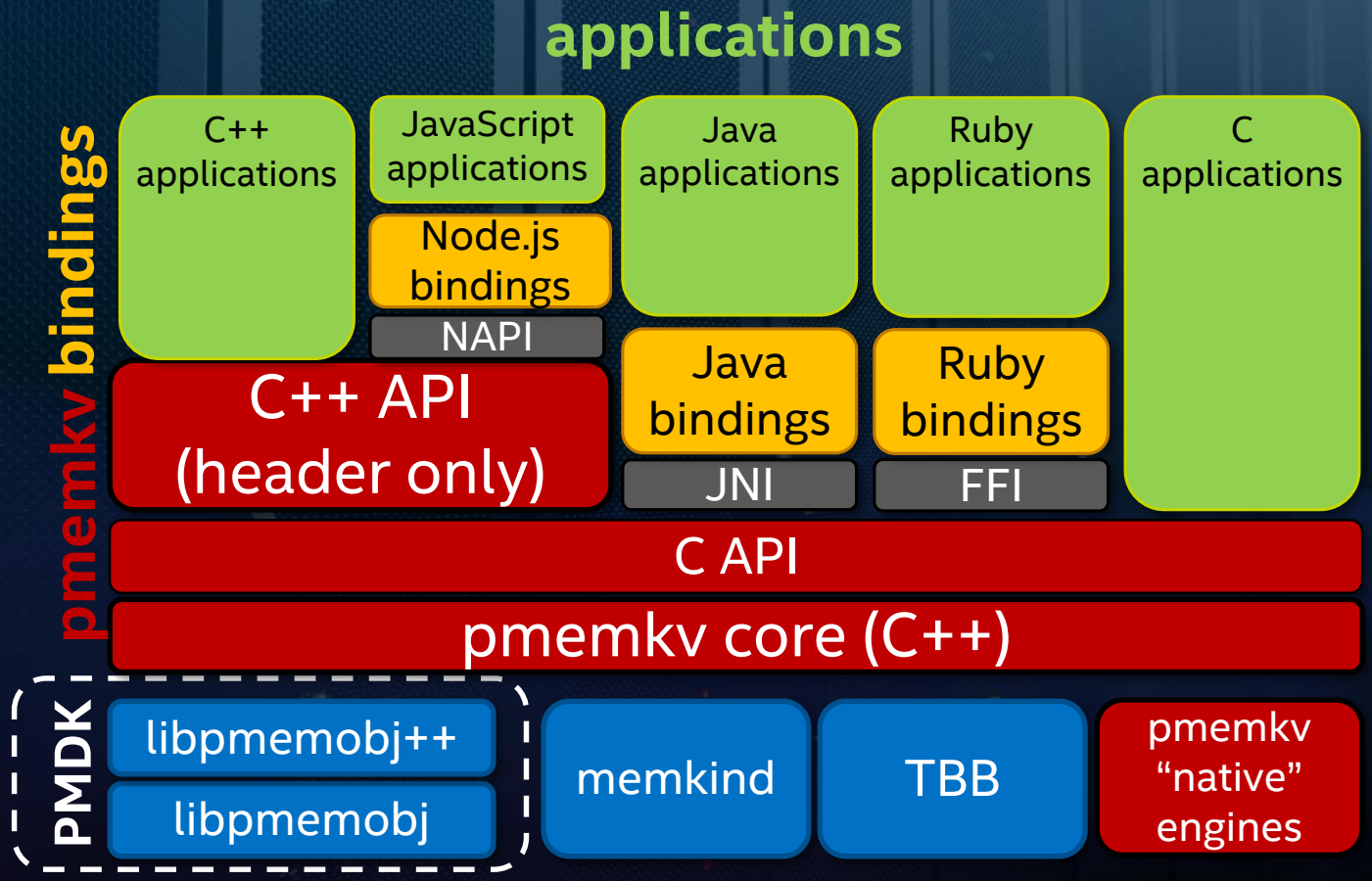
- **Libpmemlog** - persistent memory resident log file
  - Non-paged direct access
  - Improves insert performance
- Still high WA due to compaction of SSTs



