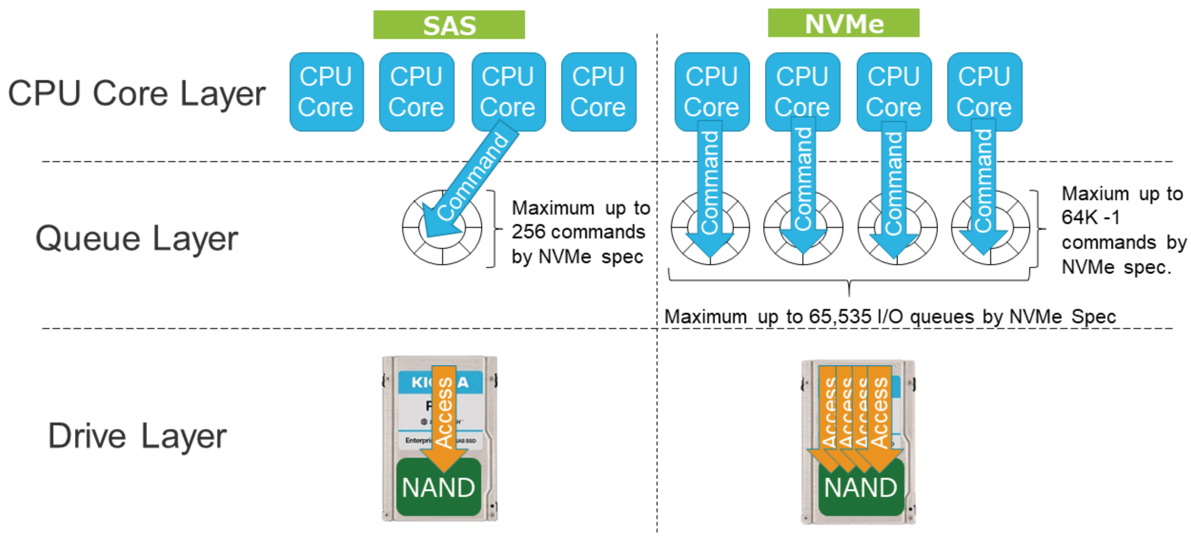
GPUDirect Storage Perfromance Report and NVMe-oF™ Contribution

We, KIOXIA provides a variety of memory storage products with the mission of uplifting the world with “memory”. You may wonder what storage has to do with AI, but we believe that high-speed storage can contribute to AI and have been working on various collaborations. For example, drawing cartoons and auto-generating quizzes with AI.

Our products include flash memories used as components in various products, SSDs for PCs and servers, memory cards and USB drives for consumer electronics, and more. Among these products and technologies, let us introduce NVMe™ SSD technology related to AI and machine learning acceleration.

First, here are the differences between traditional SCSI SAS and NVMe. The SAS method on the left enables input/output (IO) requests for a single I/O queue. Although this was good enough during the era when drives such as HDDs were the primary storage media, as NAND flash memory came into use as storage media, it became difficult to fully exploit the performance that flash memory can deliver.
In response, NVMe method, shown on the right was defined as a standard. This method allows multiple queues to send I/O requests in parallel, and storage media can be accessed in parallel as well. By increasing the parallelism in this way, NAND flash memory performance can be utilized.

NVMe SSD Advantage

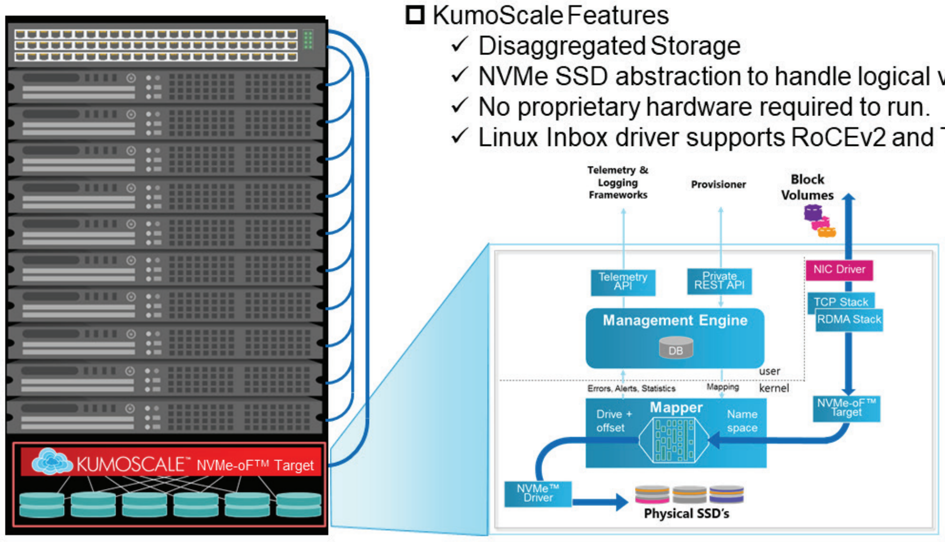

SAS
NVMe
CPU Core Layer
CPU Core
Command
Queue Layer
Maximum up to 256 commands by NVMe spec
Maxium up to 64K -1 commands by NVMe spec.
Maximum up to 65,535 I/O queues by NVMe Spec
Drive Layer
Access
NAND

This NVMe SSD is commonly utilized as a high-speed local storage. However, KIOXIA has developed KumoScale™ software to maximize the use of NVMe SSDs, which are expected to be further advanced in performance by standardization organizations in the future.

Here is an introduction to KumoScale. KumoScale is a software product that provides storage disaggregation with NVMe-oF. KumoScale provides not only a high-speed data path, but also administrative functions, integration and adaptation with telemetry and orchestration tools. In addition, generally available NICs and LINUX™ in-box drivers should work for KumoScale as NVMe-oF storage without any special hardware or software to build.

NVMe-oF Target Software : KumoScale Overview

☐ KumoScale Features
- ✓ Disaggregated Storage
- ✓ NVMe SSD abstraction to handle logical volume.
- ✓ No proprietary hardware required to run.
- ✓ Linux Inbox driver supports RoCEv2 and TCP



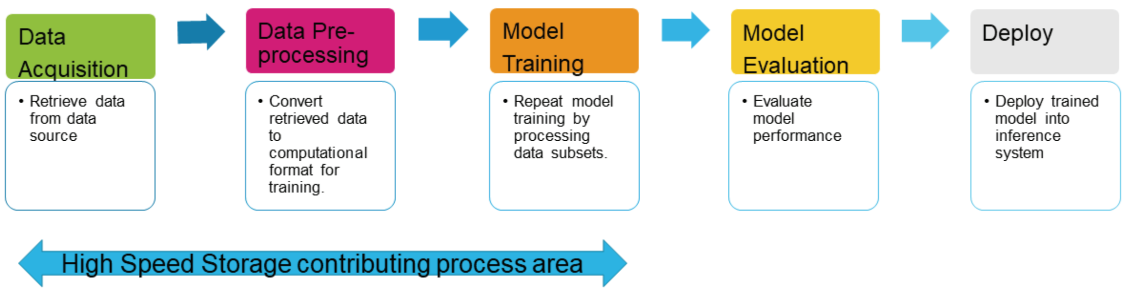
As mentioned earlier, we believe our high-speed storage products and technologies can contribute to AI and machine learning, but there is still a lack of technical information and experience to build and operate such an environment in the general public.

Current Challenge on Machine Learning Environment – Kioxia Observation

High performance GPU and high speed NVMe SSD are relatively easy to configure. But not widely spread due to low visibility.

