Hewlett Packard Enterprise

AI/ML Benchmarking and Lustre

Sakib Samar

May, 2024

Agenda

Personal and team introductions

MLPerf Storage Overview

- Objectives of the working group
- Outline of the workloads
- Goals of the tests
- Performance metrics
- Accelerator Emulation

Our Path with Lustre

- Next Steps

Personal and team introductions

Personal and Team Introductions

- Part of HPC Storage team within Hewlett Packard Enterprise (HPE), specializing in Lustre development and optimization.
- Engaged with running AI/ML benchmarks in recent months, with a focus on Lustre performance and scalability for HPC environments.
- Primarily utilizing the MLPerf Storage, which is a benchmarking suite provided by MLCommons.
- These tests were conducting using Lustre 2.15 builds.

MLPerf Storage Overview

MLPerf Storage Overview

Introduction

- MLCommons is an **AI engineering association**
  - Encourages open collaboration from the industry and academia to improve AI systems.
  - Its primary focus areas include benchmarking, datasets, and research.

- The **MLPerf benchmarks**, provided by MLCommons, are standardized metrics to evaluate system performance using AI/ML Workloads.
  - **MLPerf Storage,** one of the MLPerf Benchmarks, assess how fast storage systems can process inputs and produce results using a trained model.
  - Other benchmarks within MLCommons include MLPerf Training, MLPerf Inference, MLPerf HPC, among others.
  - Each Benchmarking Suite has their own working group responsible for defining the AI models, datasets, used and benchmarking rules.

MLPerf Storage Overview

Workloads

- The MLPerf Storage has one release so far, and two upcoming releases scheduled.

- These are the confirmed workloads for **MLPerf Storage v0.5**

Area	Problem	Model	ML Framework	Dataset seed
Vision	Image Segmentation	3D U-Net	PyTorch	KiTS 19 (140MB/sample)
Language	Language Processing	BERT-large	PyTorch	Wikipedia (2.5KB/sample)

- These are the proposed workloads for **MLPerf Storage v1.0**

Area	Problem	Model	ML Framework	Dataset seed
Vision	Image Segmentation	3D U-Net	PyTorch	KiTS 19 (140MB/sample)
Vision	Image Classification	ResNet-50	PyTorch	ImageNet (150KB/sample)
Scientific	Cosmology parameter prediction	CosmoFlow	PyTorch	CosmoFlow N-body simulation (2MB/sample)

MLPerf Storage Overview

Workloads – Dataset Generation

- MLPerf Storage utilizes **DLIO** to generate synthetic data, by following the sample size distribution and dataset seed structure.
- Each MLPerf Storage benchmark run requires a **minimum dataset size**:
  - The benchmark script calculates this based on the available memory of the host node.
  - The minimum dataset size is set at 5 times the available memory to ensure randomness of the access patterns can defeat any significant levels of caching in local DRAM
- There are **no restrictions** for using **datasets larger** than minimum dataset size.

MLPerf Storage Overview

Goals of the Tests

- MLPerf Storage releases **do not require running of the actual training jobs**.
- As a result, **no hardware accelerators like GPUs or TPUs are required**.
- Accelerator emulation involves training on the accelerator for a single batch of data with a **sleep command.**
  - The sleep interval depends on the batch size and accelerator type, and has been determined through measurement on a system running the actual training workload.
- This enables the use different types of accelerators, accomplished by using code from the **DLIO (Deep Learning I/O)** benchmark.
- The MLPerf Storage v0.5 employs simulated NVIDIA V100 GPUs, while MLPerf Storage v1.0 uses NVIDIA A100 and NVIDIA H100 GPUs.

```
[INFO] Averaged metric over all epochs
[METRIC] ==========================================================
[METRIC] Number of Simulated Accelerators: 8
[METRIC] Training Accelerator Utilization [AU] (%): 99.3699 (0.0183)
[METRIC] Training Throughput (samples/second): 79.9480 (4.0618)
[METRIC] Training I/O Throughput (MB/second): 11177.4673 (567.8716)
[METRIC] train_au_meet_expectation: success
[METRIC] ==========================================================
```

Metrics

Here are the **performance metrics** listed after each MLPerf Storage job:

- Number of Simulated Accelerators used for the job
- Accelerator Utilization (AU) during the benchmark run
- Samples/second
- Training I/O Throughput
- AU Meet Expectation

MLPerf Storage Overview

Metrics – Accelerator Utilization (AU)

- The Accelerator Utilization (AU) represents the percentage of total test time during which accelerators are active, with a minimum requirement of 90% to pass the benchmark.
  - The AU is computed as seen below:

```
AU (percentage) = (total_compute_time/total_benchmark_running_time) * 100
```

  - Compute time begins once I/O requests are completed and the data is ready for the computation.
  - Total compute time measures the duration for the specific GPU to process the data retrieved by the I/O calls.
  - The *train_au_meet* expectation will indicate the benchmark has failed when below 90%.

