SNIA DEVELOPER CONFERENCE

SDC 24

BY Developers FOR Developers

September 16-18, 2024
Santa Clara, CA

What is the Role of Flash in Data Ingestion within the AI Pipeline?

Sundararajan (Sundar) Sankaranarayanan, Suresh Rajgopal, Sujit Somandepalli

Micron

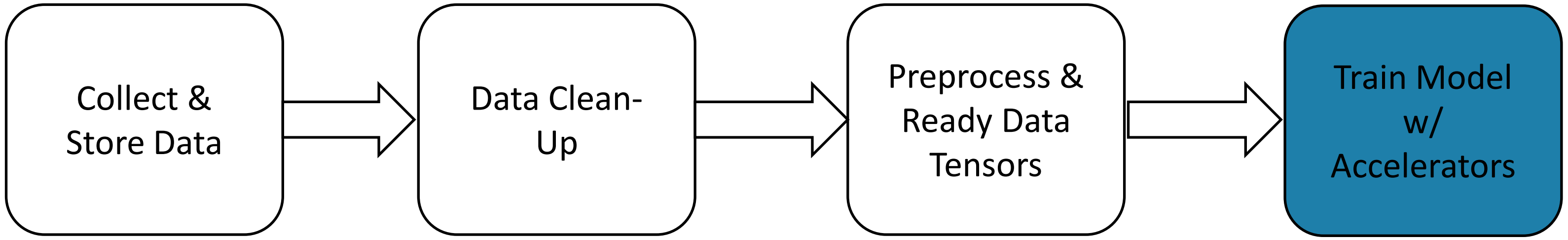
Outline

- Key takeaways
- Why is Data Storage and Ingest (DSI) key part of the AI pipeline?
- How Flash storage plays a key role in solving storage capacity vs. IO bandwidth scaling challenge?
- Results from storage trace analysis of DLRM preprocessing and training workloads
- Call to enhance existing MLPerf DLRM benchmark

Key Takeaways

Data storage and ingestion (DSI) is a critical part of the AI pipeline
Deep Learning Recommendation Model (DLRM) training presents key challenges to storage capacity and IO scaling – Flash storage is key to tackling the scaling challenge
MLPerf Training DLRM benchmark captures training portion of the model
As a call to action, we highlight the need for an extension to the benchmark that captures DSI aspects of DLRM training

AI Model Training – A Simple Pipeline



- AI models and accelerators are often the focus of a training pipeline
- The data storage and ingest (DSI) portion of the pipeline is equally critical
- DSI is designed to feed the accelerators without stall
- In this talk, we will focus on storage and ingest portion of the DLRM training pipeline