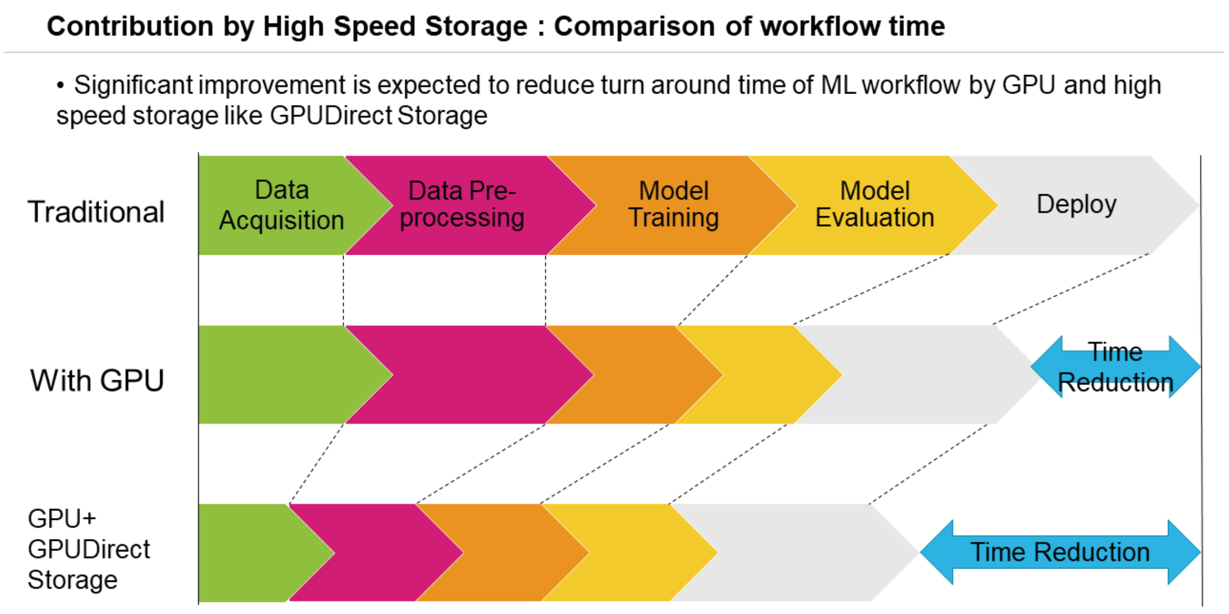
High speed network storage is getting easier to configure than before.

• NVMe/NVMe-oF makes smaller deployment easier for workflow designating expensive and dedicate storage system for large scale and high performance GPU application


Data Acquisition
• Retrieve data from data source
Data Pre-processing
• Convert retrieved data to computational format for training.
Model Training
• Repeat model training by processing data subsets.
Model Evaluation
• Evaluate model performance
Deploy
• Deploy trained model into inference system
High Speed Storage contributing process area

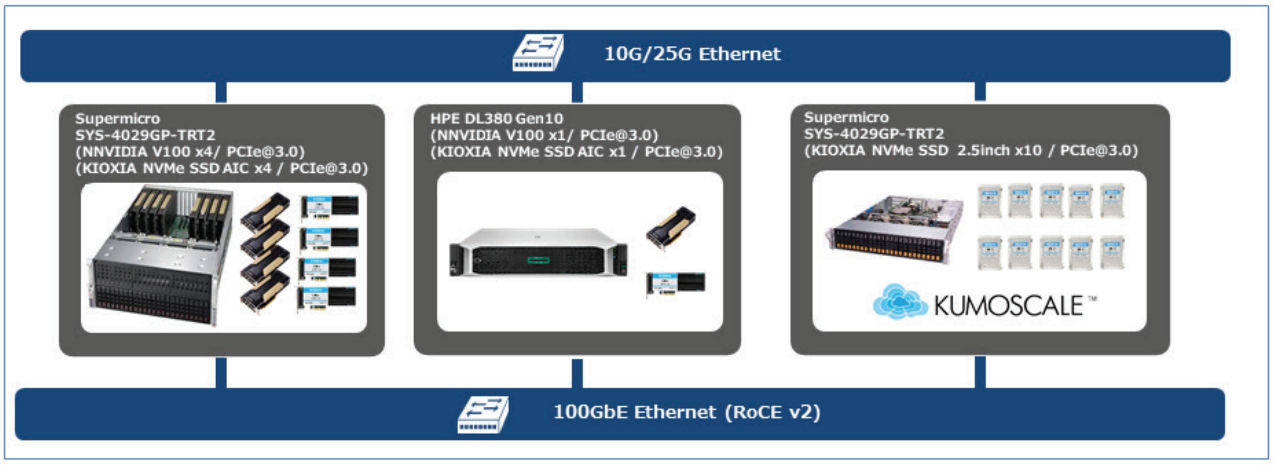
We examined the benefits of using high-speed storage products and technologies in a machine learning workflow, considering that processing performance of storage can reduce workflow processing time in a workflow of acquisition, preprocessing, learning, evaluation, and implementation. The following slides provide concrete examples.

In this figure, each processing time during workflow is represented by lengths of arrows. Based on the machine learning workflow using the CPU at the top row, the processing time for learning and evaluation can be reduced by using a GPU shown in the middle. Furthermore, we believe the use of high-speed storage can reduce the acquisition and preprocessing time, as shown in the bottom row.


Contribution by High Speed Storage : Comparison of workflow time
• Significant improvement is expected to reduce turn around time of ML workflow by GPU and high speed storage like GPUDirect Storage
Traditional
Data Acquisition
Data Pre-processing
Model Training
Model Evaluation
Deploy
With GPU
Time Reduction
GPU+ GPUDirect Storage
Time Reduction

However, in reality, we lacked expertise in utilizing storage in a machine learning environment internally and had not been able to evaluate the effects of using high-speed storage to improve workflow, etc.
Furthermore, with the recent emergence of GPUDirect Storage, a GPUDirect technology, further improvements in machine learning workflows are expected.

Here is an overview of the resource configuration of the environment we tested. A high-specification GPU server equipped with local NVMe. RoCE v2-compatible 100GbE Ethernet was connected to test the performance of NVIDIA GPUDirect Storage using local NVMe and NVMe-oF by KumoScale.


10G/25G Ethernet
Supermicro
SYS-4029GP-TRT2
(NNVIDIA V100 x4/ PCIe@3.0)
(KIOXIA NVMe SSD AIC x4 / PCIe@3.0)
HPE DL380 Gen10
(NNVIDIA V100 x1/ PCIe@3.0)
(KIOXIA NVMe SSD AIC x1 / PCIe@3.0)
Supermicro
SYS-4029GP-TRT2
(KIOXIA NVMe SSD 2.5inch x10 / PCIe@3.0)
KUMOSCALE™
100GbE Ethernet (RoCE v2)

Next, the key aspects of GPUDirect Storage environment setup will be covered.

When we started our validation, GPUDirect Storage environment setup examples were rather limited and there was little information available at the time, even if we wanted to refer to NVMe-based measurement methods. Also, the required configuration was not clear, except for the use of GPUs. Regarding benchmarking methods, we did not understand what tools were available and how to demonstrate the effectiveness.

Here are points to keep in mind based on our experience in building them. First, when it comes to GPU servers, it is essential to understand PCIe® internal topology. We experienced performance degradation for the same server if misconfigured, compared to a normal I/O operation. Also, because PCIe is involved, BIOS settings are also relevant.
And it is also important to consider the combination of software configurations. We had to check software versions for each functional layer in hierarchy to see if they were supported. Here is an example.

Important Notice to configure GPUDirect Storage

□GPU Server Model
→ It is necessary to understand internal topology of GPU,NVMe SSD and NIC.

□BIOS(EFI) Setting
→ Disable IOMMU (Input-Output Memory Management Unit), PCI ACS (Peripheral Component Interconnect Access Control Services)

□Software Combination
→kernel module(driver) ,Library,ML Framework compatibility

1. GDS kernel module vs CUDA Toolkit version
2. ML Framework vs CUDA Toolkit version
3. MLNX OFED Version when NVMe-oF is configured.

etc…

Generally, information about nvidia-fs says version 10.0 or later is supported. In fact, however, nvidia-fs and gds-tools have been available as GPUDirect Storage-related packages from version 11.4. Since GPUDirect Storage development is actively progressing, it is important to check documentation for changes like those mentioned above.