- The performance metric used in the benchmark is samples/second, **contingent upon achieving minimum accelerator utilization (AU) of 90%.**
  - The I/O operations that are excluded from the AU calculation are included in the samples/second reported by the benchmark.

MLPerf Storage Overview

Metrics – Accelerator Utilization (AU)

- The total compute time can be derived from the batch size, total dataset size, number of simulated accelerators, and sleep time:

```
total_compute_time = (records_per_file * total_files) / simulated_accelerators / batch_s
ize * computation_time * epochs.
```

- MLPerf Storage workloads will always run for 5 epochs. The AU is calculated by taking the average of the across all the epochs.
  - Checkpointing is performed at the end of the second epoch, involving small write operations.

MLPerf Storage Overview

The Storage working group

- The **storage working group**, facilitated by MLCommons and the MLPerf Storage chairs, help define the AI models, datasets, and benchmarking rules.
- One of the key objectives of the working group is to **emulate the I/O patterns of the other MLPerf Training**, without using any hardware accelerators
  - This approach enables testing the behavior of the storage system in large-scale scenarios without the need to acquire corresponding compute infrastructure
- **MLPerf Training** is the first MLPerf benchmark within MLCommons. It measures how fast systems can train models to a target quality metric.

MLPerf Storage Overview

Objectives of the Storage Working Group

- The submission of a MLPerf Storage run would include the results of 5 benchmark jobs, where each results should within a 5% margin.
  - This puts an emphasis on reproducibility for valid and consistent results.

- With MLPerf Storage, the Storage working group aims for:
  - **Comparability** between benchmark submissions to enable decision making by the AI/ML Community.
  - **Flexibility** to enable experimentation and to show off unique storage system features that will benefit the AI/ML Community.

- To achieve these goals, the working group has defined two submission categories,
  - In the **CLOSED** submission, no changes are allowed. The intent is to have a level playing field so that the results are comparable across all submissions. It restricts flexibility to ensure easier comparability.
  - In the **OPEN** submission, there is more flexibility to tune and change benchmark and storage configurations. All changes here must be disclosed.

MLPerf Storage Overview

Parameter	Description	Default
Dataset parameters		
dataset.num_files_train	Number of files for the training set	--
dataset.num_subfolders_train	Number of subfolders that the training set is stored	0
dataset.data_folder	The path where dataset is stored	--
Reader parameters		
reader.read_threads	Number of threads to load the data	--
reader.computation_threads	Number of threads to preprocess the data(only for bert)	--
Checkpoint parameters		
checkpoint.checkpoint_folder	The folder to save the checkpoints	--
Storage parameters		
storage.storage_root	The storage root directory	./
storage.storage_type	The storage type	local_fs

Submission Category – CLOSED

MLPerf Storage Overview

Parameter	Description	Default
framework	The machine learning framework.	3D U-Net: PyTorch ResNet-50: PyTorch Cosmoflow: Tensorflow DLRM: PyTorch
Dataset parameters		
dataset.format	Format of the dataset.	3D U-Net: .npz ResNet-50: .tfrecord Cosmoflow: .tfrecord
dataset.num_samples_per_file	Changing this parameter is supported only with Tensorflow, using tfrecord datasets. Currently, the benchmark code only supports num_samples_per_file = 1 for Pytorch data loader. To support other values, the dataloader needs to be adjusted.	3D U-Net: 1 ResNet-50: 1 Cosmoflow: 1
Reader parameters		
reader.data_loader	Supported options: Tensorflow or PyTorch. OPEN submissions can have custom data loaders. If a new dataloader is added, or an existing data loader is changed, the DLIO code will need to be modified.	3D U-Net: PyTorch (Torch Data Loader) ResNet-50: PyTorch (Torch Data Loader) Cosmoflow: DALI

Submission Category – OPEN

MLPerf Storage Overview

Next steps for the Storage Working Group

- Release of MLPerf Storage v1.0:
  - Introducing new workloads emulating MLPerf Training and MLPerf HPC (Resnet-50 and CosmoFlow).
  - Incorporating new GPU emulations: NVIDIA A100 and NVIDIA H100.
  - Transitioning to Distributed Training to replace multiclient benchmarks from the previous MLPerf Storage v0.5 release.
- Addition of data pre-processing – an important step during training, which exerts a significant load on the storage systems.
- Expanding the range of workloads. Including large language models like GPT3 and a diffusion models like Stable Diffusion.
- Adding a broader variety of accelerators to the benchmarking scope.

Our path with Lustre

Our path forward

Discussing next steps on our path with Lustre

- Preparing a submission for MLPerf Storage v1.0 using **HPE Cray ClusterStor E1000**.
- Investigating opportunities to optimize access to smaller files found in certain workloads.
- Investigating the benefits of **Hybrid I/O**, with current ML frameworks.
  - For **Buffered I/O** paths, measuring the benefits of automatically switching to **Direct I/O** for larger IO sizes.
- Identifying potential challenges and opportunities with additional workloads on Lustre.

Thank you

Sakib Samar
Contact: sakib.samar@hpe.com