DSI – Data Storage and Ingest, DLRM – Deep learning recommendation model

Deep Learning Recommendation Model (DLRM) – Scale & Significance

Ref / Credit: Meta's DLRM Papers

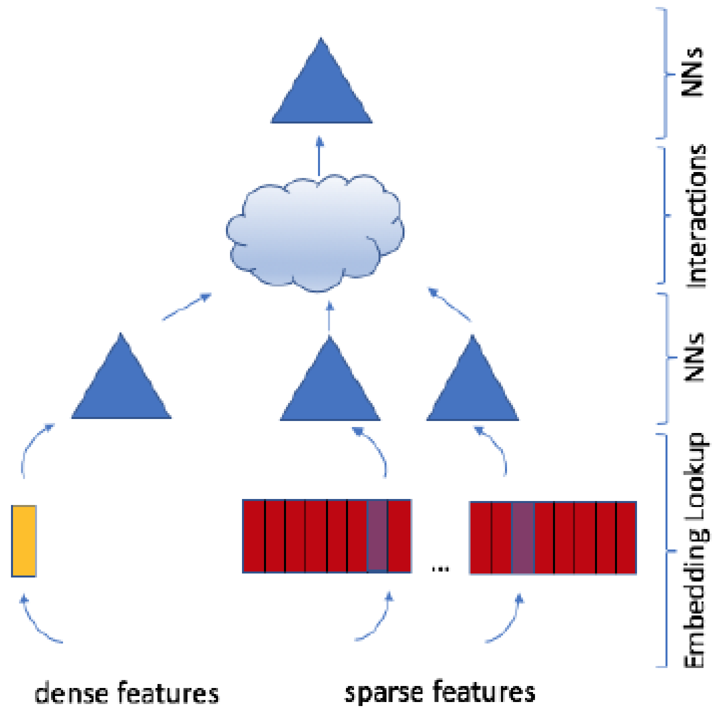


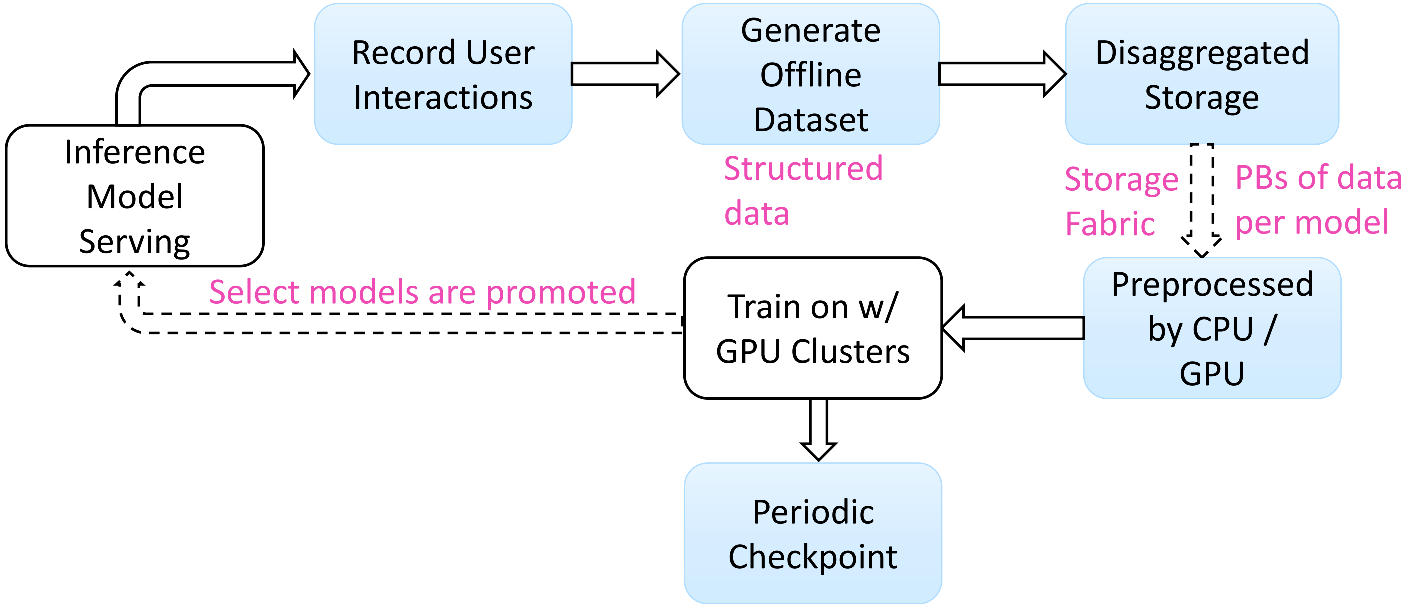
Figure 1: A deep learning recommendation model

- Recommendation models are backbone for Meta, Netflix, Google etc.
- Multiple DLRM models are maintained, and new models are continuously trained and developed
- Model parameters:
  - MLP
  - 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 model embedding table is 96 TB (or 24 TB compressed)
- Models are trained on Petabytes of data
- Scalable DSI is critical in transporting PBs of data to accelerators

