SOLUTION BRIEF

Greater Memory Capacity & Plug-Compatible Access to Persistent Memory for ML Applications

MemVerge

PROBLEM

Data Sets Are Greater Than Memory

Swapping data to/from storage defeats the benefits of memory

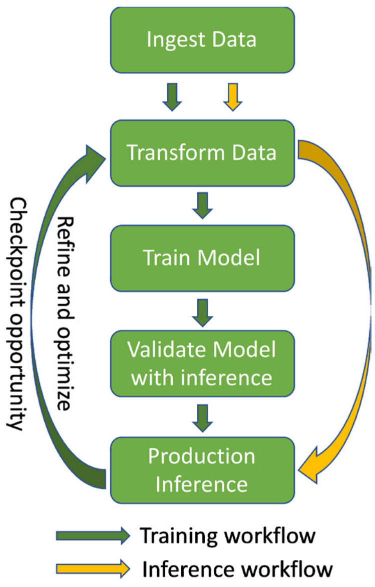
Machine Learning (ML) enables computers to identify complex patterns at human or even super-human levels of performance. ML systems that serve real-time applications are typically dependent on large data sets deployed, as much as possible, in-memory for fast execution.

Analytical models are constructed which learn from data, identifying patterns and making decisions based on applying those patterns. State-of-the-art ML models are neural networks often consisting of many deeply nested layers. Training neural networks is a massively iterative process, repeatedly applying data to a model and altering its parameters based on the data itself or on external feedback.

Inference from a pretrained model may require large databases that must reside in system memory to achieve sufficient transactional performance. It is necessary and common to run multiple models per server due to economics and resource constraints. While a single model may well fit in available DRAM, a combination will not, forcing the system to start swapping to disk. Computer operating systems perform best with all code and data in DRAM. When resources become tight, less-used data in memory are written to disk to make more space, only to be swapped back later. This back-and-forth has severe impacts, drastically reducing the performance of the entire computer - and often connected, dependent systems - until the shortage clears.

If the amount of data is greater than the amount of available memory, a solution is needed to cost-effectively and efficiently increase memory capacity.

Machine Learning Workflow



SOLUTION

Big Memory

Defining Big Memory

Big Memory is a class of computing where the new normal is mission-critical applications and data living in byte-addressable, and much lower cost, persistent memory.

It has all the ingredients needed to handle the growth of IMDB blast zones by accelerating crash recovery. Big Memory can scale-out massively in a cluster and is protected by a new class of memory data services that provide snapshots, replication and lightning fast recovery.

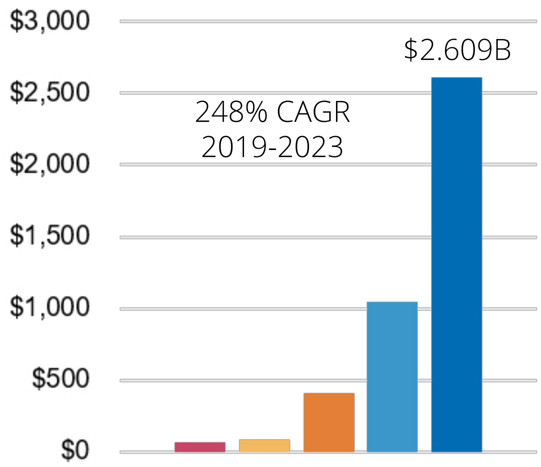
The Foundation is Intel Optane DC Persistent Memory

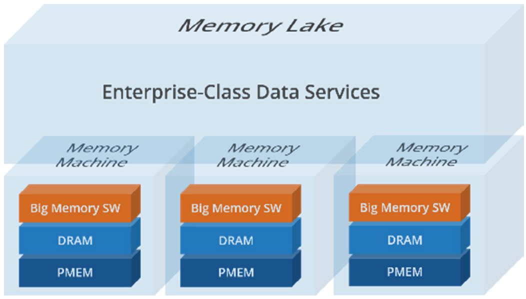
The Big Memory market is only possible if lower cost persistent memory is pervasive. To that end, IDC forecasts revenue for persistent memory to grow at an explosive compound annual growth rate of 248% from 2019 to 2023.

MemVerge Software is the Virtualization Layer

Wide deployment in business-critical tier-1 applications is only possible if a virtualization layer emerges to deliver HPC-class low latency and enterprise-class data protection. To that end, MemVerge pioneered Memory Machine™ software.

