Characterizing Data Ingest for Deep Learning Recommendation Model Training

Sundararajan (Sundar) Sankaranarayanan, Sayali Shirode

FMS 2024, 08/08/2024

Content

- Key Takeaways
- Motivation – DLRM Scale & Significance
- Storage Trace Analytics – Results
- Q & A

Key Takeaways

Takeaways	❖ Rapid increase in training dataset size and the variety of models to train will continue to put pressure on storage density, SSD capacities and throughput requirements
	❖ With increasing demand for energy from datacenter/edge devices, there will be continued pressure to make storage energy-efficient
	❖ DLRM is a key production model, and requires high-capacity and throughput from SSDs for training purposes
	❖ We examine storage traces of ▪ DLRM Data Preprocessing (under discussions to be part of MLPerf Storage suite) ▪ DLRM Training (part of MLPerf Training suite)
	❖ Storage trace analysis of AI workloads show evidences of ▪ sequential read (write) accesses ▪ large payloads for reads and write commands
Call to Action	❖ ML Commons may want to consider creating a suite that places data preprocessing inline with DLRM training

Deep Learning Recommendation Models – Scale & Significance

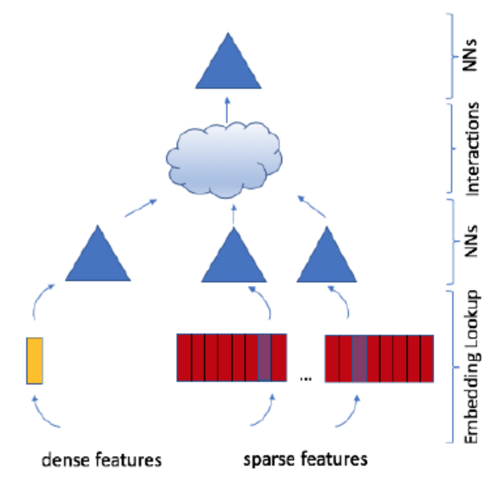
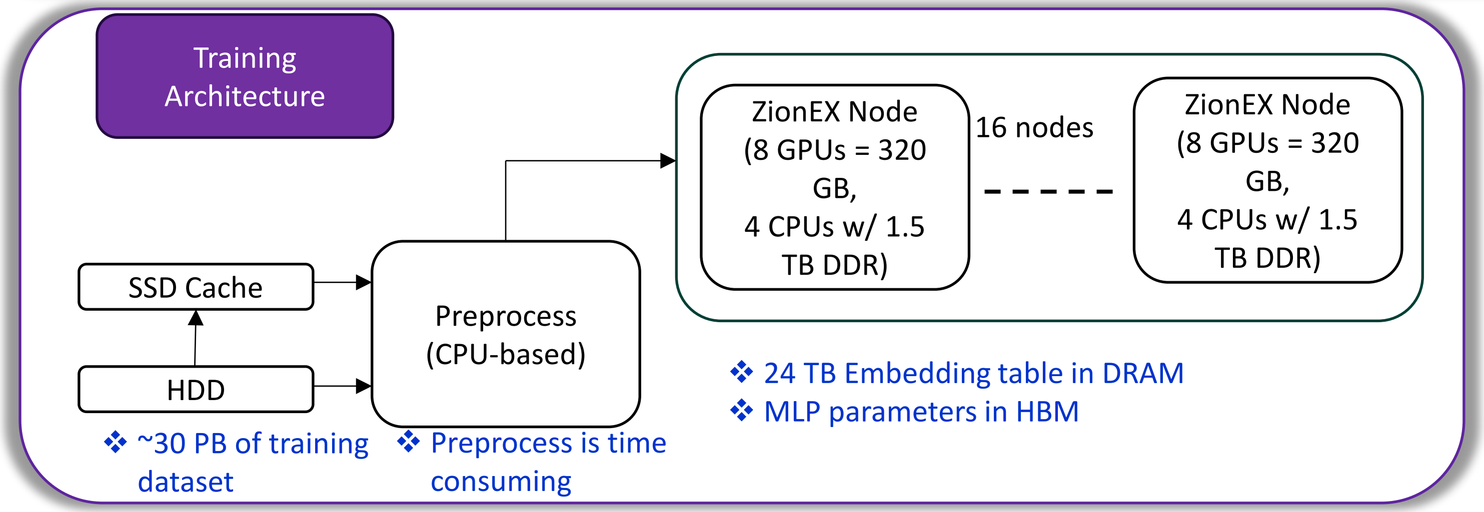


