SDC
STORAGE DEVELOPER CONFERENCE
SNIA SANTA CLARA, 2015

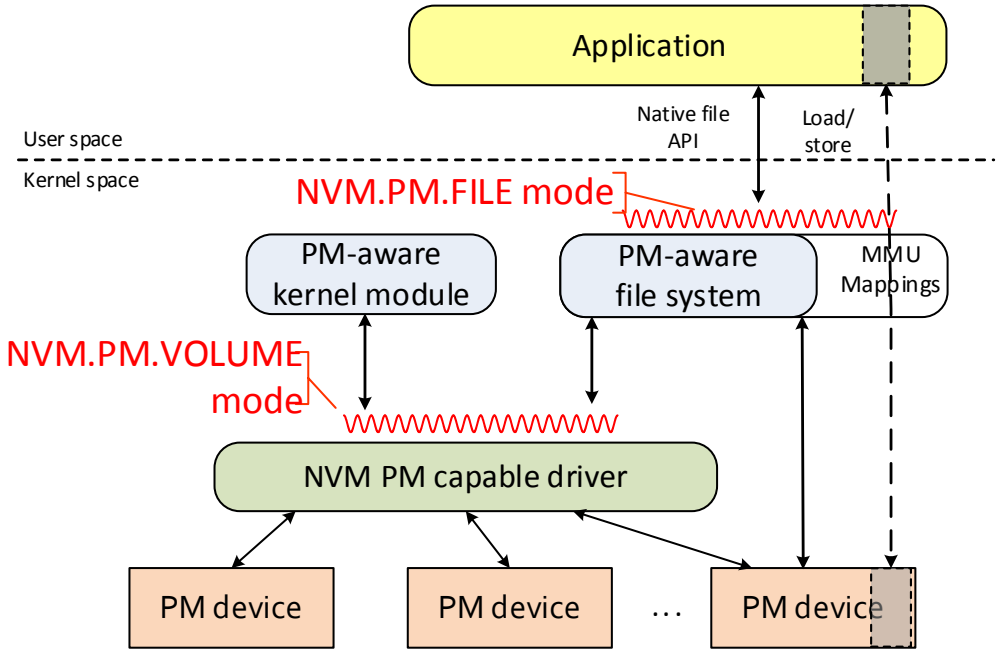
Load-Sto-Meter: Generating Workloads for Persistent Memory

Damini Chopra, Doug Voigt
Hewlett Packard (Enterprise)

Application vs. Pure Workloads

- Benchmarks that reproduce application workloads
  - Assist in system provisioning
  - Assist in tuning for nuances of particular applications
  - Get more accurate results
- Pure workload generators
  - Assist in characterizing basic system performance
  - Assist in locating root causes of performance issues
  - Evaluate design trade-offs in prototype systems
  - Explore best/worst application performance
  - Enable finer grained analysis than profiling