Persistent Memory Revenue Forecast 2019 – 2023 - IDC





Download IDC Presentation Defining Big Memory

IDC

Digital Transformation Driving New "Big Memory" Requirements
Eric Burgener, Research Vice President
Infrastructure Systems, Platforms and Technologies Group
May 2020

BIG MEMORY SOLUTION FOR

DLRM

Personalization and recommendation models like the Deep Learning Recommendation Model are a very common use for neural network-based ML systems today.

This class of systems combines data on category groupings - often influenced by a record of past user behaviors - with predictive insights. The latter uses statistical models to classify or predict the probability of an outcome based on provided data.

Facebook AI Research released DLRM , an open source ML model, to provide a platform for external development. DLRM is implemented using the well-known PyTorch framework that originated at Facebook.

DLRM consists of dense and sparse features. Dense features are represented as a vector of values while sparse features are a list of indices into “embedding tables” containing values. These tables, as concentrated pre-calculated data, can be placed in persistent memory, reserving DRAM for the model and dense feature data. In a production environment, embedding tables could be terabytes in size.

The presence of terabyte datasets such as those used for the Kaggle Display Advertising Challenge, make DLRM a great candidate for a Big Memory solution.

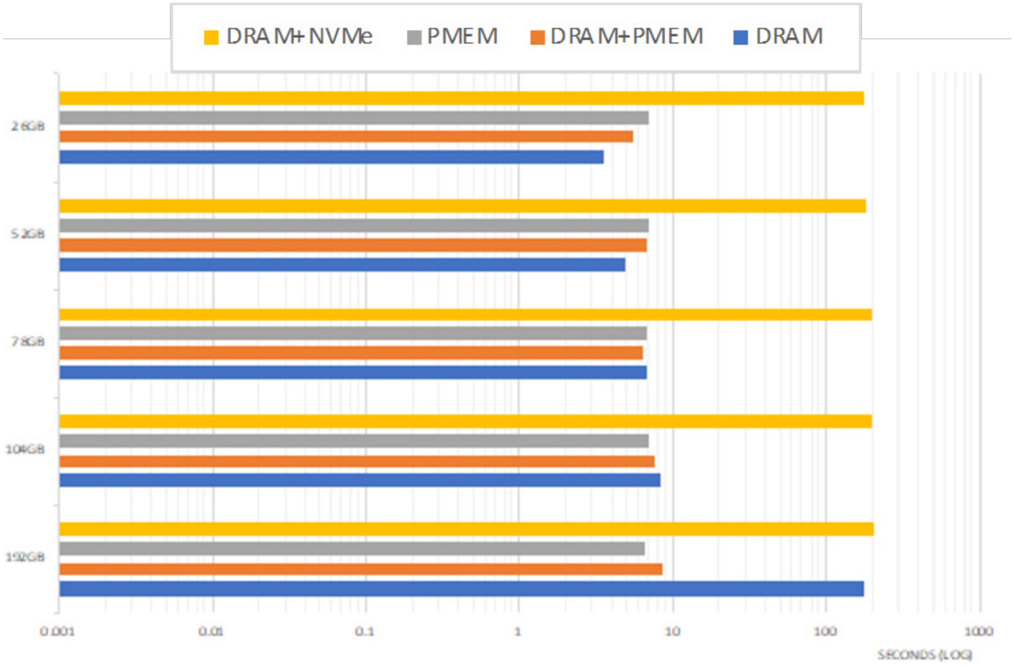
The goal of this challenge is to benchmark the most accurate ML algorithms for CTR estimation. All winning models will be released under an open source license. As a participant, you are given a chance to access the traffic logs from Criteo that include various undisclosed features along with the click labels.

CTR Prediction Contest
criteo

Testing Big Memory with DLRM

Four MemVerge test configurations were run on a single server with 192GB DRAM and 1.5TB Intel Optane DC PMEM using MVMM to direct memory allocation:

1. Model and embedding tables all in DRAM
2. Model in DRAM, embedding tables on KV store on NVMe SSD
3. Model and embedding tables all in PMEM
4. Model in DRAM, embedding tables in PMEM



These results show the power of Memory Machine software allowing DLRM to select its memory from DRAM or PMEM. By reducing or eliminating the need to swap, application and system performance remain high even if PMEM is theoretically slower than DRAM.