Figure 1: A deep learning recommendation model

- ❖ Recommendation models are backbone for Meta, Netflix, Google etc.
- ❖ Model parameters: a) MLP b) embedding tables One of Meta's models has 12 trillion parameters
- ❖ Size of embedding tables is a key bottleneck and often tiered in host memory (DRAM) Meta's embedding table is 96 TB → 24 TB compressed
- ❖ Models are trained on Petabytes of data in datacenters (Source: Meta)
- ❖ Raw training data (or click logs) must be pre-processed prior to DLRM training – time intensive and GPU may be used for acceleration 1 TB dataset takes 5000 seconds on CPU to be pre-processed
- ❖ Multiple DLRM models are maintained, and new models are continuously trained and developed



Takeaways

- ❖ Preprocessing is inline with DLRM training
- ❖ Storage and preprocessing can sometime consume more power than training
- ❖ SSD Shift layer over the HDD Tectonic layer is introduced to deal with increase in ingestion bandwidth
- ❖ Meta had explored clever data placement algorithms, data filtering algorithms, and preprocessing with GPUs to meet increasing ingestion demand

DLRM – Deep learning recommendation model

Storage Trace Analysis – Summary (1/2)

DLRM Model	❖ Large DLRM model w/ 13 numerical, 26 categorical, and 1 true label features ❖ Large DLRM model consumes ~132 GB of VRAM on GPU HBM ➔ ~4 A100 GPU HBM capacity is required
Dataset	❖ 1 TB of raw data (Criteo Click 1 TB dataset)
Preprocessing	❖ Raw data → converted to parquet format → categories represented with hash values is converted to contiguous integer representation → missing numerical feature values are zeroed → numerical feature values are normalized → 370 GB of processed data in binary format
System & Tracing	❖ AMD EPYC 7742 128-Core Processor (2x64) ❖ NVIDIA A100 – 8x 40 GB ❖ NVMe Tracing using libpf
References	❖ https://github.com/NVIDIA/DeepLearningExamples/blob/master/PyTorch/Recommendation/DLRM/README.md#model-overview

Storage Trace Analysis	DLRM Preprocessing w/ GPU	DLRM Preprocessing w/ CPU	DLRM Training on GPU
Experimental Setup (0)	Preprocessed with 8 GPUs (DGX A100)	Preprocessed with 2 64-core CPUs	Trained with 8 GPUs (DGX A100) – batch size = 8K, # of batches = 64014
What's in storage?	Criteo click dataset in Gen. 4 drive	Criteo click dataset in Gen. 4 drive	Preprocessed dataset in 2 Gen. 4 drives (RAID0)
Run time (secs)	1900	5181	445
% Read Volume (#)	72 (7.7M)	55 (17M)	100 (469K)
Perf. (MBps) (5)	$1500\text{-}6000_{Read}$ 3000_{Write}	$500\text{-}6000_{Read}$ $1800\text{-}3000_{Write}$	454_{Read}
QD	$250_{mean} \rightarrow 10_{mean}$	1-11	4-5
Read Payload (KB) (1)	$512_{90\%}$	$512_{89\%}$	$512_{71\%}$
Read – Sequential Volume % (persistence count > 50, multi-threaded) (2)	43-55	50-90 (in large portions of the trace)	68
Write Payload (KB) (3)	$1280_{65\%}$	$1280_{40\%}$	N/A
Write – Sequential Volume % (persistence count > 50, multi-threaded) (4)	85-95	90-99 (in large portions of the trace)	N/A