Software Combination Sample

□GDS Kernel Module：nvidia-fs prerequisites
https://github.com/NVIDIA/gds-nvidia-fs

```
Requirements
    •NVIDIA Tesla or Quadro class GPUs based on Pascal, Volta, Turing or Ampere
    •NVMe/NVMeOF storage devices or supported distributed filesystem
    •Linux kernel between 4.15.0.x and 5.4.0.x
    •MOFED 5.1 or above
    •cuda toolkit 10.0 and above
    •GPU display driver >= 418.40
```

□CUDA Toolkit 11.4 or greater provides GDS packages.
✓Relevant Package: nvidia-fs, gds-tools
→ Need to change CUDA Tooklit 11.2 (officially announced) to 11.4(reality)

Recommendation :check latest document as GDS development is actively going on.

Software configuration for this benchmark is shown below.

Benchmark Software

- ☐ GPU Server System
  - ✓ OS: Ubuntu Server 20.04.4 LTS
  - ✓ NVIDIA Driver: 470.57.02
  - ✓ CUDA Toolkit: 11.4.1
- ☐ GPUDirect Storage
  - ✓ nvidia-fs : 2.7.50
  - ✓ gds-tools : 1.0.1.3 (gdsio version :1.5)*

    * gds-tools package provides binaries for data verification, GDS config verification and a GPU based synthetic IO benchmarking tool.
- ☐ NVMe-oF
  - ✓ MLNX OFED : 5.4-1.0.3.0
  - ✓ KumoScale : 3.19

Here is the benchmark tests we performed this time. For the benchmark tool, we decided to use the gdsio provided in gds-tools. A script called gds_perf.sh, which executes gdsio for different block sizes and modes, is also available in gds-tools, and we used this script to compile the measurement results.
The measurement results are output in CSV format as shown on this slide. From this output, a comparison graph was drawn comparing GPUDirect Storage and I/O via conventional CPUs.

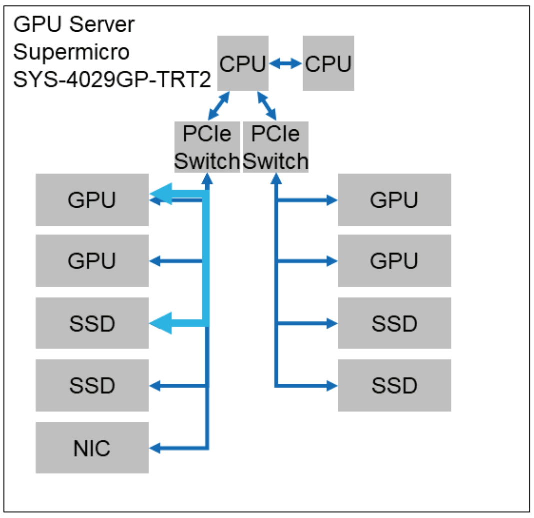
The measurement environment is as following configuration.

Benchmark Configuration

- ☐ GPU Server
  - ✓ Supermicro® SYS-4029GP-TRT2
  - ✓ CPU: Intel® Xeon® Gold 6148 CPU @ 2.40GHz (2 CPU x 20 core)
  - ✓ Memory: 384GB
  - ✓ 100 GbE NIC: NVIDIA ConnectX-5
  - ✓ Local SSD: NVMe SSD KIOXIA CM5 (1.6 TB) x 4
  - ✓ GPU: NVIDIA GV100GL x 4
- ☐ NVMe-oF Target
  - ✓ Supermicro AS-2113S-WN24RT (PCIe 3.0)
  - ✓ CPU: AMD EPYC™ 7702 64-Core Processor
  - ✓ Memory: 512 GB
  - ✓ NIC: NVIDIA ConnectX-6 (100 GbE single port only)
  - ✓ SSD: NVMe SSD KIOXIA CM6 x 10
- ☐ Network Switch
  - ✓ NVIDIA SN2700
  - ✓ Onyx 3.7.1200

First, here are the results of GPUDirect Storage from a single local NVMe SSD to a single GPU. The internal topology of the GPU server is shown on the right side of this slide. One of the two CPUs on this GPU server has two PCIe switches connected to the GPU, SSD, and NIC, respectively.

Case Local-1: Single Local NVMe SSD and GPU

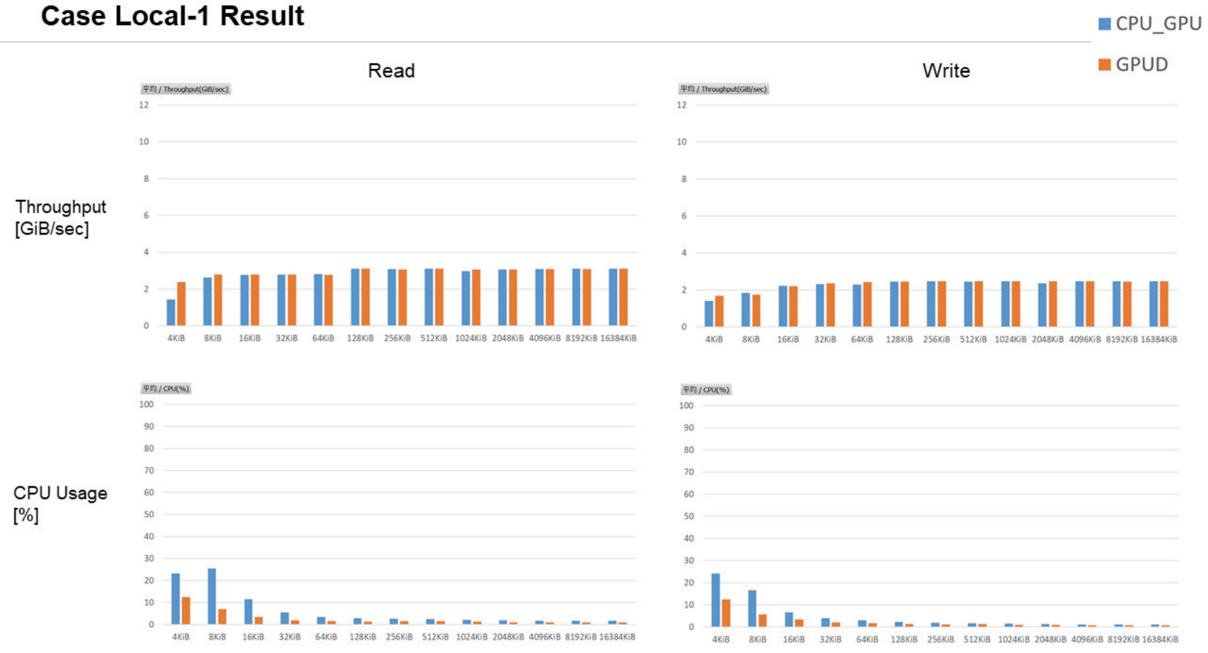


The measurement of local NVMe SSDs is performed by designating to use GPUs and NVMe SSDs under the same PCIe Switch. Note that if GPUDirect Storage is performed from the GPU under the right PCIe Switch incorrectly designating the NVMe SSD under the left PCIe Switch, the expected performance will not be obtained. To specify above for GPU and storage, set environment variables for GPU and SSD-mounted directory as script data storage directory for storage.

Here are the measurement results. In this table, Read on the left and Write on the right, throughput in the upper row, and CPU utilization in the lower row. Here are how to read the graph. The blue bar graph shows the I/O results using a conventional CPU. Orange bar graphs represent the I/O performance results using GPUDirect Storage.