Summary results:

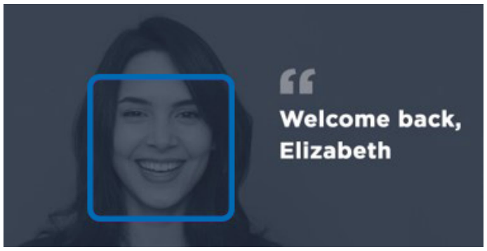
- Configuration 1 is the fastest if the model and table fit within available DRAM
  - When DRAM is exceeded, performance slows by 50x as system begins to swap
- Configuration 2 is drastically slower (~50x), showing access penalty of even the fastest disk
  - Performance remains close to constant at this slow value as DRAM usage increases
- Configuration 3 is somewhat slower than C1 but maintains constant performance
  - As DRAM usage increases, performance soon exceeds that of DRAM
- Configuration 4 is somewhat slower than C1 but faster than C3 until DRAM is depleted
  - As DRAM usage increases, performance begins to slow but system remains performant

With resources fully available to run DLRM in DRAM, performance is high. As memory resources are depleted, performance rapidly begins to fall off. Once ~90% of system memory is consumed, the operating system enters the swapping regime and performance drops catastrophically (for a transactional system). It is also demonstrated that disk storage, while capacious, is also too slow for a system of this time.

BIG MEMORY SOLUTION FOR

Image Recognition

An increasingly important AI workflow is quick recognition of images from real-time camera inputs.



This is an excellent example of massive inference from extremely large databases needing to be accomplished in the shortest possible time. Many identity attempts are in process at any given time, requiring the most efficient use of servers to make this practical in the real world.

MemVerge testing involved identifying 5000 images from four database configurations (enumerated in results, below). Each test was performed at concurrencies from 1 to 256 simultaneous users.

The baseline configuration consists of DRAM + a MongoDB instance on SSD. This is compared with (a) allocating all memory from PMEM and (b) with an additional optimization of selectively allocating from DRAM for smaller memory areas.

Summary results:

- Single database containing 100 million images
  - Computational performance in PMEM is 60-90% of DRAM + MongoDB, as PMEM is slower than DRAM especially when the working set fits entirely into DRAM
- 1000 databases, each containing 1 million images
  - PMEM can maintain much higher transaction rates than DRAM + MongoDB
  - PMEM latency is lower even the fastest SSDs especially at higher concurrency levels
- Single database containing 1 billion images
  - DRAM is superior for candidate generation algorithms that fit, but less so with load
  - Even so, PMEM is greatly superior for data access than the SSD-based MongoDB
  - Selectively allocating from DRAM and PMEM, 10x DRAM + MongoDB TPS is achieved
- Single database containing 2 billion images
  - DRAM + MongoDB cannot run or complete successfully
  - DRAM + PMEM returns search results within 500ms

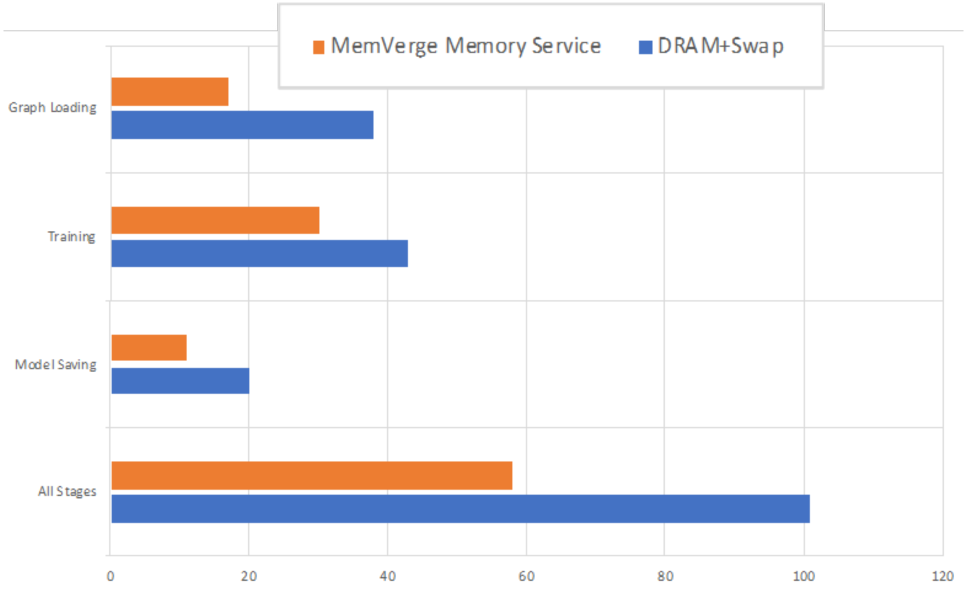
BIG MEMORY SOLUTION FOR

Stanford GraphSAGE

GraphSAGE uses large amounts of memory when its graph is large.

Graphs are well-known mathematical structures describing a set of somewhat related objects, commonly expressed as nodes connected by edges. GraphSAGE is a current ML framework, originated at Stanford University , intended for inductive learning on large graphs (>100,000 nodes), especially for those rich in node attributes. GraphSAGE is implemented using the well-known TensorFlow framework that originated at Google.

