SNIA PERSISTENT MEMORY + COMPUTATIONAL STORAGE SUMMIT 2022

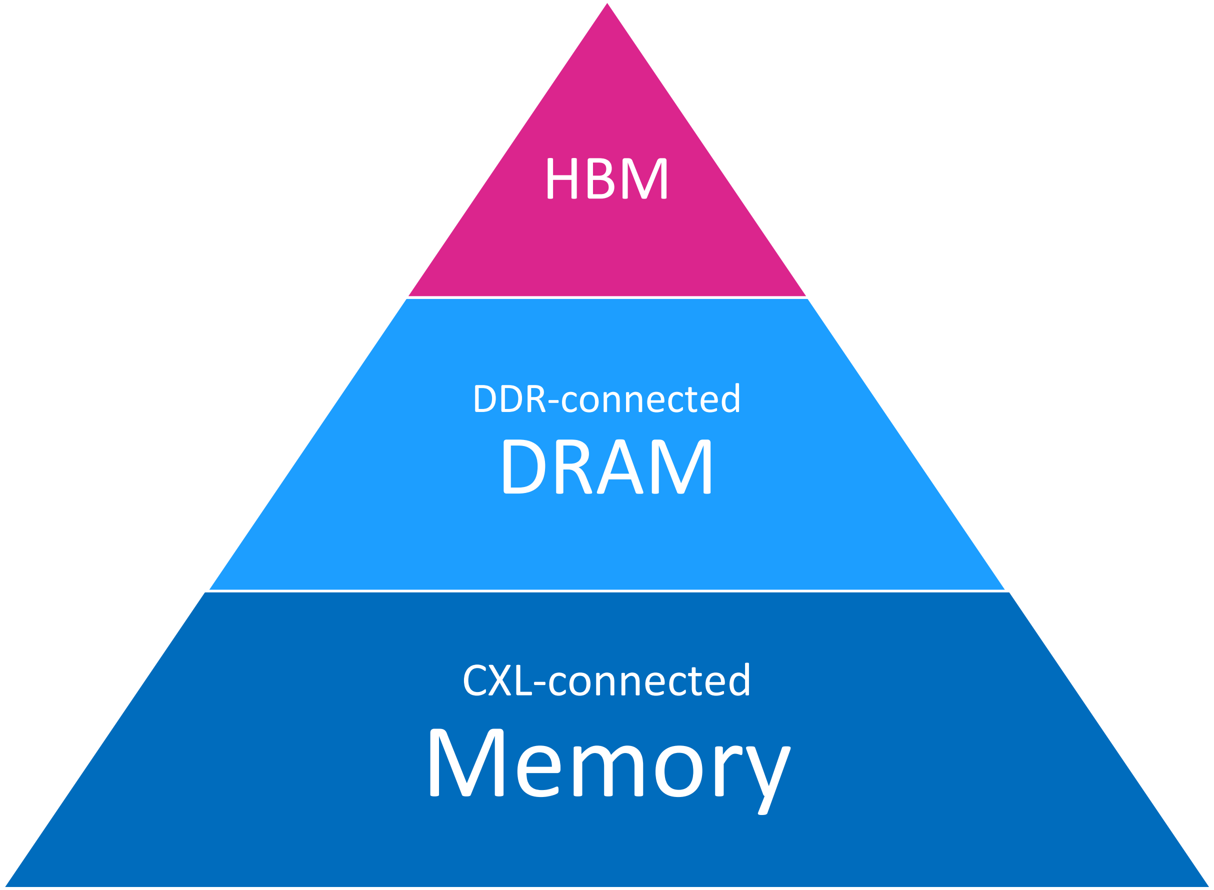
VIRTUAL EVENT • MAY 24-25, 2022

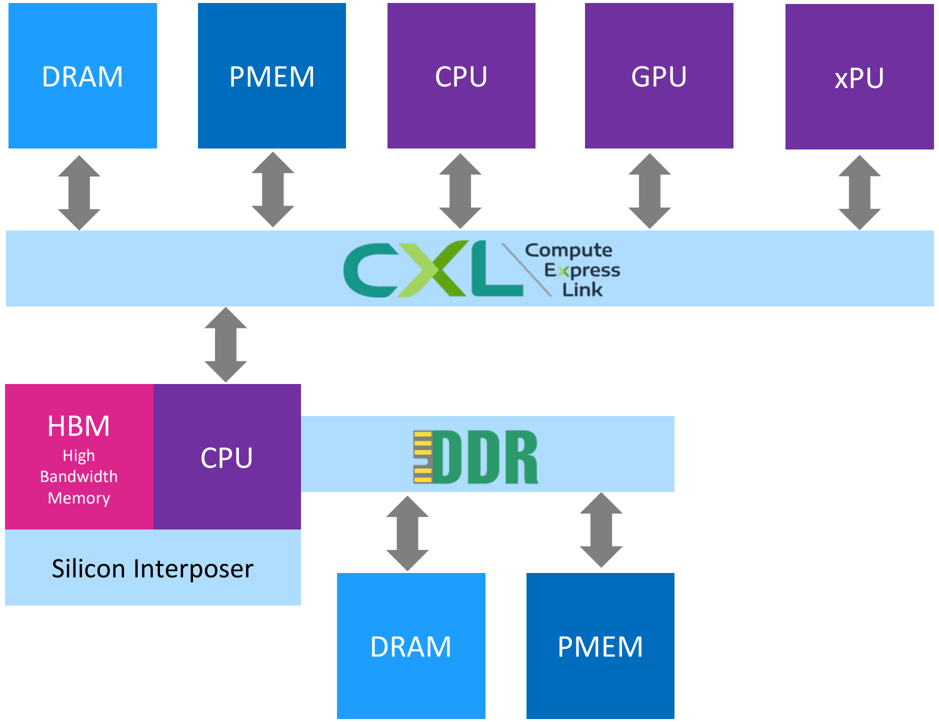
How CXL Will Change The Datacenter

Presented by Bernie Wu, VP Business Development

bernie.wu@memverge.com

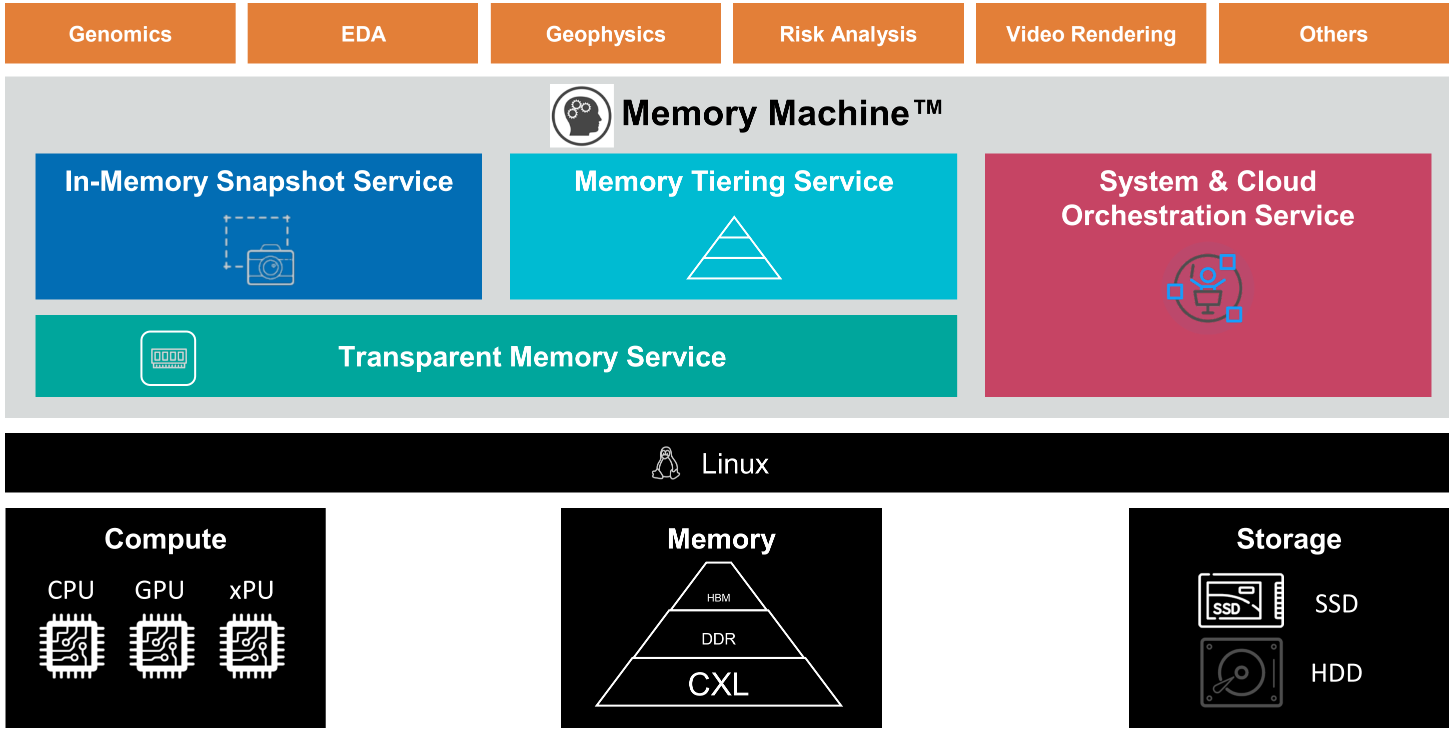
CXL – A game changer





But hardware Innovation Alone Is Not Enough!

MemVerge Memory Machine™



Memory Virtualization

- Software-defined Memory Pool
- Intelligent Auto-tiering