The x-axis shows the I/O block size, with 4KiB on the left end and 16MB on the right end. The vertical axis shows the maximum throughput of 12 GiB/sec, and the maximum CPU utilization of 100%. The local results here show that GPUDirect Storage has an advantage in throughput at 4KiB for both Read and Write.
However, for other block sizes, the results appear to be almost identical. Regarding CPU utilization, on the other hand, GPUDirect Storage has lower CPU utilization in most cases. These results show that GPUDirect Storage is effective in reducing CPU resources as well as improving performance.

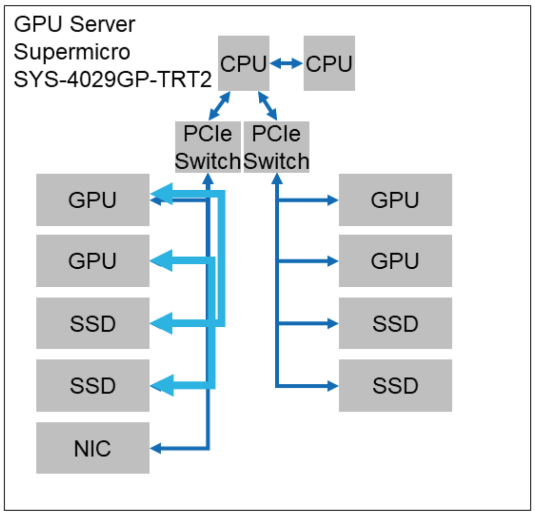
Case Local-1 Result



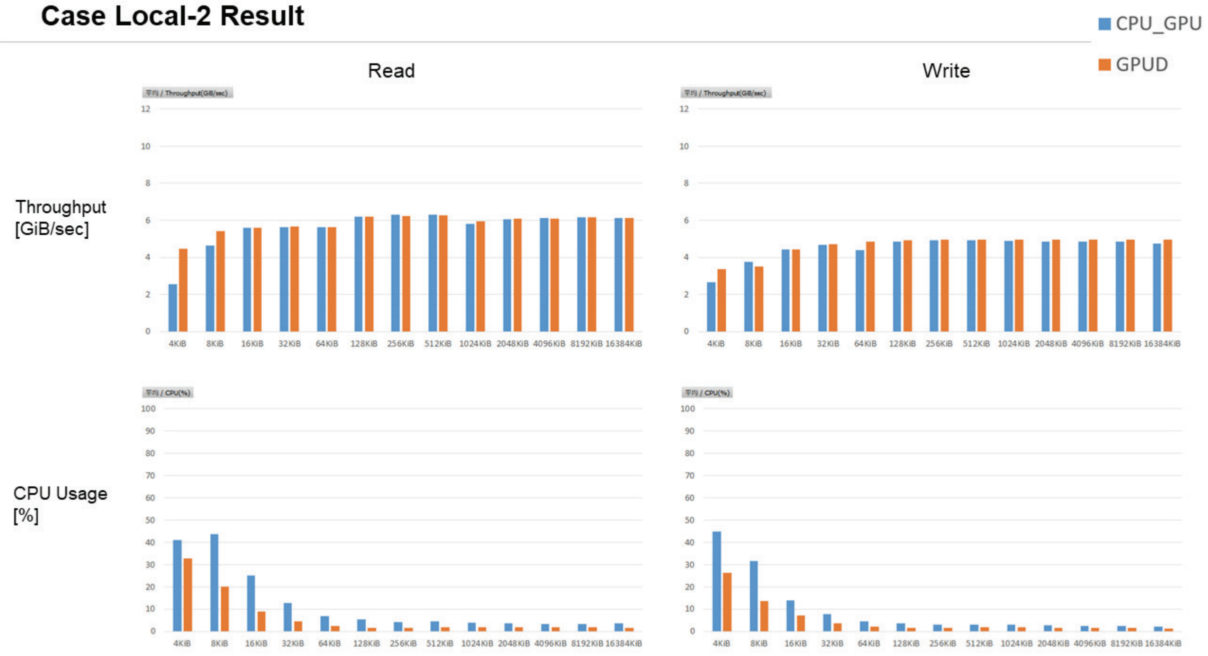
Next, the results of GPUDirect Storage from two local NVMe SSDs to two GPUs are shown.

The throughput is approximately double compared to the previous case of one SSD, NVMe SSD to one GPU. As for CPU utilization, GPUDirect Storage has a lower CPU utilization compared to the previous case.

Case Local-2: 2 Local NVMe SSD and 2 GPU



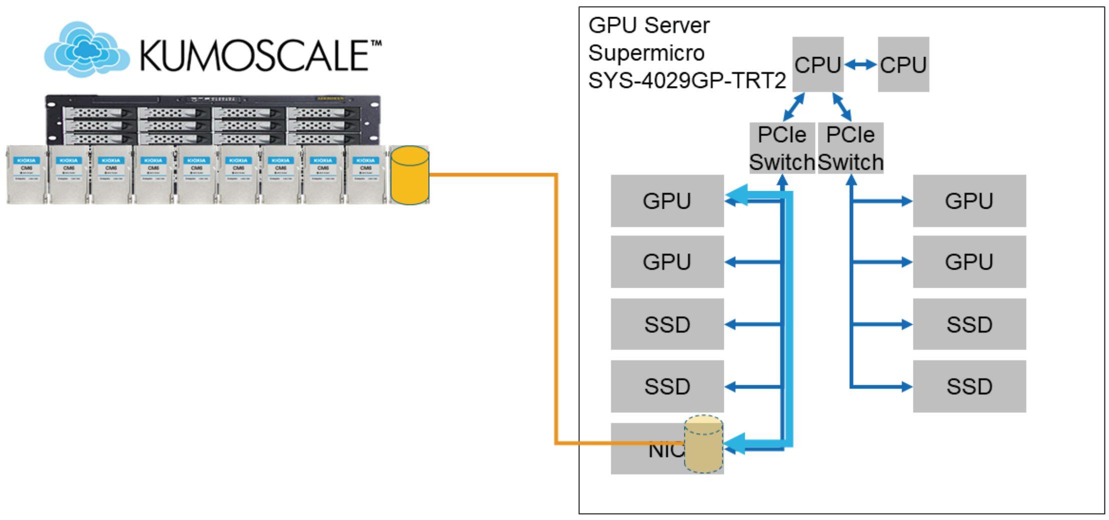
Case Local-2 Result



Now, we move on to the NVMe-oF measurement. As in the first measurement, the configuration here is the same, with GPUDirect Storage from one GPU to one SSD, but data is transferred with KumoScale through the GPU server's NIC.