Difference Between IO and PM Workloads

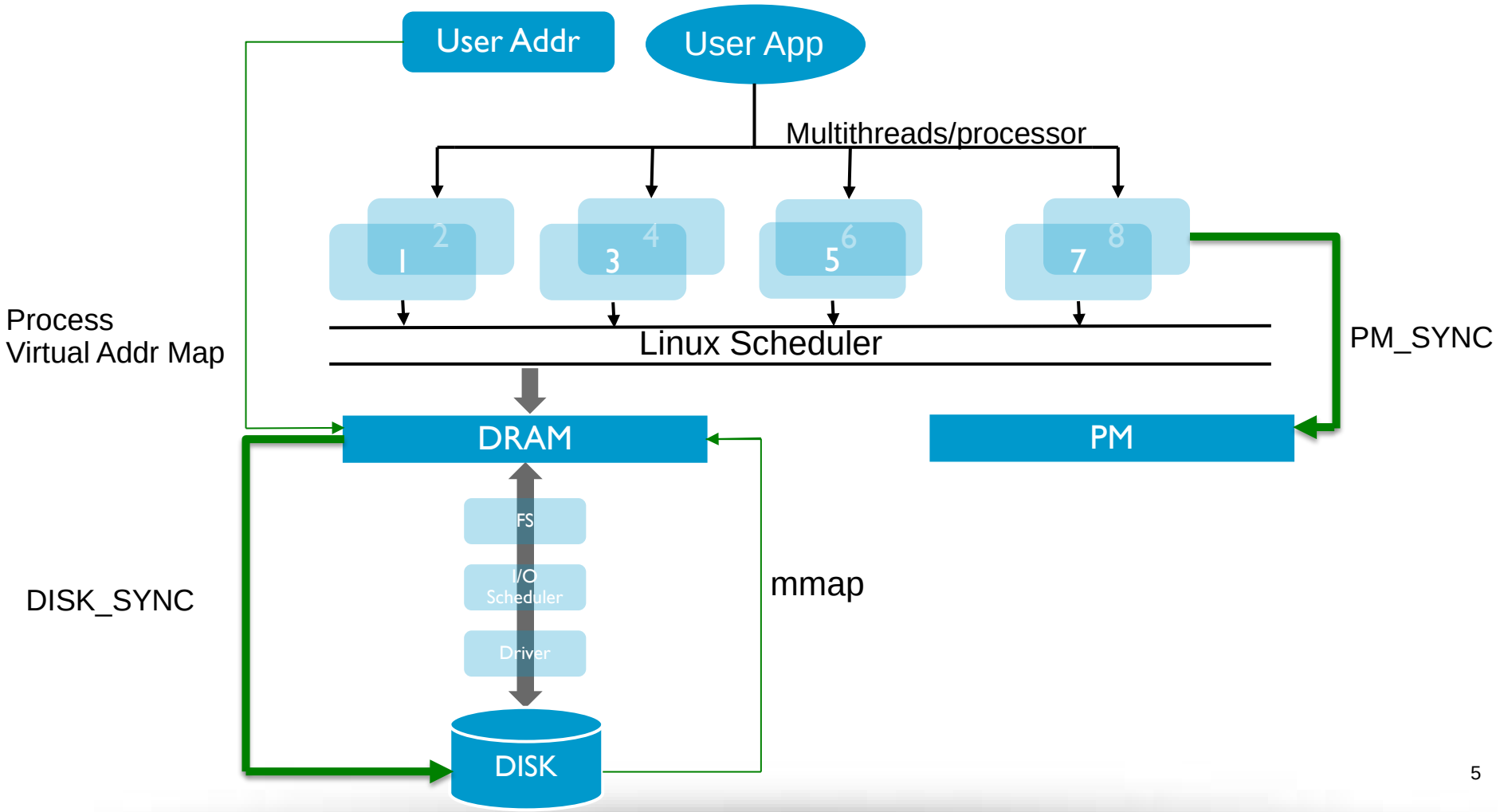


- Byte Addressability
- No software involved in CPU Load/Store instructions
- Additional sync operation
  - Avoid context switch from user space to kernel space (no I/O syscalls)
  - Eliminate expensive sync operations to disk by introducing CPU – Memory syncs

