HPC USER FORUM

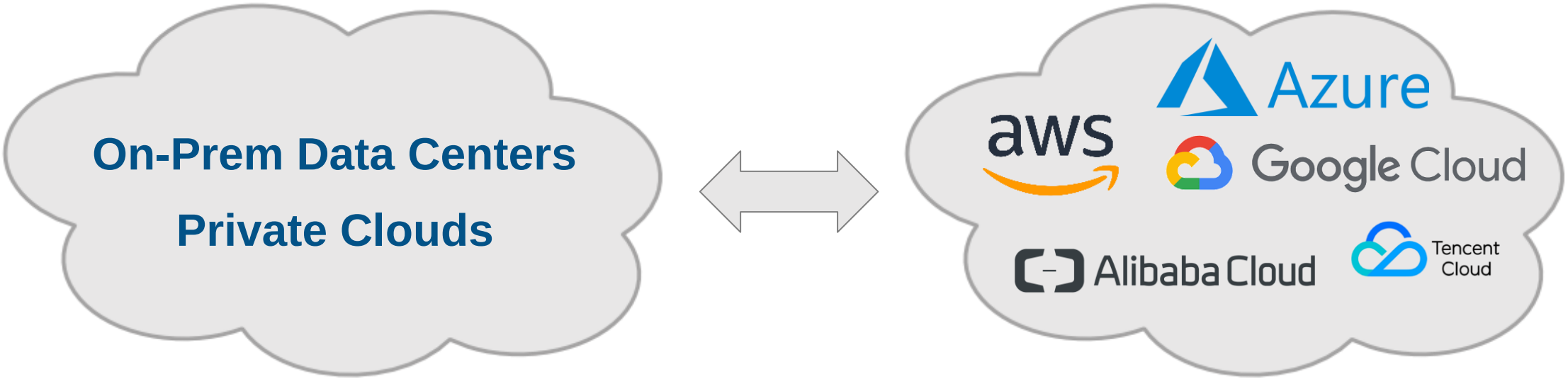
Big Memory Software for HPC

Dr. Charles Fan
CEO, MemVerge

MemVerge

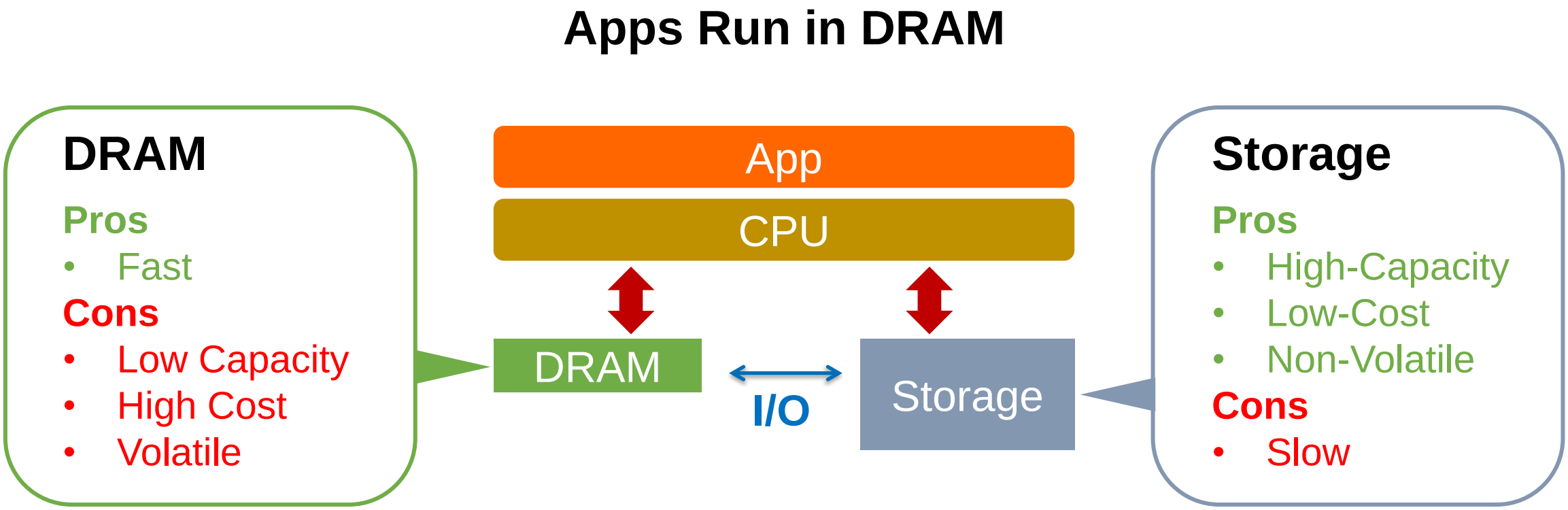
Future of Infrastructure

- **Multi-Cloud**
- **Memory-Centric**
- **Software-Composed**