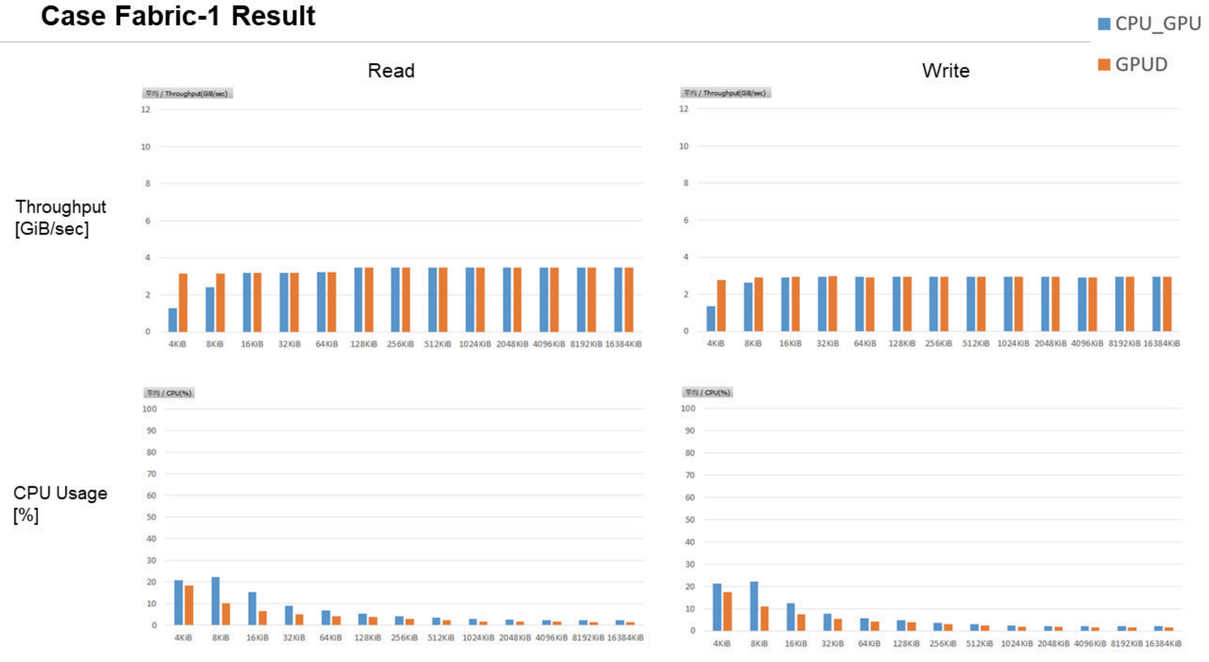
As shown in the previous configuration diagram, the GPU server topology on the right and the NVMe-oF target server running KumoScale on the left are shown. In this configuration, the GPUs and NICs under the same PCIe Switch are designated to run. Similar to the local storage case, we have to note that running GPUDirect Storage specifying the NIC under the left PCIe Switch from the GPU under the right PCIe Switch will not yield the expected performance. In this case, GPU and storage are specified by environment variables and directories as with local storage.

Case Fabric-1: Single NVMe SSD and Single GPU with NVMe-oF



Here are the measurement results of GPUDirect Storage on NVMe-oF. The results are similar to those obtained with a local SSD.
The advantage of GPUDirect Storage lies in the throughput for small block I/Os of 4KiB and in throughput. However, for other aspects, the results are almost the same. As for CPU utilization, the GPUDirect Storage case shows good results, with low CPU resource usage.
However, the difference here is not as great as with local storage. This is probably due to network processing in NVMe-oF, which does not reduce CPU utilization as much as in local storage.

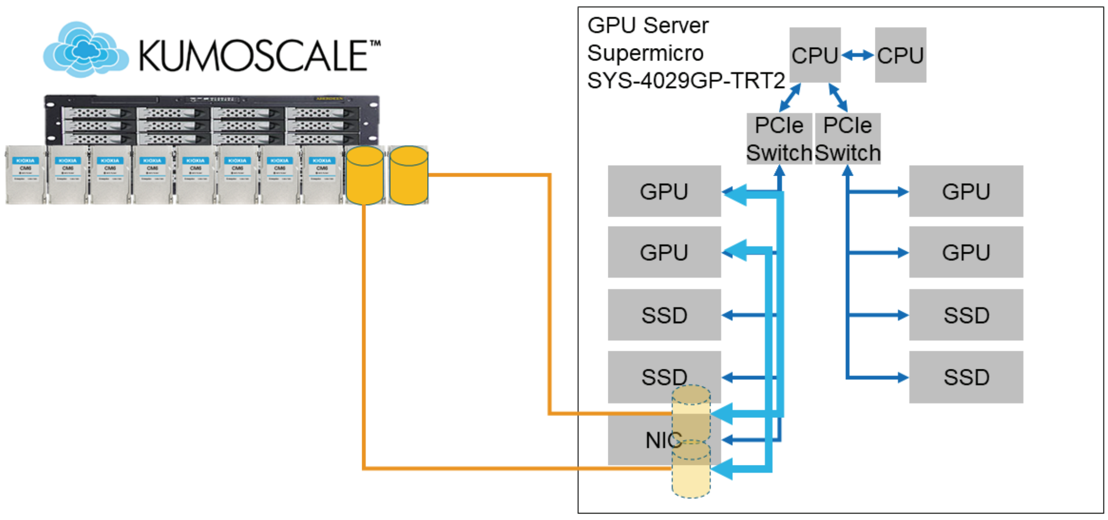
Case Fabric-1 Result



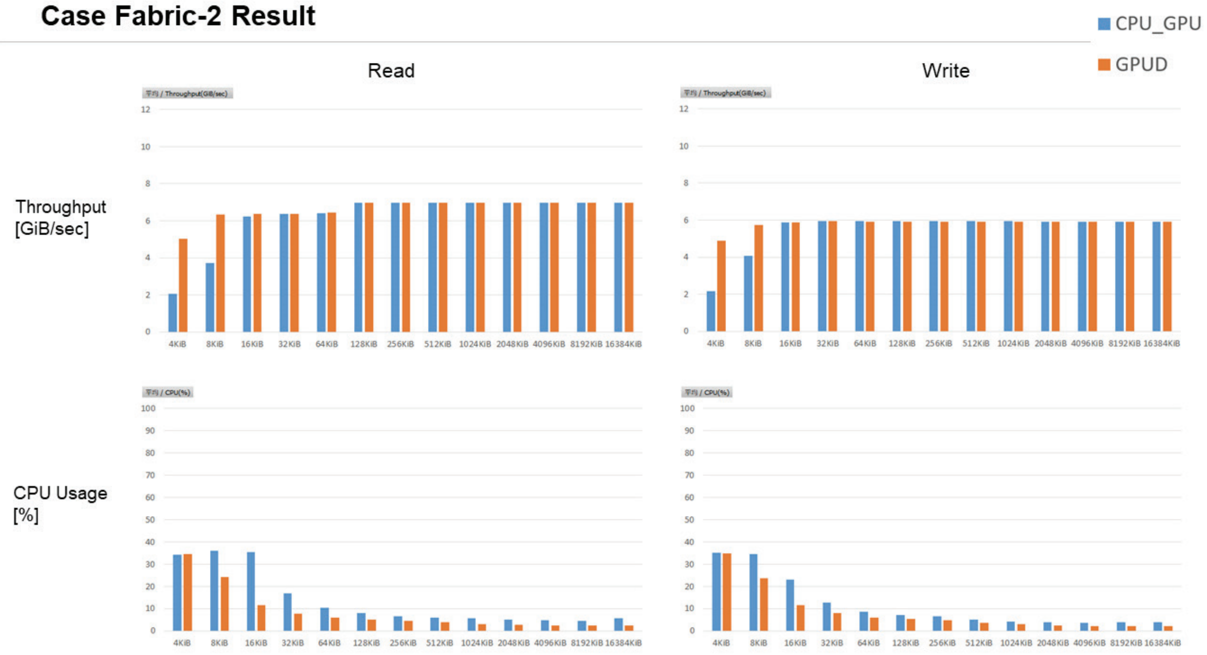
Next is the result of running GPUDirect Storage from two NVMe SSDs to two GPUs with NVMe-oF.

As with the local storage, increasing to two units shows twice as much better performance than with one unit, and GPUDirect Storage appears to have an advantage with regards to CPU utilization.