Takeaways

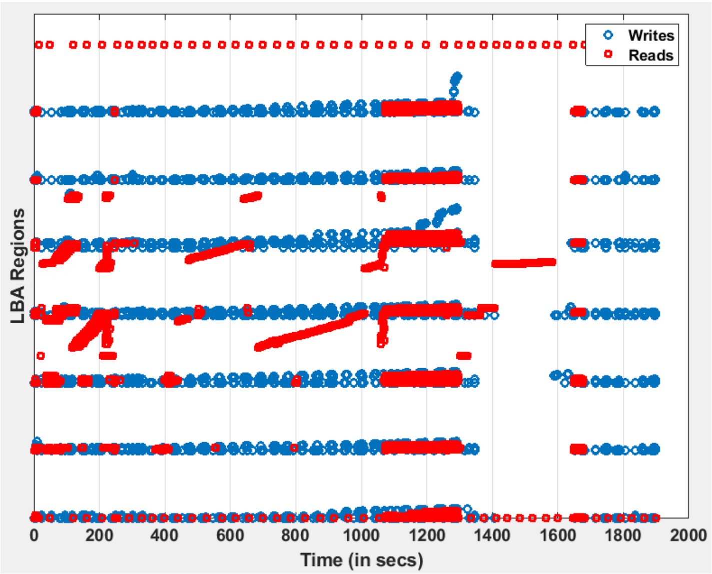
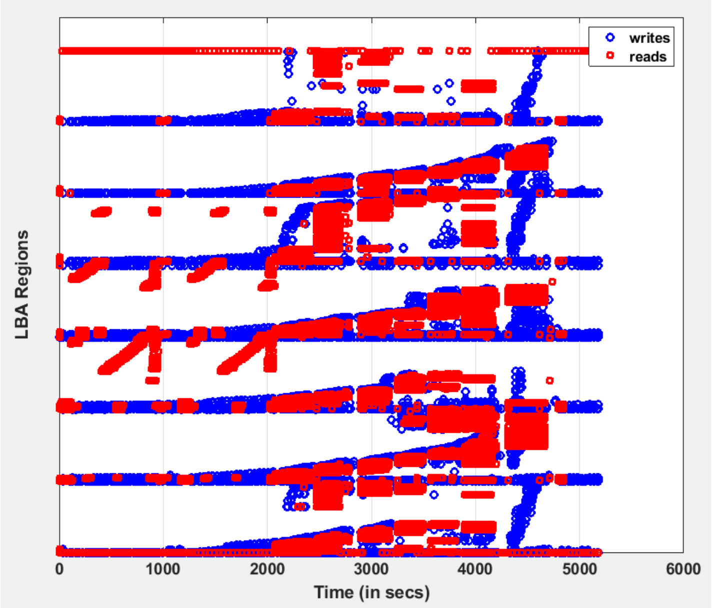
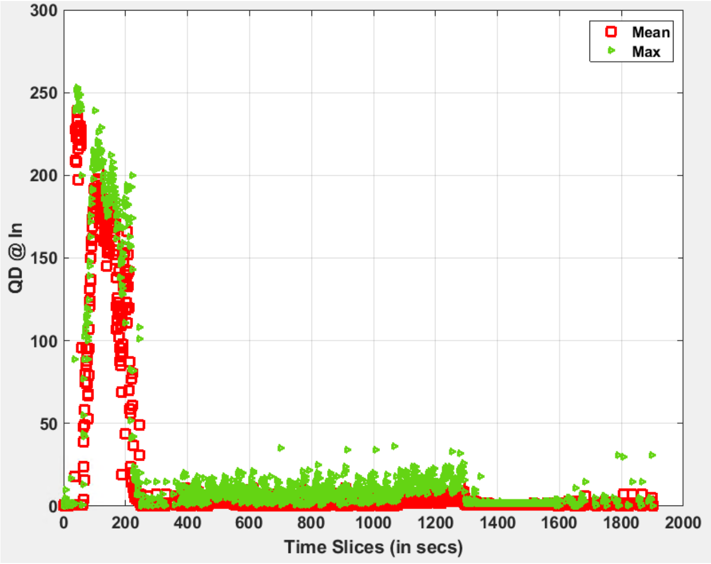
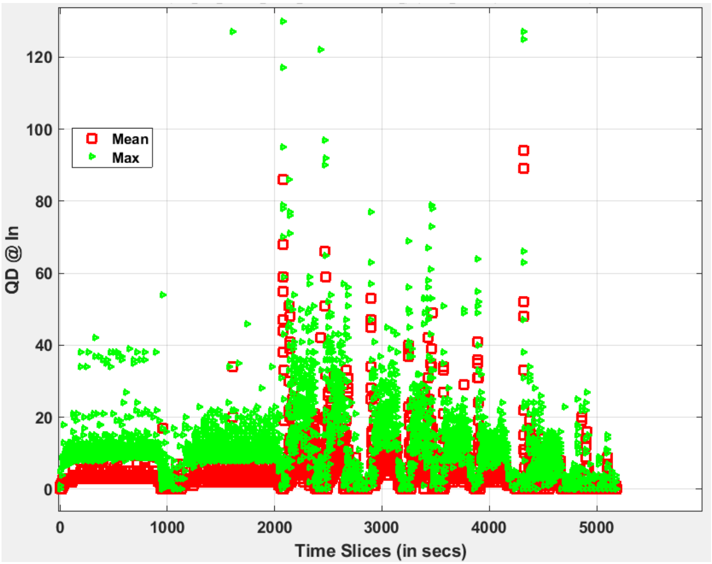
1 3 Reads and write payloads are large