MemVerge

Today's Computer



MemVerge

Data Has Become Big & Fast

Demanding Memory-Centric Infrastructure


Big Data Analytics

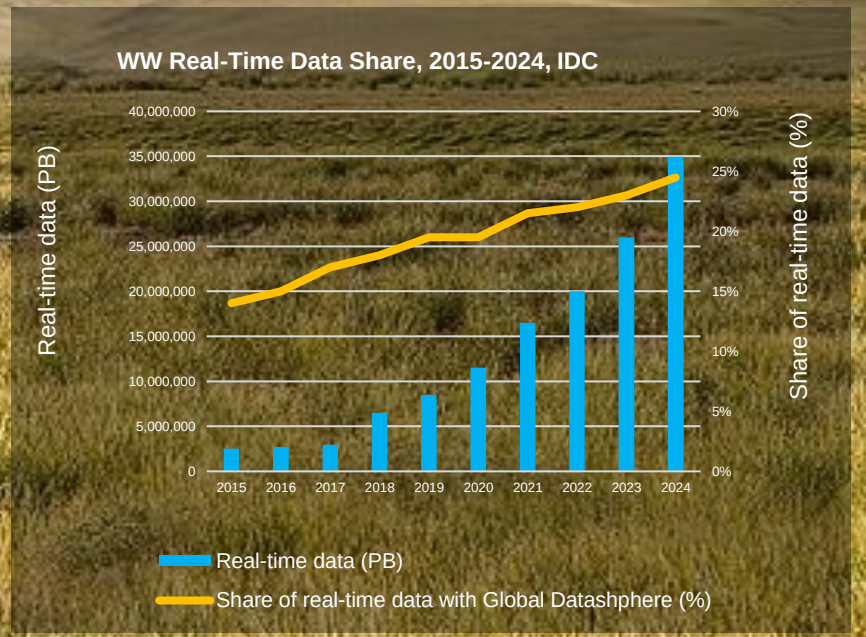

AI/ML Inference


Capital Markets


3D Animation


Virtual Servers


Oil & Gas



MemVerge