- **Delivers big memory capacity without application change**

Memory Snapshot Service

- Fully captures running application state
- An Application can be rolled back, restored or cloned from anywhere at any time
- **Delivers higher application mobility and availability**

Memory Object Sharing

- Enables Objects to be stored/shared in-memory with tiering into storage

Use cases

- Genomics — secondary and tertiary analytics sequencing/assembly
- EDA/CAE — simulation and modeling
- AI/ML/Analytics — Deep learning training, batch and interactive IDE
- Media & Entertainment — Simulation/Rendering & SFX
- HPC- simulation/modeling
- FSI- analytics/decision support, low-latency persistent messaging
- In-Memory Databases — node consolidation, performance improvements, backup/recovery
- Cloud- KVM snapshots, Spot-instance for non-FT workloads

CXL Benefits

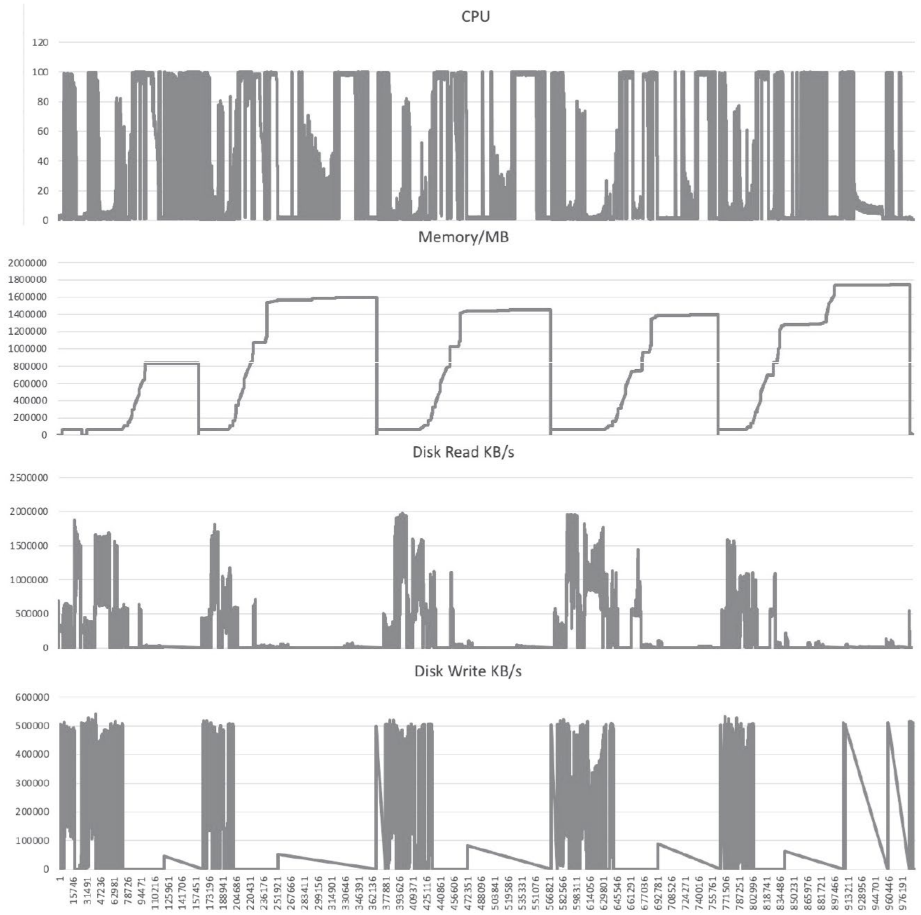
- Industry-wide Standard that will initially benefit Memory-Bound applications
  - Increase total amount and types of memory
  - Memory Bandwidth
  - HPC and AI/ML type workloads biggest benefactors