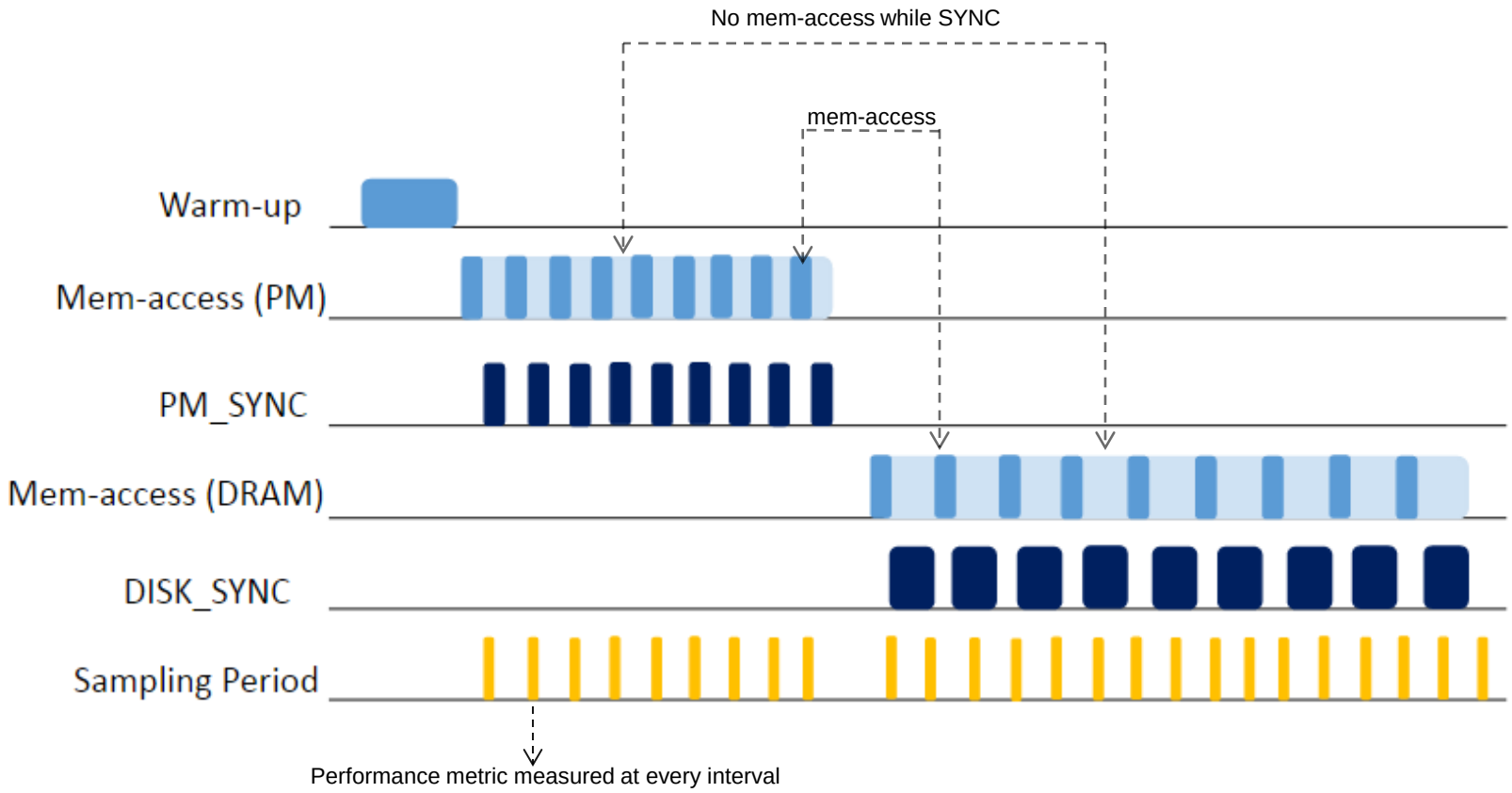
Role of Pure Workload Generator

- Analyze the performance and behavior of Persistent Memory
  - Tune the parameters of Load-Sto-Meter
  - Simulate real application behavior
- Provide aggressive Multithreaded Benchmarking
- Analyze PM sync mechanism versus DISK sync
  - in terms of performance and reliability
  - over various parameterized workloads
- Evaluate PM sync implementations