Case Fabric-2: 2 NVMe SSD and 2 GPU with NVMe-oF

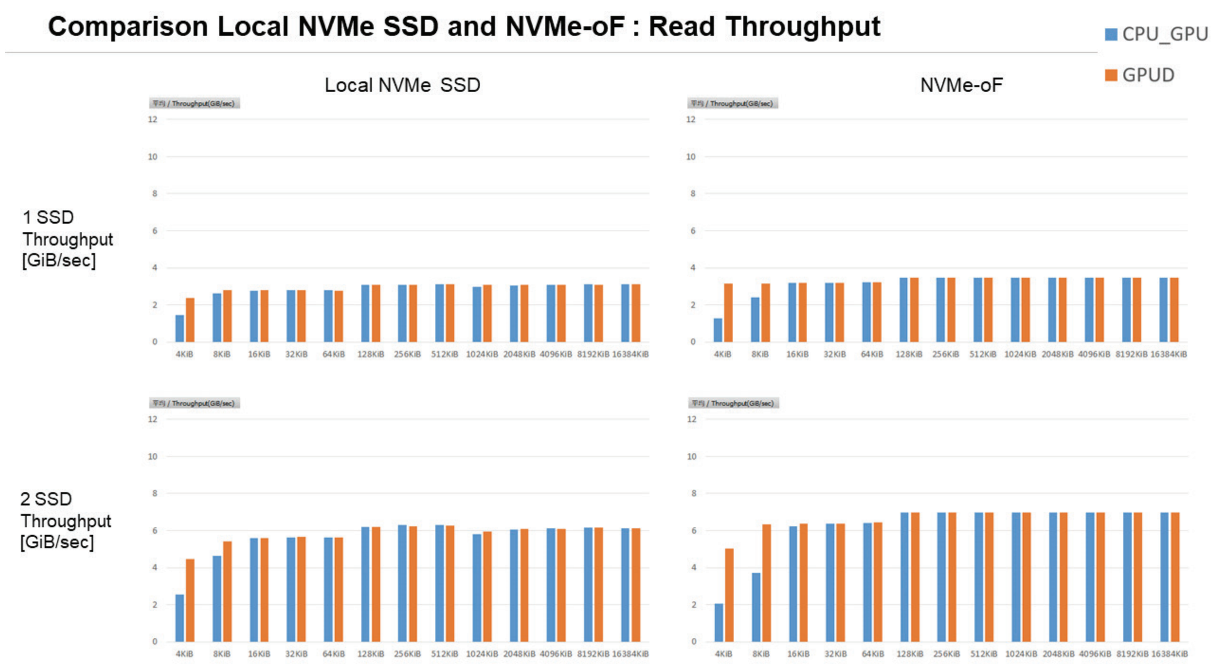


Case Fabric-2 Result

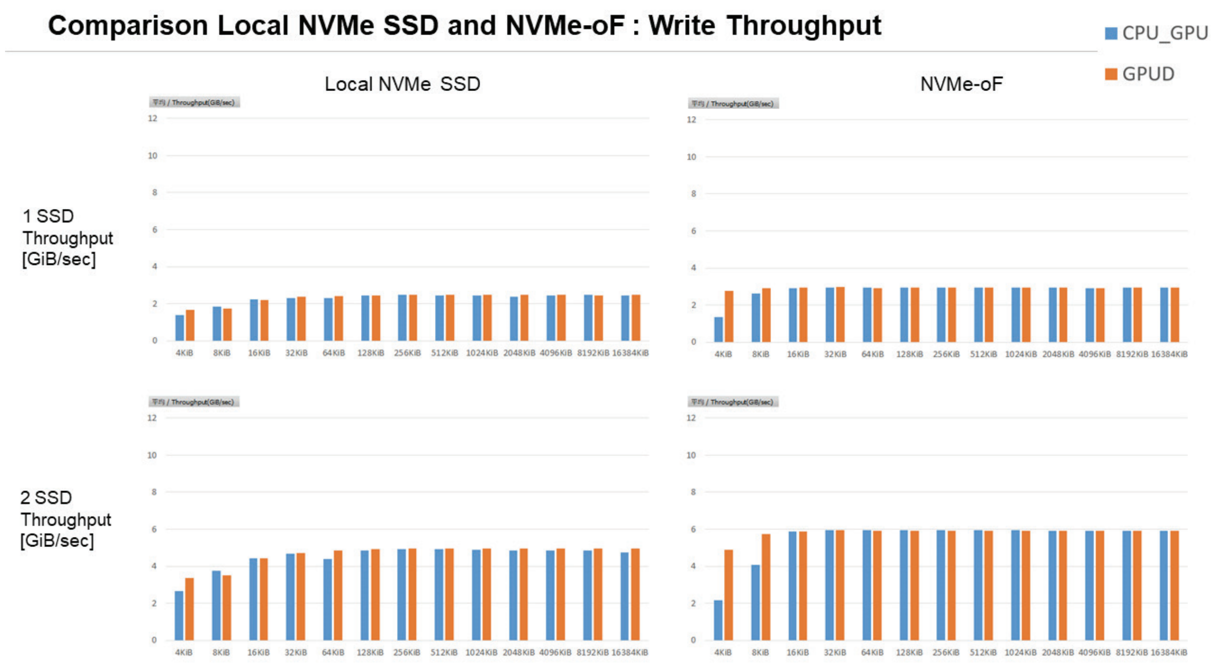


Next, we compare the measurement results between the local NVMe SSD and the NVMe-oF SSD. Please note that both local and NVMe-oF drives are compared using PCIe Gen3 connections.

First, let's take a look at Read throughput comparisons.
The left shows the local and the right shows the NVMe-oF results. The top row shows the results of one SSD and the bottom is with two SSDs. You can see that I/O was performed without much difference between two cases. This is as expected, and shows that KumoScale is capable of providing the same performance as local by aggregating physical storages even though they are provided over the network.

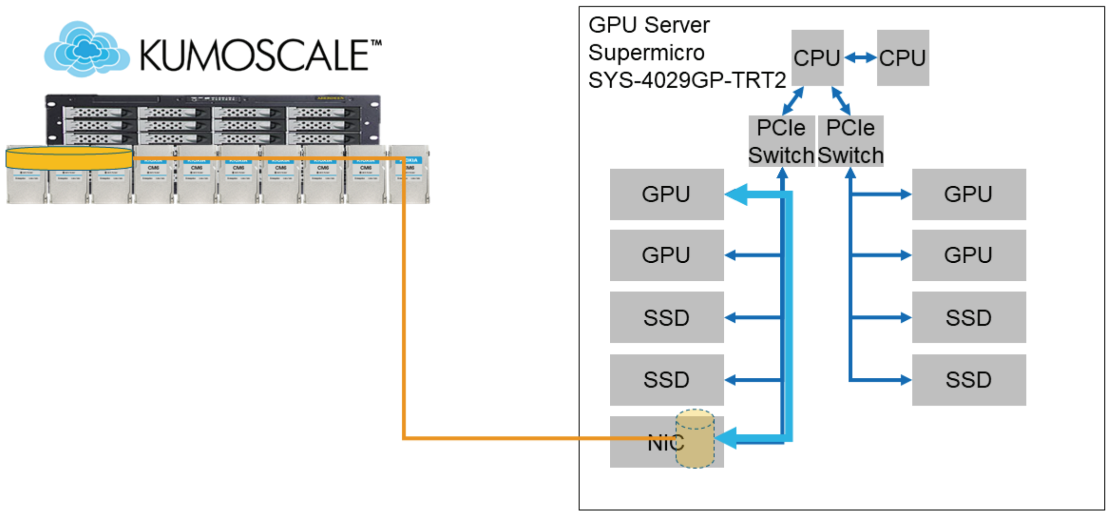


A similar trend is observed regarding the throughput of Write.