- Cloud
  - Ability to Improve overall utilization, efficiency of heterogeneous computing
- Composability (HW-level)
  - Memory becomes a First-Class Citizen, eventually consolidated into centralized pools
- Collaboration
  - Facilitate Data-sharing across applications and servers
- Concurrency
  - Enable concurrent heterogenous processes to run on same in-memory data sets to reduce wall-clock time
- Cache
  - Reduced movement of data to/from storage and memory as a result of larger pools/caches including persistence

Challenges to CXL Adoption

- Many HPC and AI/ML application/pipelines are “pets” with wide-ranging memory access patterns and behaviors.

HPC “Pet” Workload Compute Profile Example



MetaSpades Application alternates between compute, memory, and IO bound

https://www.biorxiv.org/content/10.1101/2022.04.20.488965v1.full.pdf

First Step in Adopting CXL: Achieving Application Transparency

Goal: Targeted HPC and AI/ML Applications Must run Better, Faster, Cheaper without Modification

Pre-CXL objective: Transparent, Automated Memory Tiering

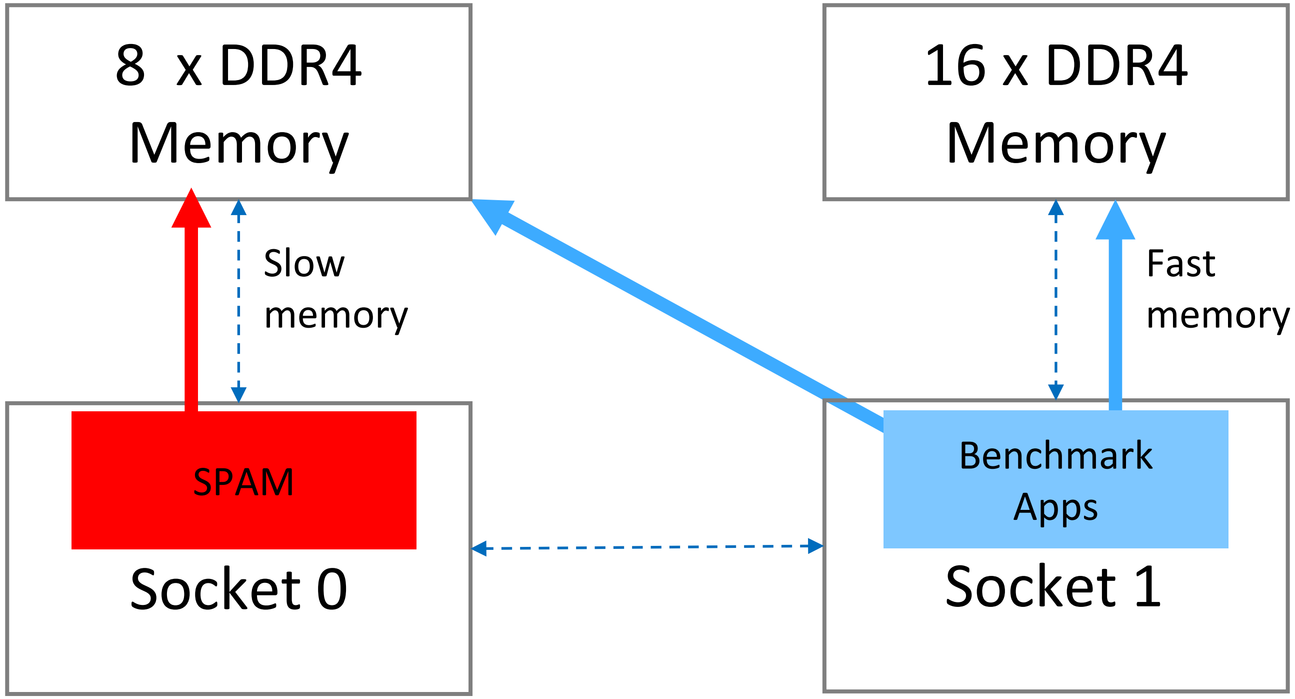
A). Memory Capacity: Memory tiering between DRAM and PMEM on same DDR bus