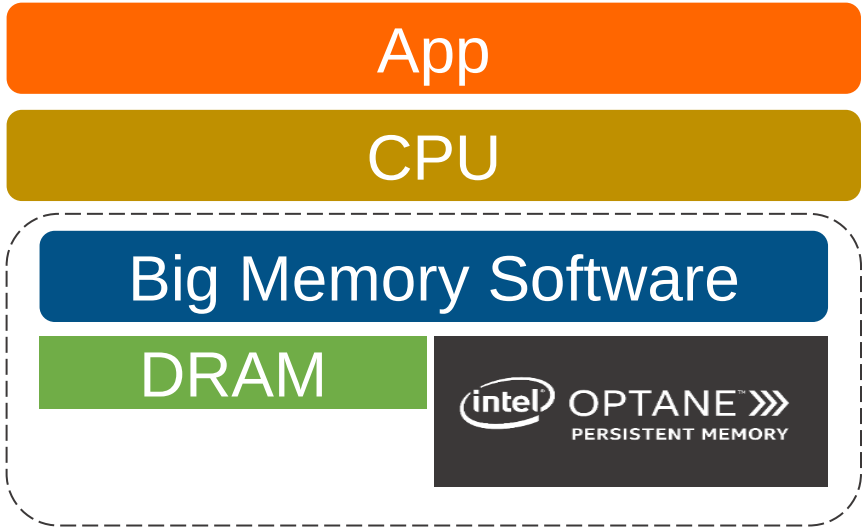
The Rise of Big Memory Computing

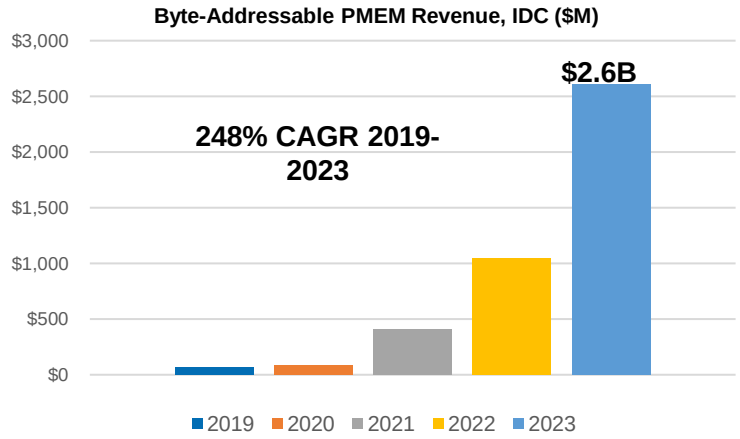
Apps Run in DRAM *and* PMEM

DRAM + PMEM

Pros

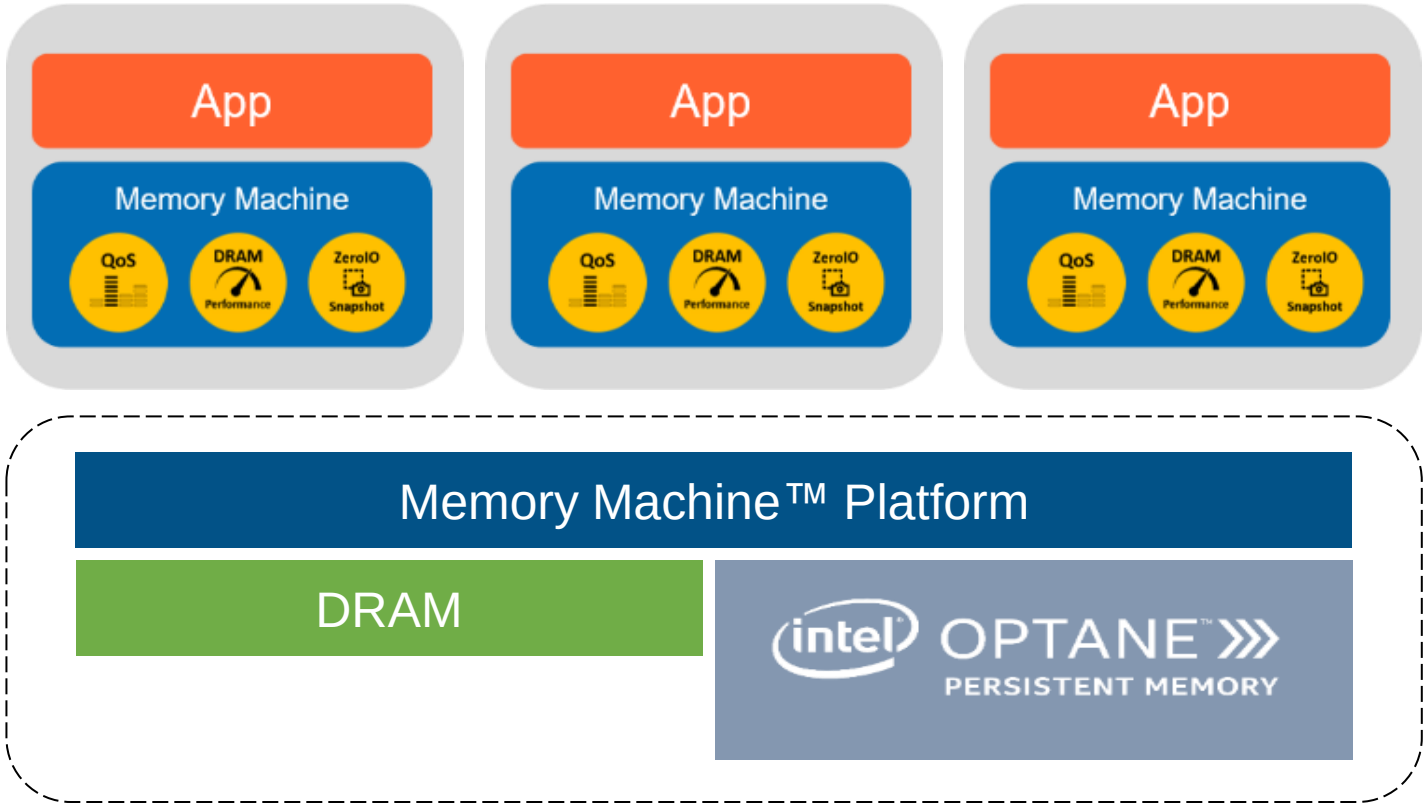
- Fast
- High-Capacity
- Low-Cost
- Non-Volatile