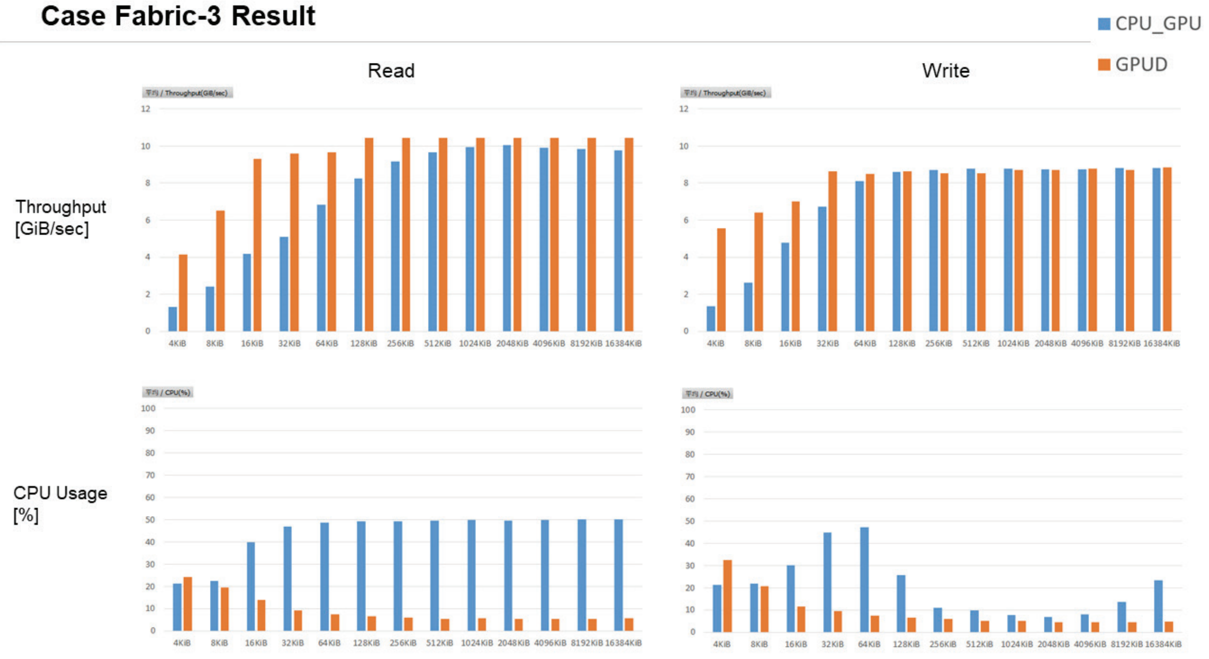
Next, measurement results are presented between a stripped namespace using three SSDs on KumoScale and a single GPU.

Case Fabric-3: 3 NVMe SSD and Single GPU with NVMe-oF



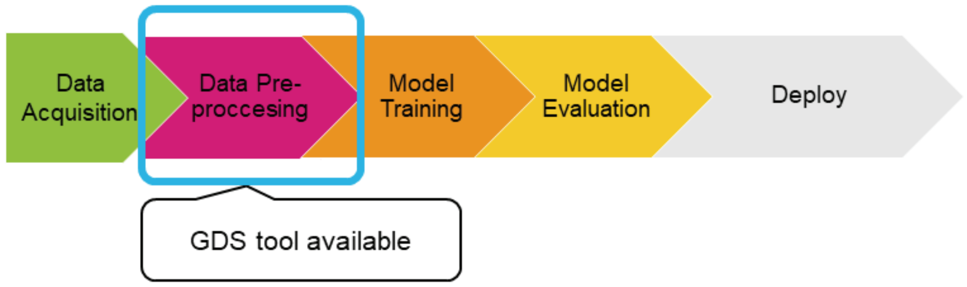
First, the throughput of Read shows larger differences compared to an ordinary I/O through CPU. For throughput of 128KiB or beyond, the bandwidth of 100 Gigabit Ethernet (100GbE) has reached its upper limit, so no higher throughput can be achieved. Regarding Write, significant improvement was observed compared to the single SSD mentioned earlier.
Overall, GPUDirect Storage shows lower CPU utilization except for the 4KiB area, where GPUDirect Storage shows higher CPU utilization.

Case Fabric-3 Result



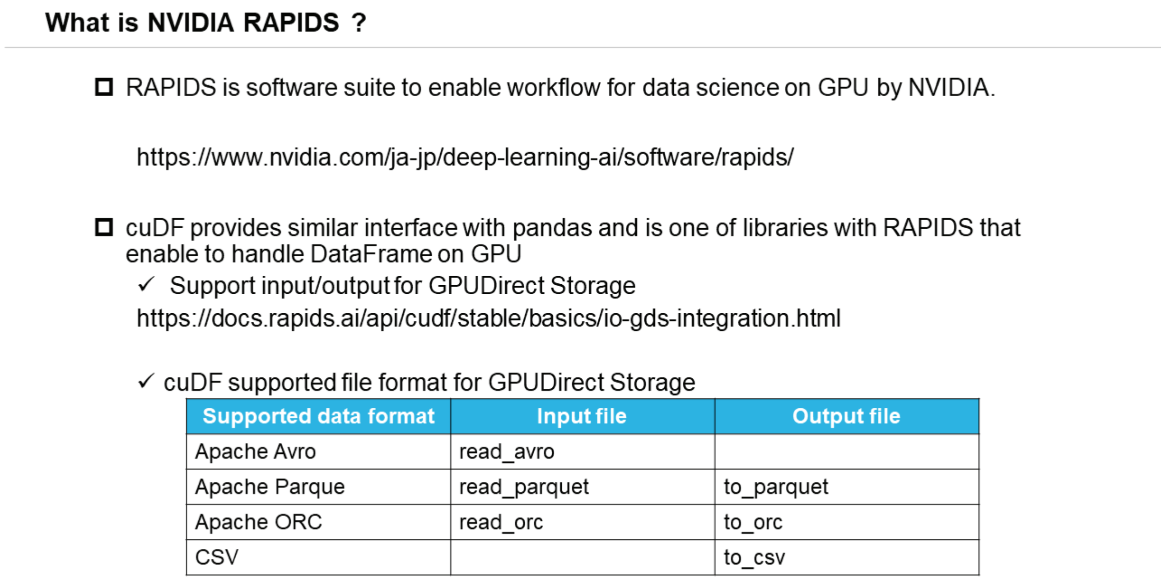
Next, we move on to benchmarking GPUDirect Storage application integration.

In order to highlight GPUDirect Storage effectiveness in machine learning, we focused on preprocessing in machine learning workflows. Benchmarking was conducted on RAPIDS among GPUDirect Storage compatible data analysis libraries.



Among a number of APIs included in RAPIDS, we measured the difference between data input and output using a function called cuDF, which has a pandas-like interface commonly used for data analysis. Note that cuDF supports four data formats, but GPUDirect Storage file input and file output support differs.

*This status is as of July 2022. As development is ongoing , software environment may differ depending on progress from when the author describes.



What is NVIDIA RAPIDS ?

- ☐ RAPIDS is software suite to enable workflow for data science on GPU by NVIDIA.

  https://www.nvidia.com/ja-jp/deep-learning-ai/software/rapids/

- ☐ cuDF provides similar interface with pandas and is one of libraries with RAPIDS that enable to handle DataFrame on GPU
  - ✓ Support input/output for GPUDirect Storage
  https://docs.rapids.ai/api/cudf/stable/basics/io-gds-integration.html

  - ✓ cuDF supported file format for GPUDirect Storage

Supported data format	Input file	Output file
Apache Avro	read_avro	
Apache Parque	read_parquet	to_parquet
Apache ORC	read_orc	to_orc
CSV		to_csv

Versions of pandas and RAPIDS used for benchmarking are shown below. In most cases, but not all, converting from import pandas to cuDF makes it easy to work with GPU as almost the same description and processing can be used.
A simple source code comparison is provided as an example. The blue text represents pandas, and the green shows the changes to cuDF. In this code, two changes made it possible to support GPUs.

Software components for data analysis benchmark

- For CPU
  - ✓ pandas : 1.3.5
  - ✓ NumPy : 1.21.6

- For GPU
  - ✓ RAPIDS (cuDF): 22.04.00-stable
  - ✓ CuPy : 10.4.0

- pandas and RAPIDS comparison
  - ✓Restatement from pandas to RAPIDS(cuDF) is relatively easy
  - ✓When restatement to RAPIDS is done, it is operational on GPUDirect Storage.

pandas	RAPIDS (cuDF)

`import pandas as pd`
`df = pd.read_parquet( “parquet format data")`
`df[“column name"].fillna(False, inplace=True)`
`df.to_csv("csv format output")`
	`import cudf`
	`df = cudf.read_parquet( "parquetformat data")`
	`df[“column name"].fillna(False, inplace=True)`
	`df.to_csv("csv format output")`

Next, here is the data set used for processing this time. For processing, the data in Parquet format is loaded, and after simple processing, integrated and exported in CVS format.
The processing highlighted in green is the input/output processing of data, where GPUDirect Storage is expected to contribute.
Numbers in the table represent processing numbers, which indicate where data is transferred to NVMe, system memory, or GPU memory for processing.