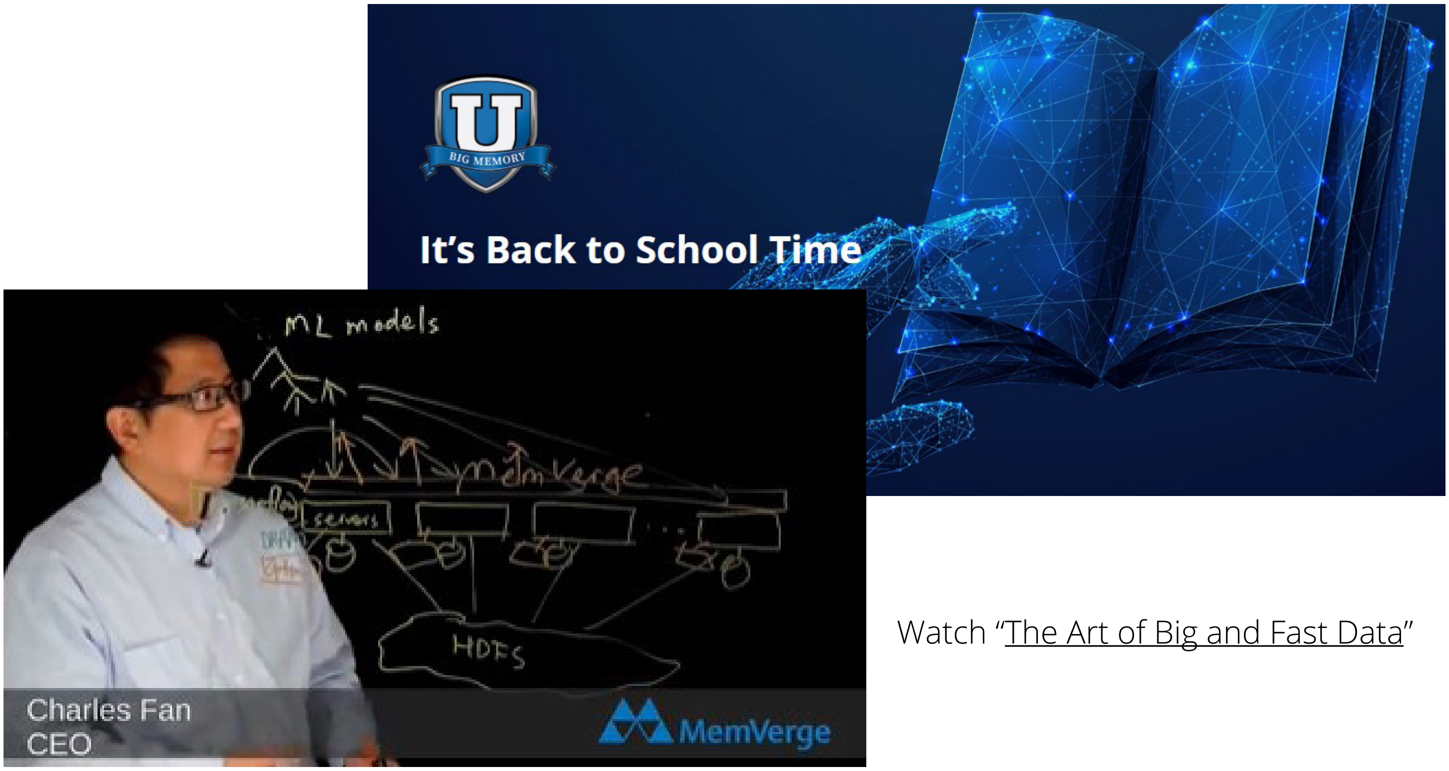
MemVerge selected GraphSAGE as representative of an ML production system using large amounts of memory when its graph is large. The baseline measurement uses system DRAM first, swapping to NVMe SSD as DRAM is depleted. This is compared with using MVMM to allocate memory from PMEM.



Summary results:

- Baseline configuration slows drastically once Linux begins to swap
- By keeping large data structures in PMEM, application can maintain nearly full DRAM speed

These results demonstrate the impact of MVMM-managed PMEM. By shifting its large, complex data structures into PMEM, GraphSAGE reduces its DRAM footprint to where the application can run at full speed without being affected by common memory management issues.



Watch "The Art of Big and Fast Data"

Help us shape the future of Big Memory

- Join the Big Memory Community to receive news, tech tips and early access to new software releases.
- Attend Big Memory University and earn a certificate.
- Deploy a Memory Machine PoC to see the power of Big Memory for yourself.

Send me info