MemVerge

Memory Machine™: World's First Big Memory Software



Bigger Memory at Lower Cost without Performance Compromise

- Up to 9TB memory/2-way server
- 30-50% Memory Cost Savings
- DRAM-Performance

Persistence On-demand

- ZeroIO™ In-Memory Snapshot
- Fast Crash Recovery
- Thin-Clones

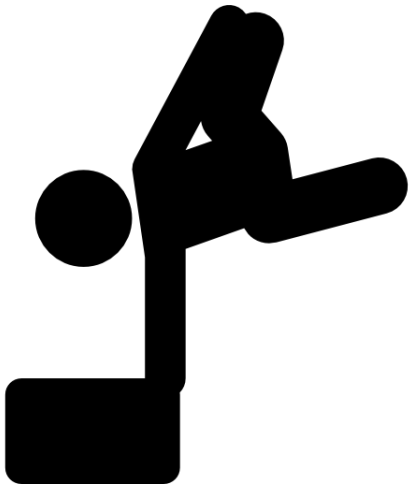
No Application Change!

Big Memory Software Impacts HPC

Performance

Availability

Agility



MemVerge

Inference with Large Model and Feature Embeddings

- Motivation
  - Large model and embedding table size
    - Model size to GB level, embedding table size to TB level
    - Multiple models on single server
  - Online inference service: real time and low latency
    - Return results in tens of ms
- Ideal solution
  - Put models and embedding tables into DRAM
- Limitations
  - High TCO
  - Limited DRAM space
  - Volatile

Inference on Memory Machine

- Our solution
  - Models and embedding tables in DRAM + PMEM

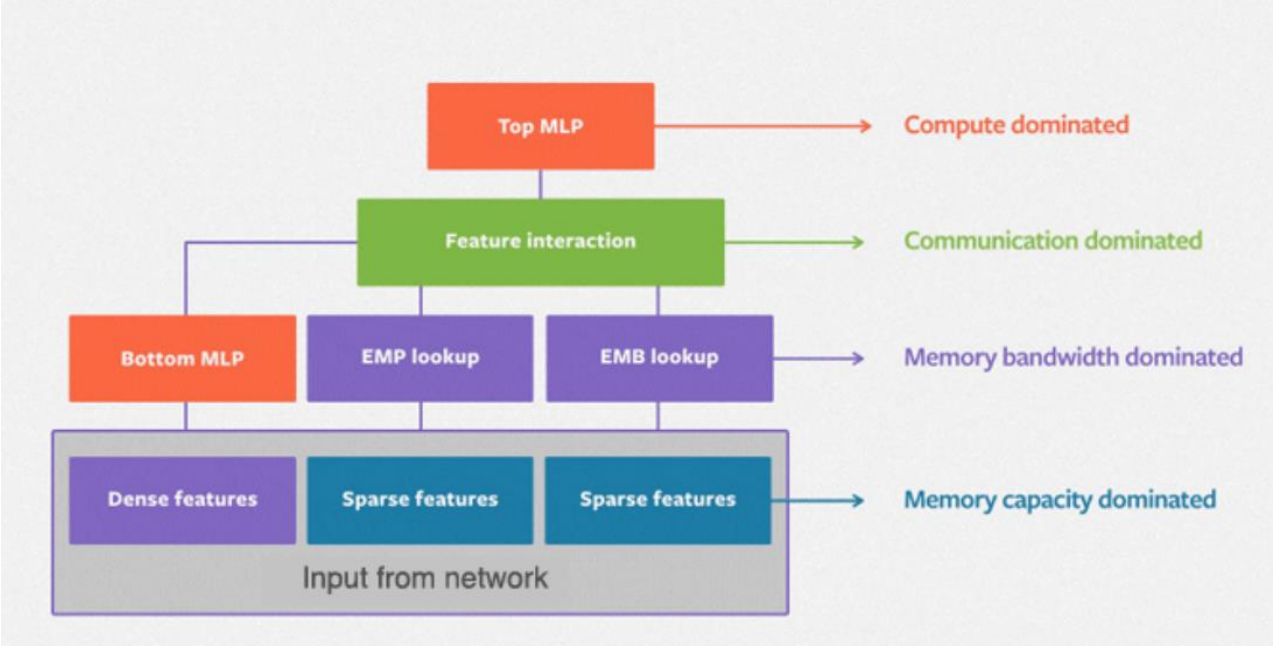
- Benefit
  - Big memory can include all embedding tables on one server
  - Similar read performance as DRAM, very suitable for read-heavy scenario such as online inference
  - Data persistence on PMEM