B) Memory Bandwidth/Latency: Tiering across local and remote NUMA DRAM.
CXL 1.1 memory expansion latency expected to equivalent to one NUMA hop

Validate impact of various pooling, profiling, policy, and placement approaches to memory tiering

Emulating CXL Two-Tiered Memory with DRAM

- DRAM 384GB DDR4 in total
  - Slow tier on socket 0: DDR4 2666 MHz 16GB x 8
    - Use less DIMMs and running spam app to further consume memory bandwidth
  - Fast tier on socket 1: DDR4 2666 MHz 16GB x 16
    - Benchmarking app runs on socket 1, has the fastest access to its local memory



	Fast Tier	Slow Tier
Access Latency	108ns	335ns

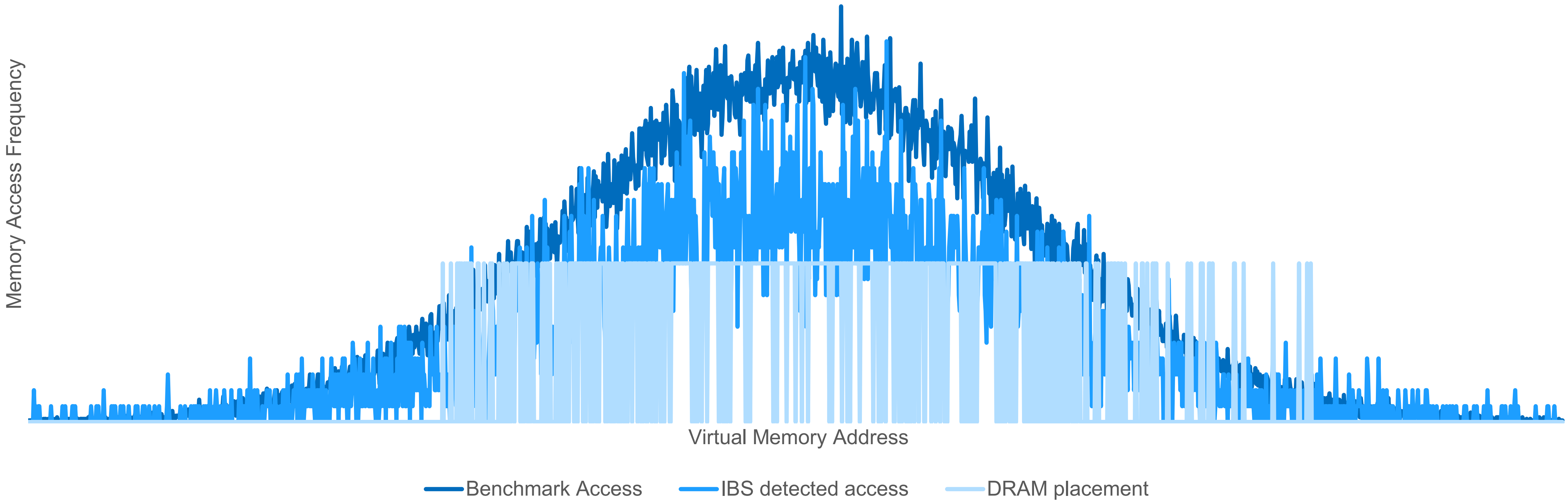