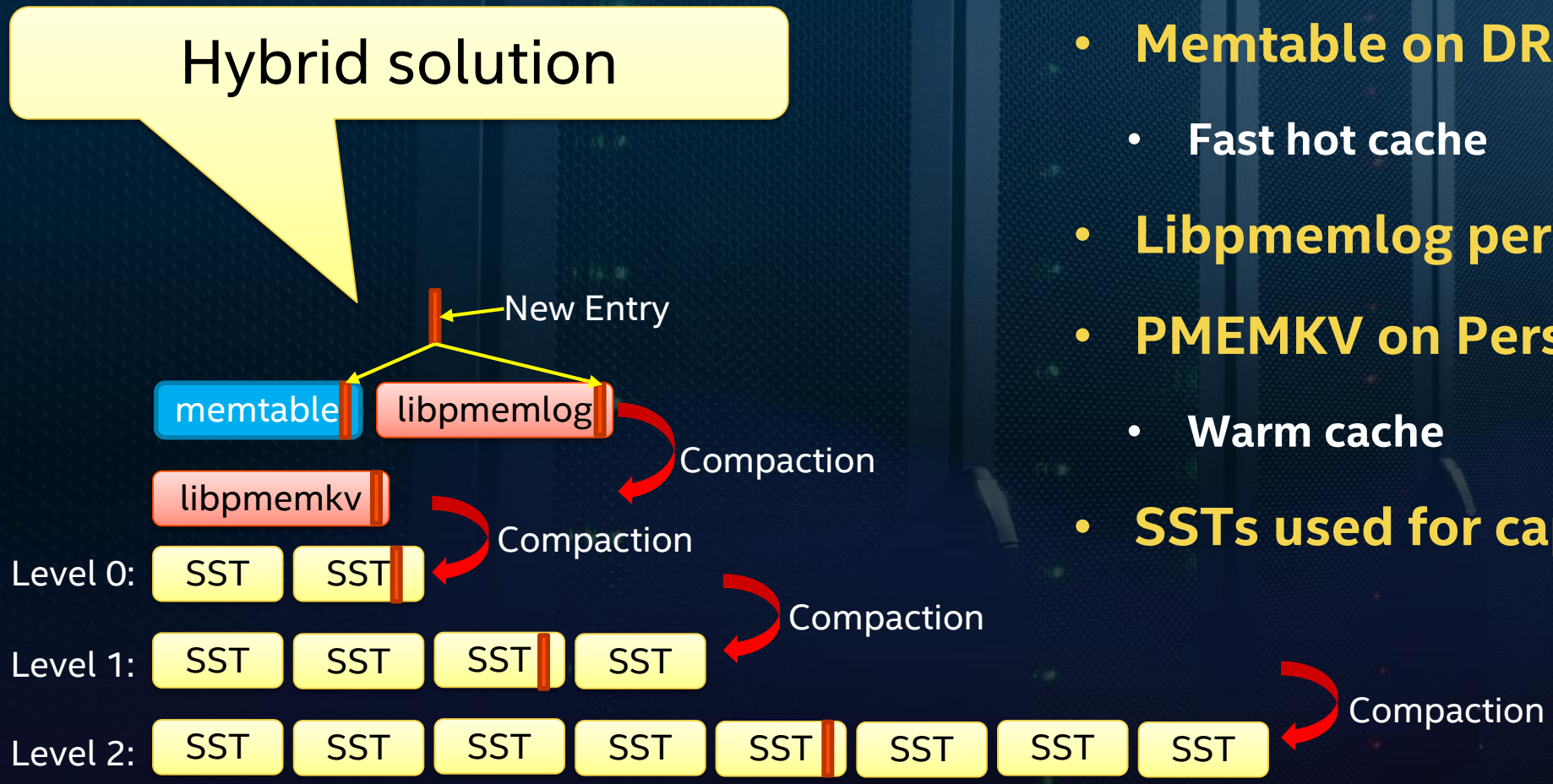
POSSIBLE SOFTWARE MODERNIZATION (3/4)

Replace existing key/value storage with PMEMKV

- Developed for Persistent Memory
- Pluggable engines
  - Native engines for persistent and volatile use-cases
- Native API for C/C++
- Easy integration with high-level language bindings



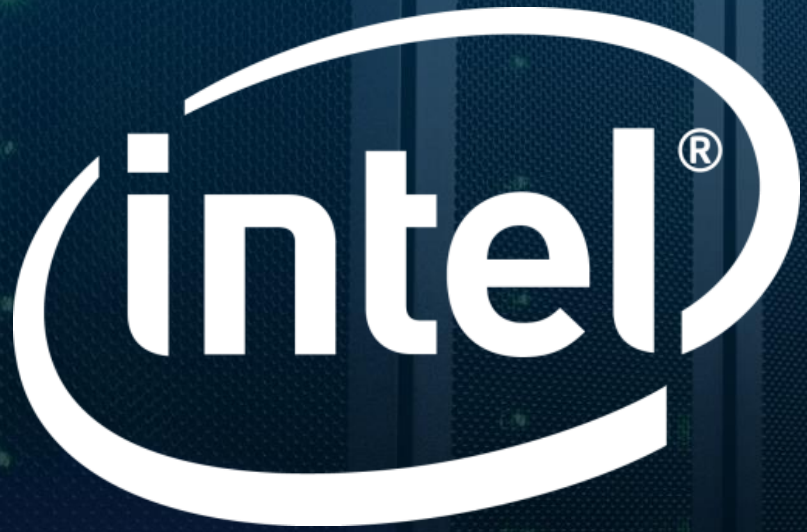
POSSIBLE SOFTWARE MODERNIZATION (4/4)



- **Memtable on DRAM**
  - **Fast hot cache**
- **Libpmemlog persistent log**
- **PMEMKV on Persistent memory**
  - **Warm cache**
- **SSTs used for capacity**

SUMMARY

- Intel Optane Persistent Memory is a new layer in a storage hierarchy
  - better endurance comparing to NAND SSD
  - better performance comparing to NAND SSD
- Intel Optane Persistent Memory might help to decrease TCO on write intensive workloads if SSD wear-out quickly
- PMDK provides set of libraries for easier code modernization and enabling Persistent Memory in applications



Storage Performance Development Kit (SPDK)
Persistent Memory Development Kit (PMDK)
Intel® VTune™ Profiler

Virtual Forum