2 4 Significant volume of read and write data is sequential

5 Demands on storage can be time-variant – small MBPs to saturation

DLRM – Deep learning recommendation model

Storage Trace Analysis – Summary (2/2)

DLRM Preprocessing w/ GPU	DLRM Preprocessing w/ CPU
	
	

Storage Trace Analysis	DLRM Preprocessing w/ GPU & DLRM Preprocessing w/ CPU
Takeaways	❖ GPU preprocessing is set up differently from CPU preprocessing ❖ In both runs, q-depths aren’t constant, there is significant variation across a run ❖ Note, LBA regions show visual evidence of sequential accesses

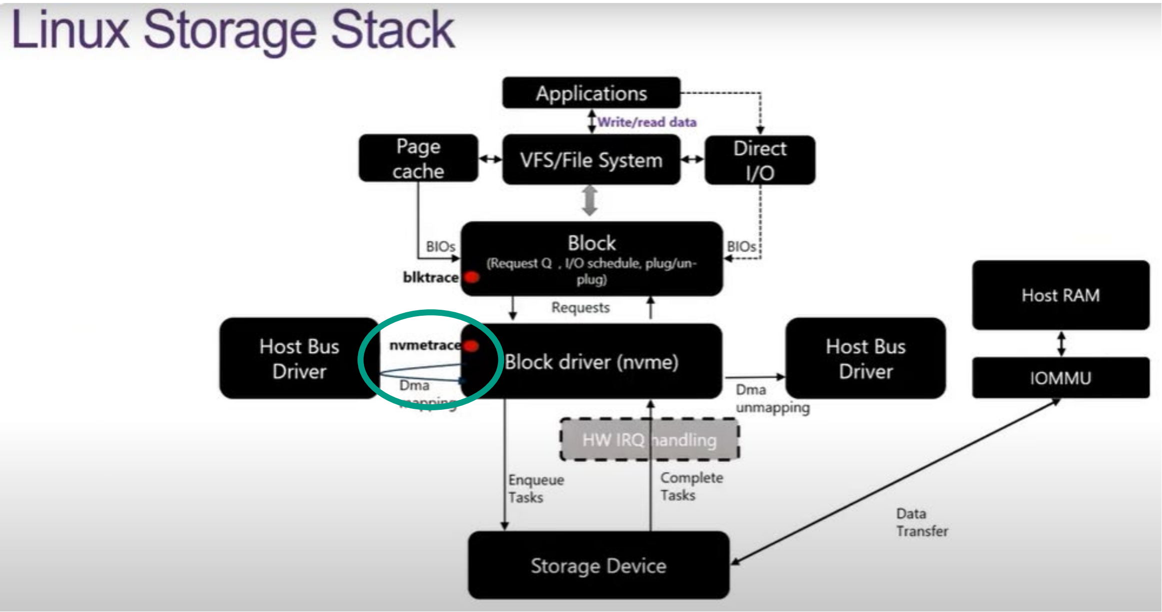
Q & A

Takeaways	❖ Rapid increase in training dataset size and the variety of models to train will continue to put pressure on storage density, SSD capacities and throughput requirements
	❖ With increasing demand for energy from datacenter/edge devices, there will be continued pressure to make storage energy-efficient
	❖ DLRM is a key production model, and requires high-capacity and throughput from SSDs for training purposes
	❖ We examine storage traces of ▪ DLRM Data Preprocessing (under discussions to be part of MLPerf Storage suite) ▪ DLRM Training (part of MLPerf Training suite)
	❖ Storage trace analysis of AI workloads show evidences of ▪ sequential read (write) accesses ▪ large payloads for reads and write commands
Call to Action	❖ ML Commons may want to consider creating a suite that places data preprocessing inline with DLRM training

Backup

Ref / Credit: John Mazie, Understanding Applications Through NVMe Driver Tracing Using BPF | SDC 2024 (sniadeveloper.org)

NVMe Trace



NVMe tracing happens in the block driver

- Collections information on every transaction in the nvme driver.
  - Starting LBA
  - Transaction Size/Length
  - Start Time/Completion Time/Latency
  - Process ID/Name
  - Device
  - Queue ID
  - Transaction Type
    - Read, write, flush, admin…