*Measured with memory latency checker

9

MEMVERGE CONFIDENTIAL

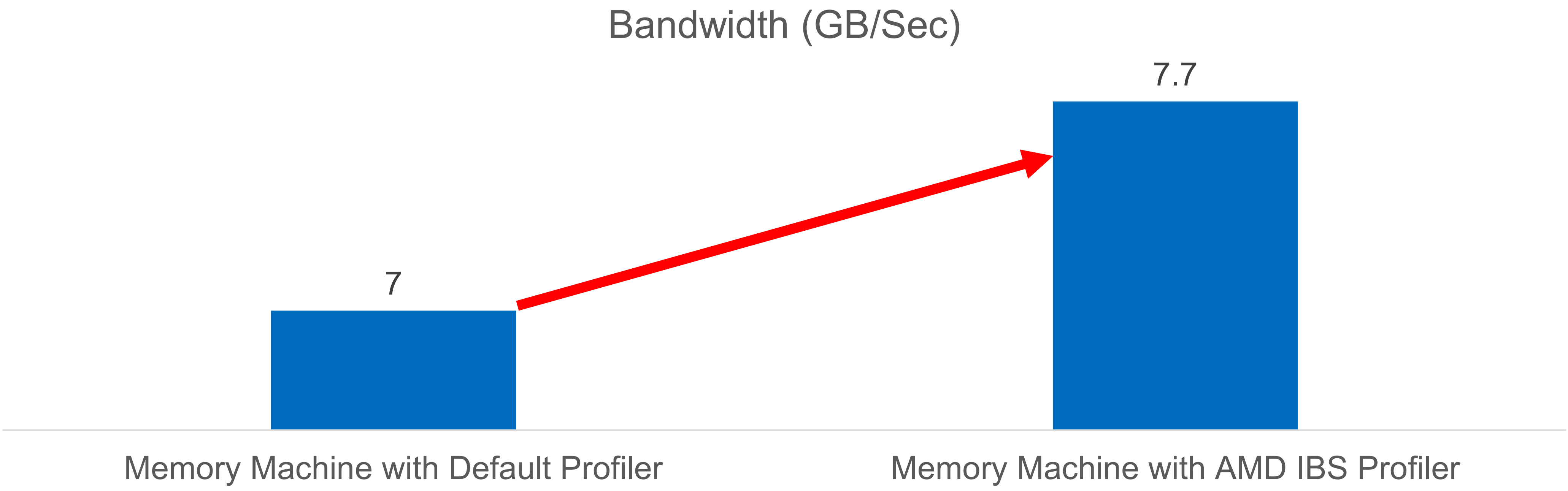
Memory Profiling Supplemented by HW-based IBS

- MemVerge synthetic memory load generator simulates memory access that follows a Gaussian distribution (i.e., data locality)
- Memory Machine places faster DRAM to back the hottest virtual memory addresses detected by IBS-assisted memory profiler



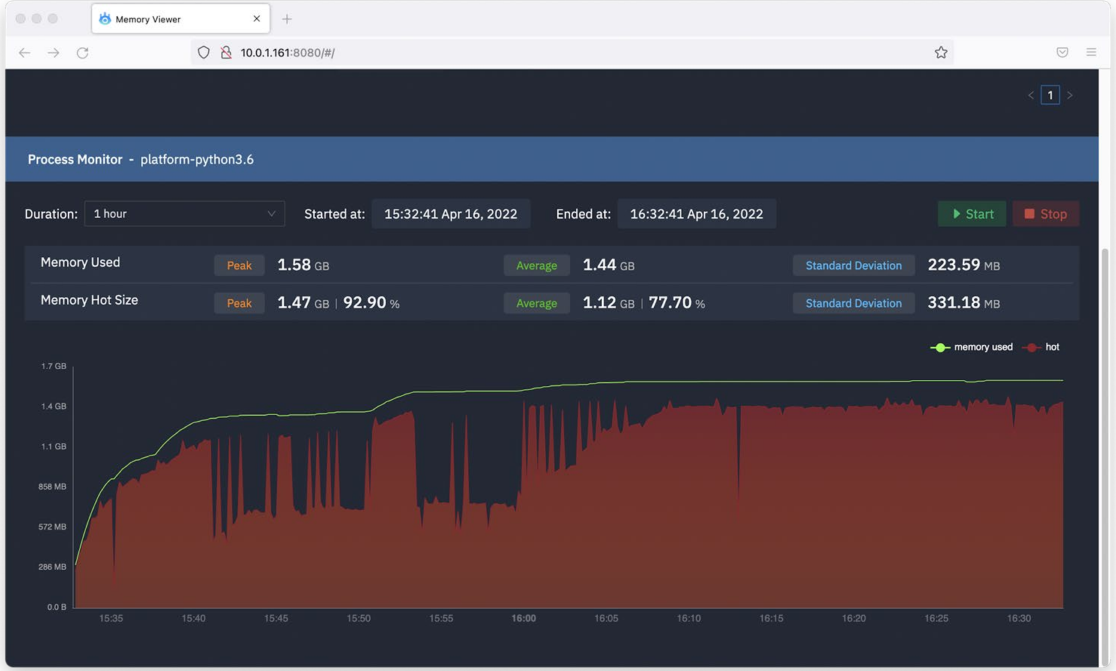
Synthetic Benchmark Performance

- MemVerge synthetic memory load generator simulates memory access that follows a Gaussian distribution (i.e., data locality)
  - Benchmark used 2GB fast memory + 6GB slow memory
  - Measured memory access bandwidth