Example 1: Facebook's DLRM

- **Deep learning recommendation model for personalization and recommendation systems**
  - Consists of dense and sparse features
  - Dense feature: a vector of floating-point values
  - Sparse feature: a list of sparse indices into embedding tables
  - Open source:

https://github.com/facebookresearch/dlrm

PyTorch



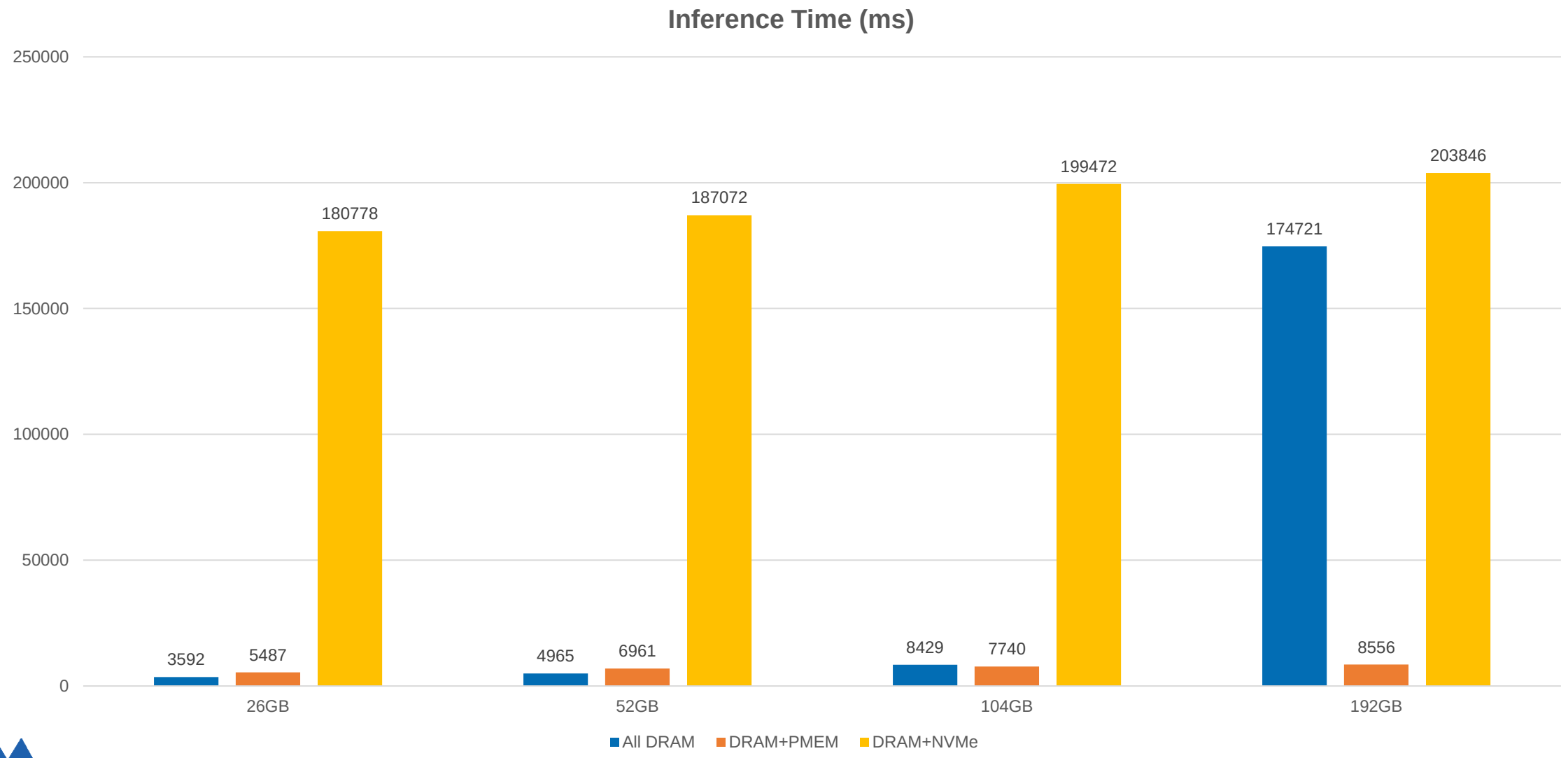
M. Naumov, et al. *Deep Learning Recommendation Model for Personalization and Recommendation Systems*, 2019 https://arxiv.org/abs/1906.00091