DSI – Data Storage and Ingest, DLRM – Deep learning recommendation model, DRAM – Dynamic Random Access Memory

 |

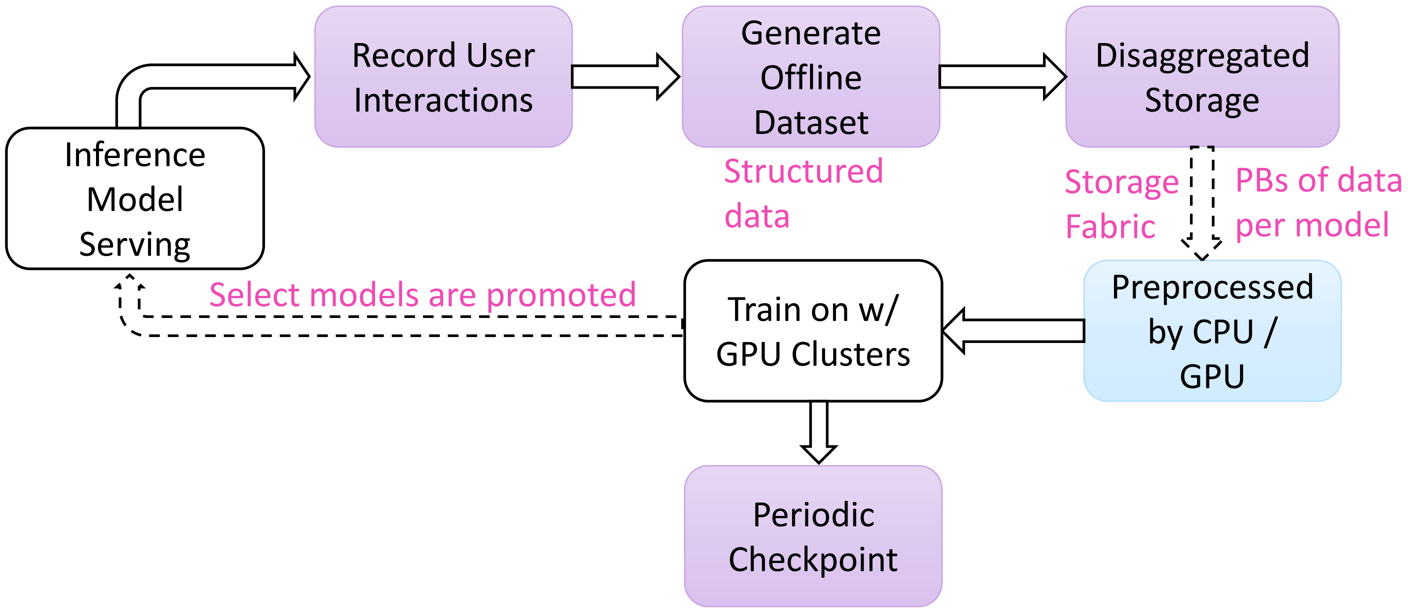
DLRM Training Pipeline– An Illustration



- ❑ DLRM pipeline is a closed system – DSI portion of the pipeline is highlighted in Blue
- ❑ Data for subsequent training grows with every user interaction
- ❑ Petabytes of data is read out of storage → preprocessed to form tensors → fed to trainers
- ❑ DLRM training is a single-epoch process
- ❑ Total power budget = power for Storage + power for preprocess + power for trainers