Memory Viewer for Looking at Application-Level Memory Heatmap

Not suited for Memory Tiering

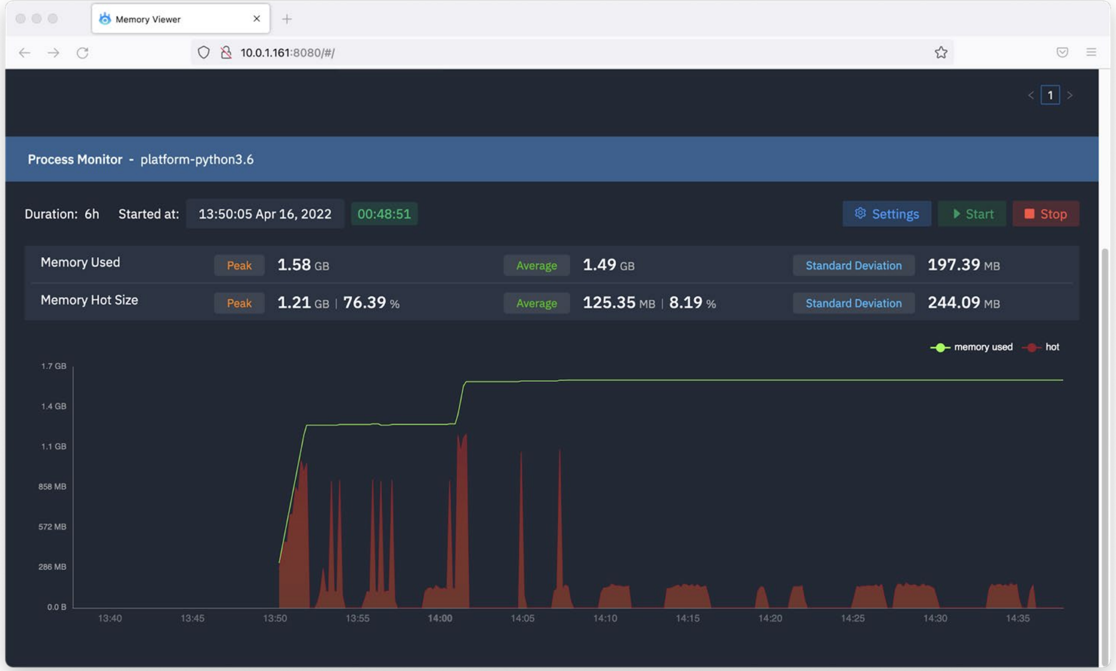


Process Monitor - platform-python3.6

Duration: 1 hour Started at: 15:32:41 Apr 16, 2022 Ended at: 16:32:41 Apr 16, 2022

Memory Used	Peak	1.58 GB	Average	1.44 GB	Standard Deviation	223.59 MB
Memory Hot Size	Peak	1.47 GB | 92.90 %	Average	1.12 GB | 77.70 %	Standard Deviation	331.18 MB