MemVerge

Evaluation Setup

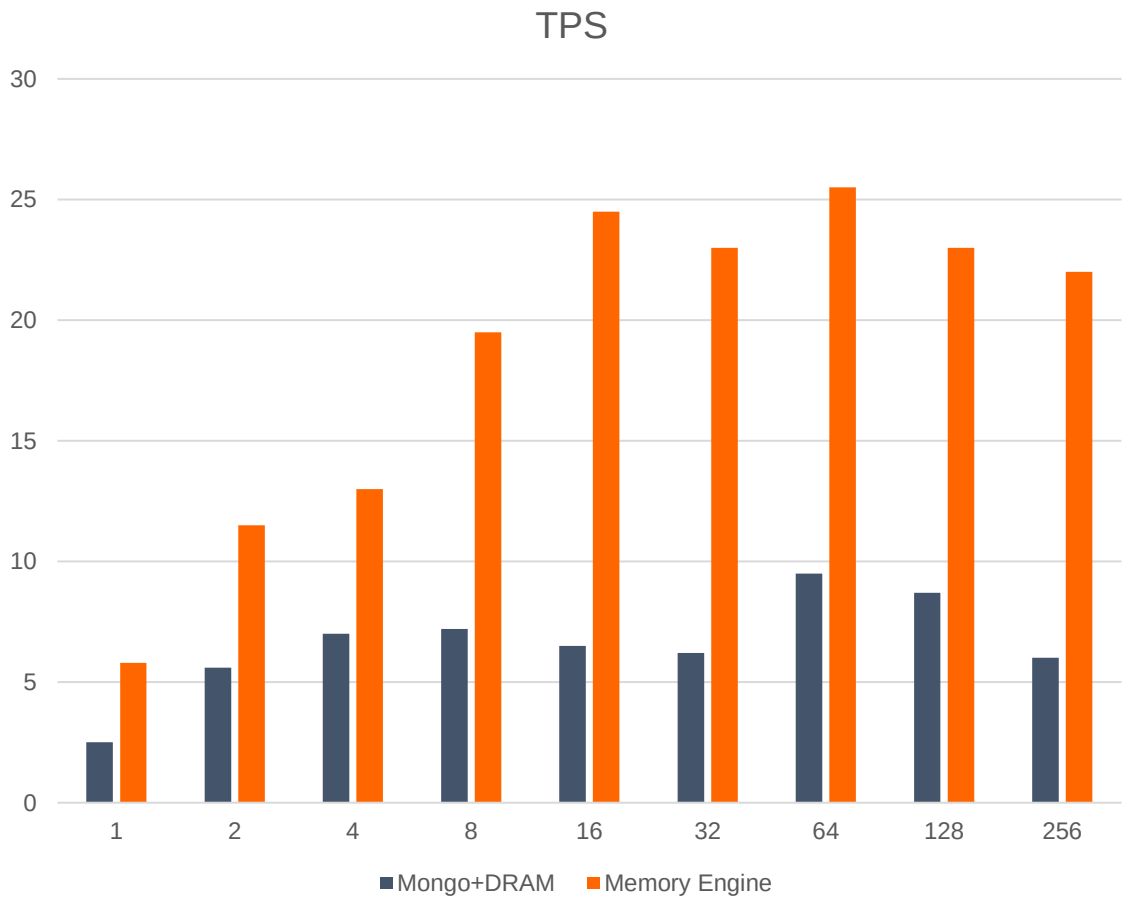
- Hardware:
  - Intel(R) Xeon(R) Platinum 8280 CPU @ 2.70GHz (112 cores)
  - 192 GB DRAM, 1.5TB PMEM, 400GB NVMe SSD
- Software
  - RHEL 8.2
  - Memory Machine v1.0
  - Latest DRLM framework
- Testing cases: model + embedding
  - In memory data size 26G/52G/104G/192G
  - Features: 100 sparse features (100 embedding tables, embedding vector dimension is 64), 512 dense features
  - Measuring inference time for 20480 records in one batch (Criteo Dataset)