DSI – Data Storage and Ingest, PB - Petabytes

Role of Storage in the DLRM Pipeline

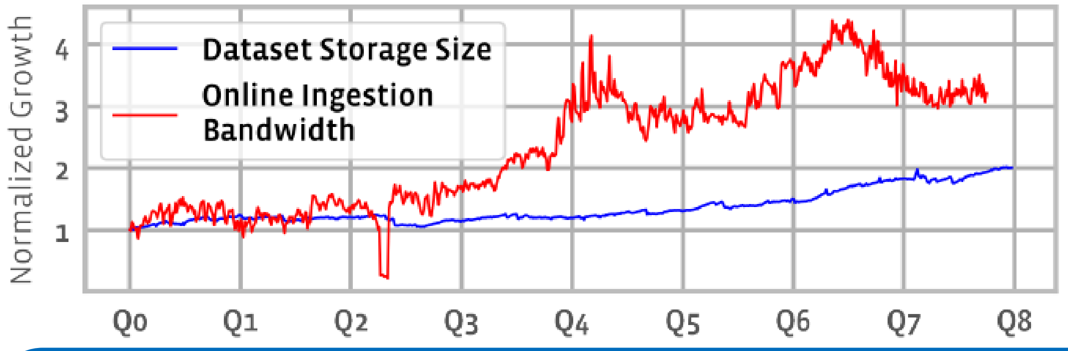


- User interactions are logged and stored in a database
- Logged data is cleaned and structured and stored as Data Warehouse tables
- Tables, often in compressed formats, are stored in distributed storage lake – often exabyte scale of storage
- Periodic checkpointing during training is critical to recover from failures
- Storage and IO bandwidth needs are met with the right mix of HDDs and SSDs ← **Scaling Challenge**

DSI – Data Storage and Ingest, PB – Petabytes, HDD – Hard Disk Drive, SSD – Solid State Drive

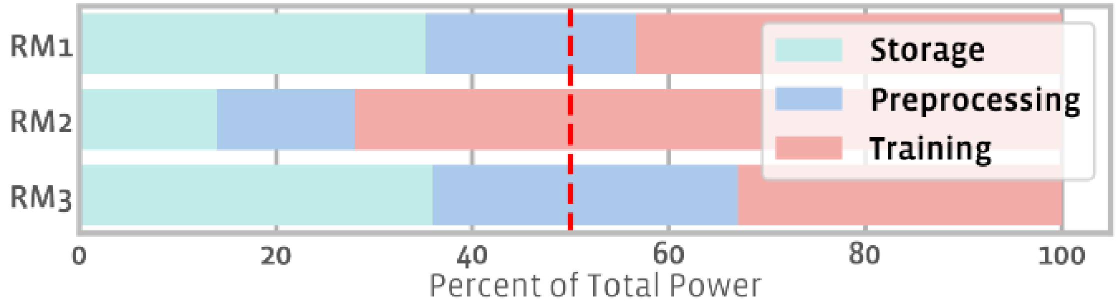
Scaling Challenge for AI Training – Meta's Solution

Ref / Credit: Meta's Understanding DSI paper



- ❑ Dataset storage size has grown 2x in 2 years
- ❑ IO bandwidth demand has grown 4x in 2 years
- ❑ A scalable architecture should meet storage and IO demands

Ref / Credit: Meta's Understanding DSI paper



- ❑ > 40% power is spent on Storage + Preprocess
- ❑ Less power for trainers ➔ inefficient use of compute
- ❑ Power is an important dimension in scaling

- ❑ Meta solves IO scaling with an SSD caching layer over Tectonic (HDD-based) called Shift
- ❑ Data in Shift is not durable, and Shift leverages underlying Meta's CacheLib
- ❑ Tectonic-Shift saves 29% of power relative to using HDDs alone