Ideal for Tiering

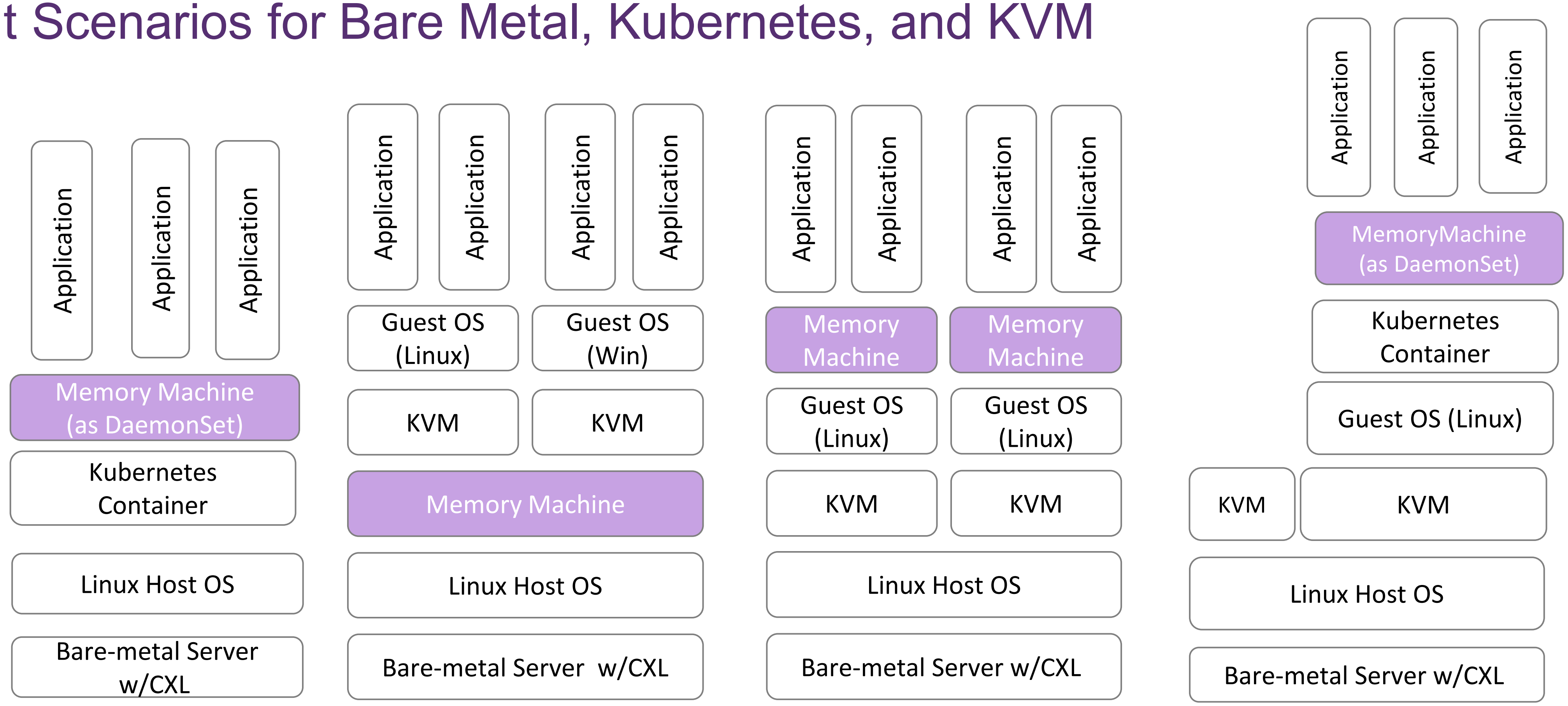


Process Monitor - platform-python3.6

Duration: 6h Started at: 13:50:05 Apr 16, 2022 00:48:51

Memory Used	Peak	1.58 GB	Average	1.49 GB	Standard Deviation	197.39 MB
Memory Hot Size	Peak	1.21 GB | 76.39 %	Average	125.35 MB | 8.19 %	Standard Deviation	244.09 MB

Deployment Scenarios for Bare Metal, Kubernetes, and KVM



Application | Application | Application

Memory Machine (as DaemonSet)

Kubernetes Container

Linux Host OS

Bare-metal Server w/CXL

Application | Application | Application | Application

Guest OS (Linux) | Guest OS (Win)

KVM | KVM

Memory Machine

Linux Host OS

Bare-metal Server w/CXL

Application | Application | Application | Application

Memory Machine | Memory Machine

Guest OS (Linux) | Guest OS (Linux)

KVM | KVM

Linux Host OS

Bare-metal Server w/CXL

Application | Application | Application

MemoryMachine (as DaemonSet)

Kubernetes Container

Guest OS (Linux)

KVM | KVM

Linux Host OS

Bare-metal Server w/CXL

Flexibility of Memory Machine deployment allows operators to decide level of granularity of memory services and/or target specific applications for optimization

Summary

- CXL will help revolutionize datacenter architecture starting with memory
  - Memory as first-class citizen – more bandwidth, varying latencies, varieties, and degrees of composability

- CXL will help accelerate many HPC & AI-ML application workflows – faster time to insight

- Software-defined Memory is key to provisioning and managing various CXL memory pools that optimize application/workflow pipeline behavior.

- Partner with us in Shaping the Future of Big Memory!

PERSISTENT MEMORY
+ SUMMIT 2022
COMPUTATIONAL STORAGE

VIRTUAL EVENT • MAY 24-25, 2022

Thank You