NVM Programming Model – Map and Sync



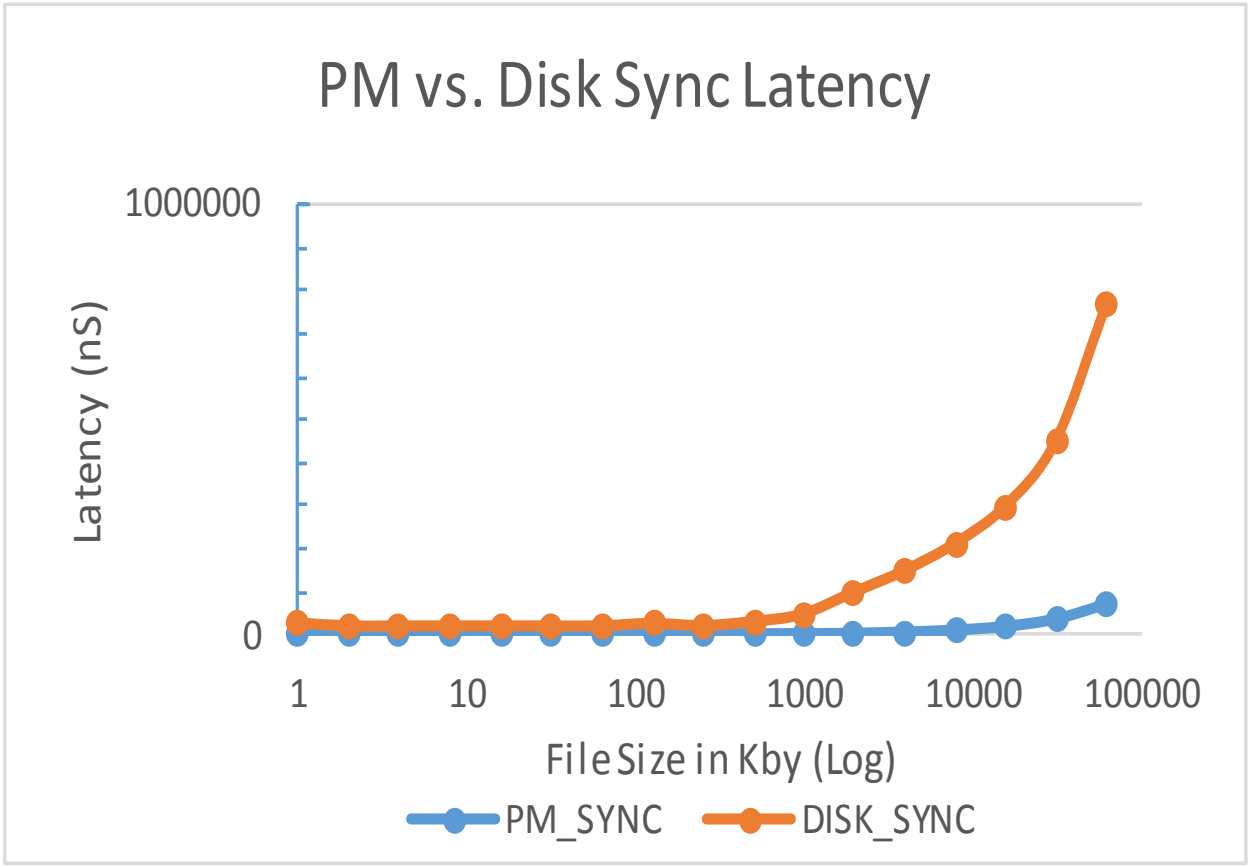
Basic Ld/St/Sync Template



Workload Generation Parameters

- Store to Sync ratio
- Read to Write Ratio
- Workload Threads
- Shared v/s Private memory access
- Sequential v/s Random memory access
- Granularity of Memory Access (Load/Store record size)
- PM Data Structures
- NUMA switch