Key Characteristics of Meta's DLRM Pipeline

Ref / Credit: Meta's Understanding DSI paper

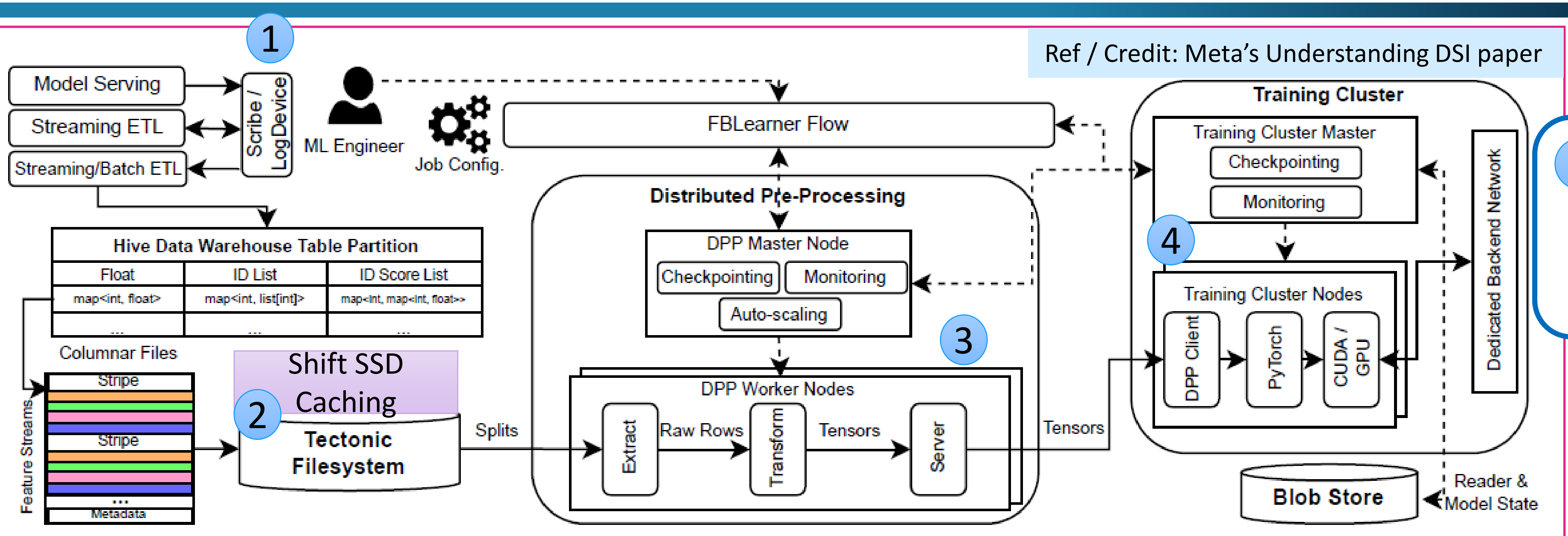


Figure 3: Production data storage and ingestion pipeline architecture. Solid and dashed lines represent data and control flow, respectively.

4 Goal of the pipeline is to keep training cluster fed without stalls

1 User interactions are logged ➔ growth in training dataset

2 Data compressed and stored in chunk store
- ❑ SSD Cache serve AI data → leverages data locality across RM jobs → Help tackle scaling challenge

3 Pre-processing is inline
- ❑ Splits are independent and self-contained work items managed by preprocessing worker nodes

DLRM – Deep Learning Recommendation Model, PB – Petabytes, RM – Recommendation Model, SSD – Solid State Drive