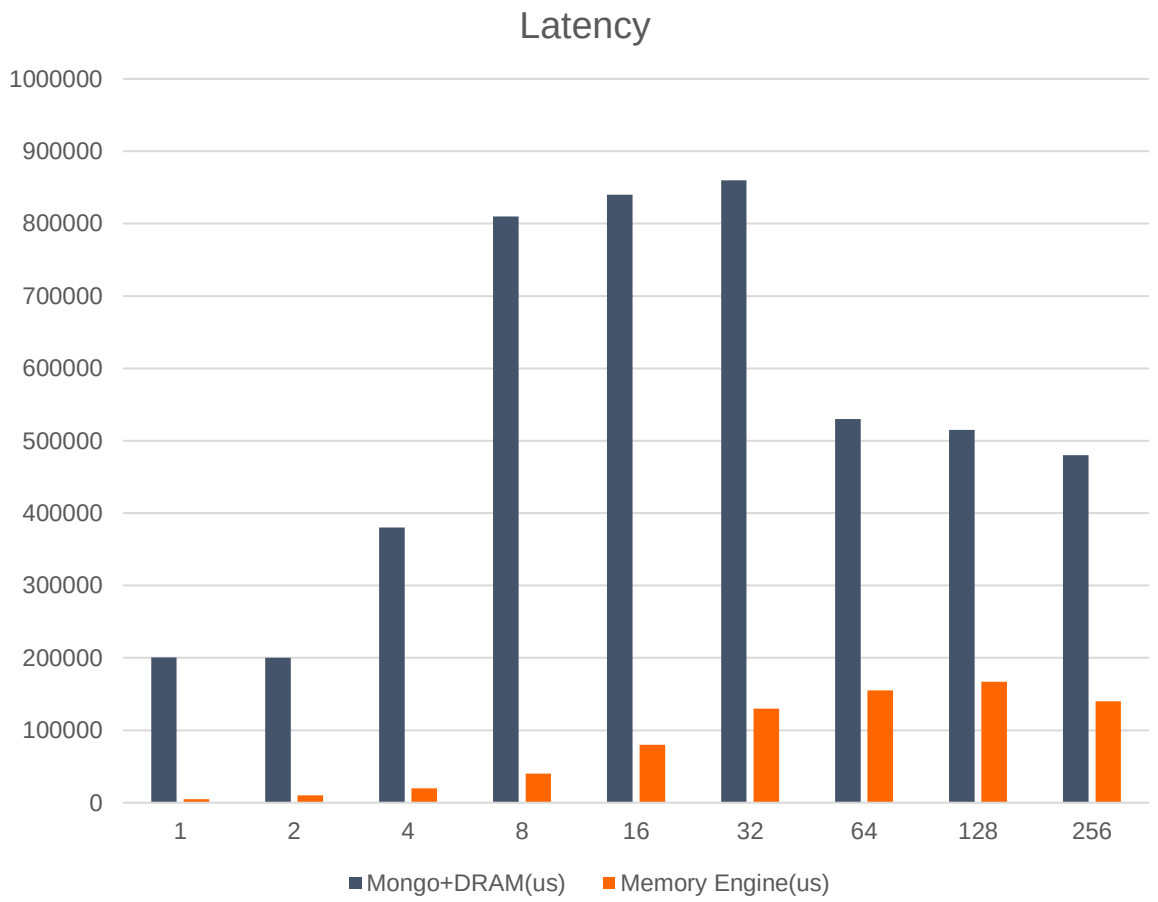
Example 1: DLRM Inference Performance



MemVerge

Example 2: Image Recognition Performance





MemVerge

Persistent Memory for Instant Model Rollback/Recovery

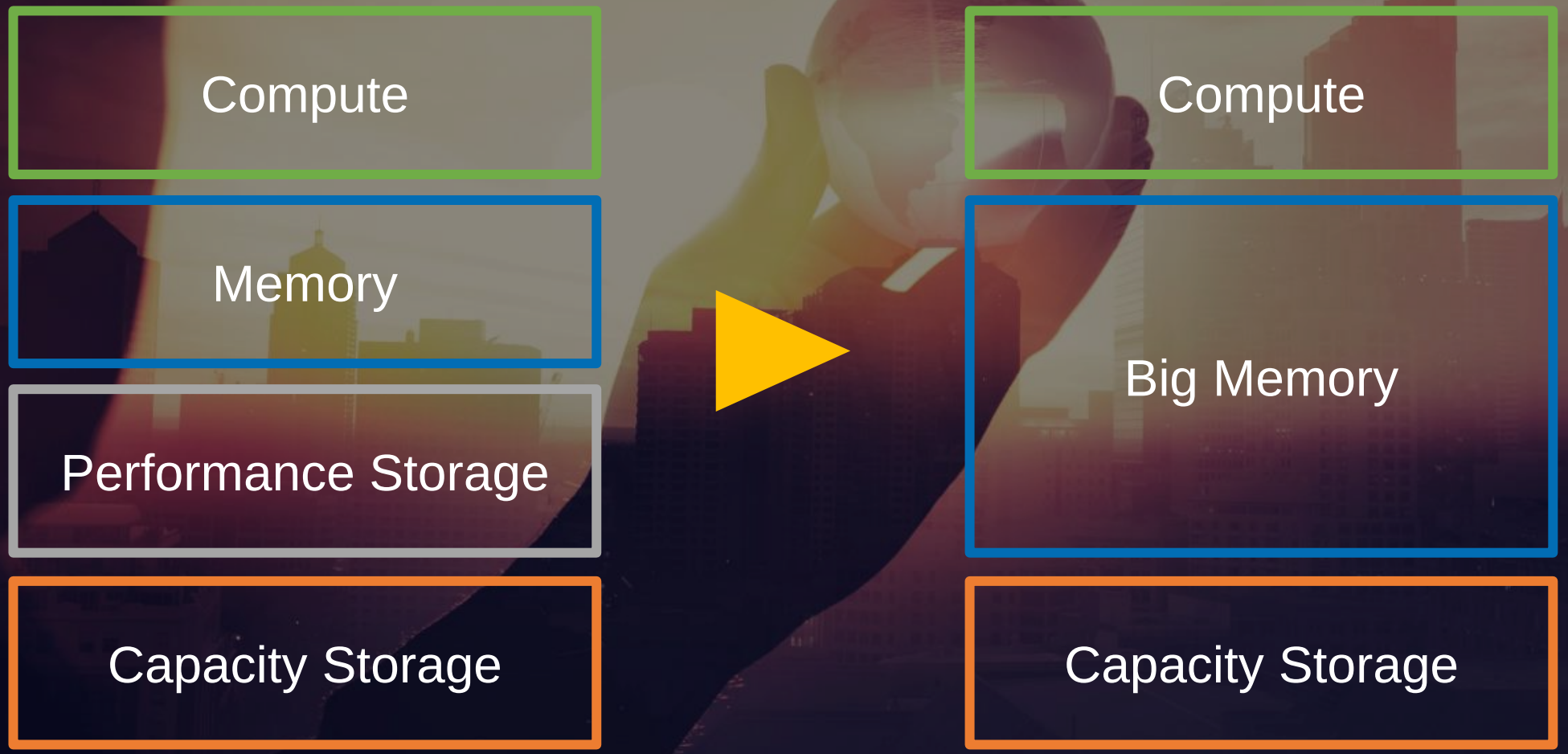
- How to improve the fault tolerance of new model publishing?
  - Pushing new model into production is risky
  - If failed, revert to last workable version ASAP
  - Rollback/Model reloading takes time (for large models) due to slow I/O
- Leveraging PMEM's persistence
  - Take a snapshot of the model serving application
  - Restore a snapshot without reloading from disk or remote storage
  - Snapshot can be published to many serving nodes via memory-to-memory snapshot replication
- Solution
  - Instantaneous snapshot without interrupting online inference
  - Instantaneous rollback without loading and publishing time
  - Snapshot, rollback, and recovery are within 1 second

Summary

- Memory Machine provides
  - Larger and cheaper heterogenous memory for faster inference
  - Persistent memory for instant model snapshot and recovery
  - No application change is needed

- Human reasons everything fully from memory
  - So will machine learning in the era of Big Memory

Big Memory Software Will Be a $10B+ Market



MemVerge