Typical Output



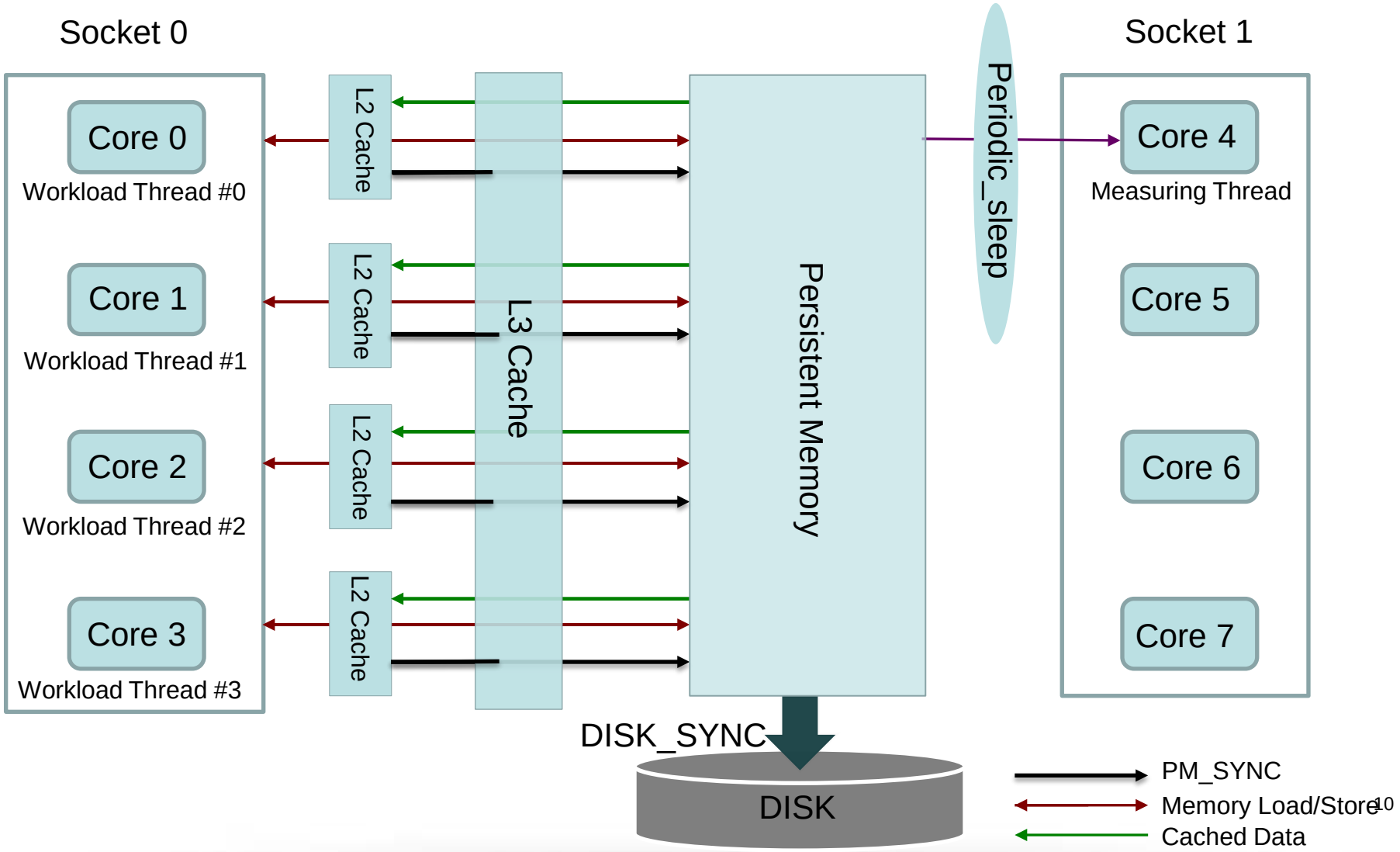
Performance Metric
Giga-Updates Per Second
Latency

Potential Measurement Domain
Filer Size
Granularity
Thread Count
Sync Ratio
Latency (vs GUPS)

Abstracting Syncs

- Sync implementation depends on many things
  - Disk vs. PM
  - User vs. Kernel space
  - Processor instruction set
  - Additional features such as High Availability
- Performance is sensitive to NVM.PM.SYNC implementations
- NVM Programming Model specifies sync/optimized flush abstraction
  - Application independent
  - Processor architecture independent
  - Implementation independent
- Workload generator should support multiple sync implementations
  - User plug in feature for their own implementation of sync
  - Easily test multiple PM aware file systems with the same workload

Multi-Threading



Sources of Performance Noise

- Trade-off between performance and thermal management policy of OS
- Partial utilization of CPU, causing discrepancies in acquiring pure load/store performance
- Hyper-threading management, while trying to embark equal workload on each core of the processor

A new type of workload generator is needed!

- Performance measurement specification?
  - Workload generation parameters
  - Definition of parameter driven behavior
- Implementations?
  - Commercial opportunity
  - Open source opportunity
    - SNIA SW TWG?
    - Open source contributions?

Load-Sto-Meter: Generating Workloads for Persistent Memory

Thank You