 |

Key Characteristics of Meta's DLRM Pipeline

Ref / Credit: Meta's Understanding DSI paper

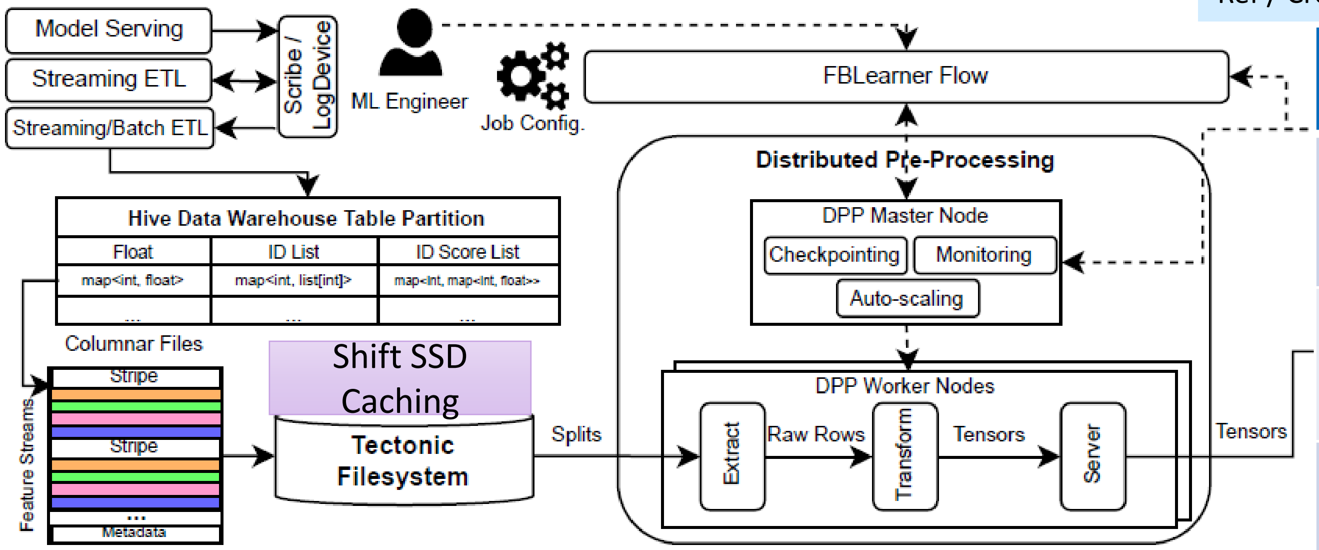


Figure 3: Production data storage and ingestion pipeline architecture. Solid and dashed lines represe

Meta's DLRM Pipeline
Data stored in columnar format in a distributed filesystem
Significant portion of DLRM data is read out of SSD caching layer
Inline preprocessing of data includes decompression, decryption, transformation, and data filtering
Preprocessing is self-contained within a mini-batch operated by a DPP worker

 |

DLRM – Deep Learning Recommendation Model, PB – Petabytes, RM – Recommendation Model, SSD – Solid State Drive

MLPerf Training Benchmark - DLRM

Ref / Credit: https://mlcommons.org/benchmarks/training

- ❑ MLPerf Training benchmark suite measures how fast systems can train models to a target quality metric
- ❑ Each benchmark is defined by a Dataset and Quality Target for training
- ❑ Here, we focus on results from our DLRM benchmarking effort

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	❑ Offline preprocessing of the dataset ❑ Raw data converted to columnar compressed parquet format ❑ Categories converted to contiguous integer representation ❑ Missing numerical values are zeroed and feature values are normalized
System & Tracing	❑ AMD EPYC 7742 128-Core Processor (2x64) ❑ NVIDIA A100 – 8x 40 GB ❑ NVMe Tracing using libpf
Reference	❑ Github-NVIDIA-DeepLearningExamples

DLRM Storage Trace Analysis - Results

Storage Trace	DLRM Preprocessing w/ GPU	DLRM Preprocessing w/ CPU	DLRM Training on GPU
Experimental Setup	1 Preprocessed with 8 GPUs	Preprocessed with 2 64-core CPUs	Trained with 8 GPUs: 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)	4 $1500\text{-}6000_{Read}$ 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)	2 $512_{90\%}$	$512_{89\%}$	$512_{71\%}$
Read – Sequential Volume %	4 43-55	50-90 (in large portions of the trace)	68
Write Payload (KB)	2 $1280_{65\%}$	$1280_{40\%}$	N/A
Write – Sequential Volume %	3 85-95	90-99 (in large portions of the trace)	N/A

Preprocessing is an offline task 1

Read and write payloads are large 2

Writes during preprocessing are sequential 3

Reads on certain portions of the workload are highly sequential 4

Performance demands from storage are time-variant 5

SDC 24

Meta's DLRM Pipeline vs. DLRM Benchmark

Meta's DLRM Pipeline	DLRM Benchmark
Preprocessing is inline with training	Preprocessing is offline and separate from training
Data stored in columnar format in a distributed data warehouse format	Data is converted to a columnar parquet format as part of preprocessing
Training consume PBs of data with portions read out SSDs	Training consumes TBs of data
Key innovations in data placement on storage devices to enable features filtering	Feature column filtering is not supported

Call To Action

- ❑ Create more representative benchmark datasets
- ❑ Place data on storage devices like that in large production pipeline
- ❑ Create an option to make preprocessing inline with training

Key Takeaways

Data storage and ingestion (DSI) is a critical part of the AI pipeline
Flash storage is key to tackling the storage capacity vs. IO bandwidth scaling challenge – often, large dataset for training is stored in flash storage
MLPerf Training DLRM benchmark captures training portion of the model
As a call to action, we highlight the need for an extension to the benchmark that captures DSI aspects of DLRM training

<u>Call To Action</u>

- ❑ Create more representative benchmark datasets
- ❑ Place data on storage devices like that in large production pipeline
- ❑ Create an option to make preprocessing inline with training

Micron

SDC 24