Benchmarking with RAPIDS

- Data used for benchmarking「VTuber 1B: Live Chat and Moderation Statistics」
  - ✓ NLP data set for academia purpose(Natural language processing)
  - ✓ Open Data Commons Public Domain Dedication and License (PDDL) v1.0
  - ✓ Open public and standard version in Kaggle and Github
  - ✓ Data size : 65.63MB ~ 4.77GB in Parquet format

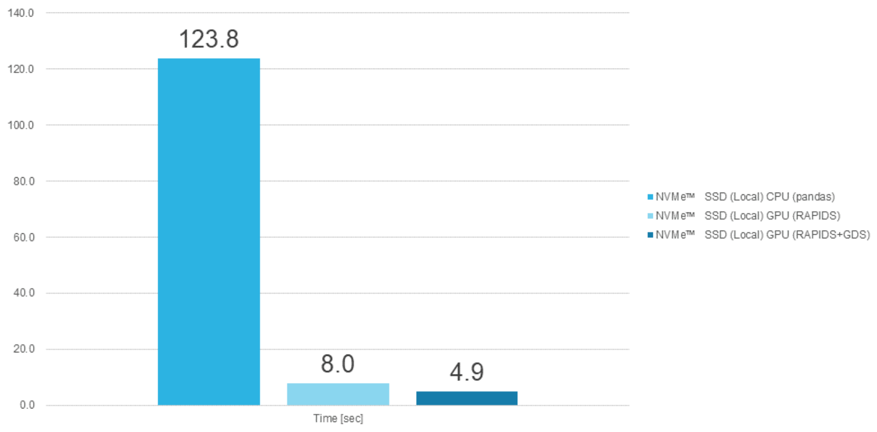
- Processing step
  1. Ingest 2 parquet format data with approx.2.8GB into DataFrame
  2. Insert “True” flag into one DataFrame
  3. Combine two DataFrame
  4. Replacing “false” on empty field on DataFrame
  5. Write DataFrame into CSV format

[Data placement]	NVMe	System Memory	GPU Memory
pandas	1, 5	1, 2, 3, 4, 5	
RAPIDS	1, 5	1, 5	1, 2, 3, 4, 5
RAPIDS + GDS	1, 5		1, 2, 3, 4, 5

Benchmark test results when using NVMe local on GPU servers are shown below. Average runtime per transaction, with lower values representing better results.

Benchmarking with RAPIDS : GPUDirect Storage Result

- RAPIDS gains significant reduction of processing time comparing to CPU based pandas
- Further reduction is achievable with GPUDirect Storage

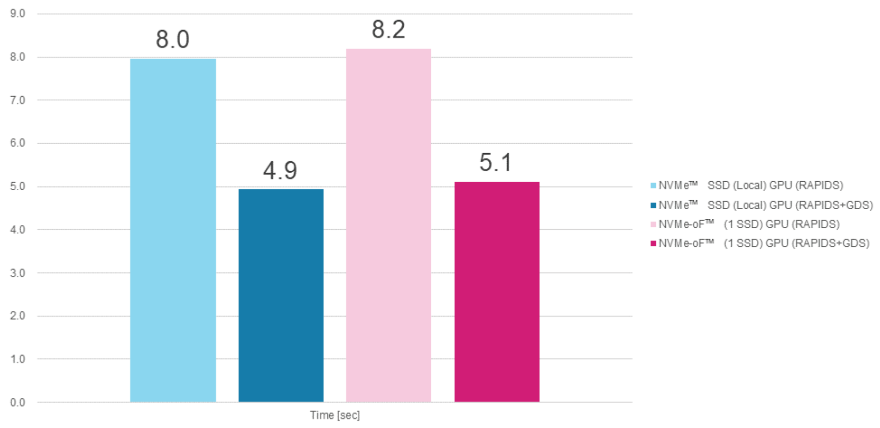


Switching from CPU processing in pandas to GPU processing in RAPIDS greatly improves performance. The middle shows the results with RAPIDS only, and the right shows the results with GPUDirect Storage enabled in addition to RAPIDS. From here, GPUDirect Storage works effectively and yields good results with less value.
The difference in values with and without GPUDirect Storage is small, but this process is repeated many times in data analysis, so the slight difference becomes increasingly large.

Next is a comparison between local NVMe and NVMe-oF. The left is local NVMe and the right is NVMe-oF. As in the results presented in GDS I/O, no significant difference is found, and the comparison results are comparable.

Benchmarking with RAPIDS: NVMe SSD vs NVMe-oF

- RAPIDS and GPUDirect Storage run in minimal overhead with NVMe-oF

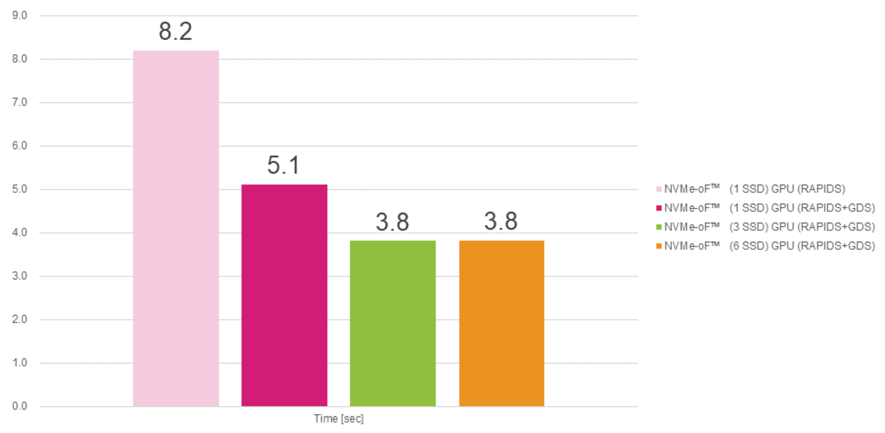


In addition, I/O performance can be improved by stripping NVMe SSDs using KumoScale. This distinctive feature gives higher performance than a single NVMe by increasing the number of physical SSD. As like GDS I/O, we see the same effect in general processing as seen in results by dedicated benchmarking software.

This contribution could be achievable up to the theoretical performance limit of a PCIe slot, and is considered effective when seeking better performance than that of local NVMe SSD only.

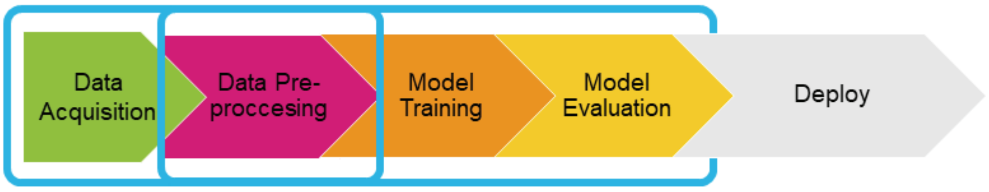
Benchmarking with RAPIDS: Striped Volume with KumoScale

- Striped configuration with 3 SSD could reduce elapsed time against single SSD configuration.
- No gain observed with 6 SSDs due to network bandwidth bottleneck.



This validation was intended to measure the effect of improved processing times for machine learning workflows with the contribution of GPUDirect Storage. We confirmed that the I/O benchmarking tool showed better data transfer performance to GPU and reduced CPU utilization. Furthermore, we confirmed that pre-processing time using conventional CPU can be significantly reduced by utilizing GPUDirect Storage compatible libraries and GPU, which is closer to actual application usage.

We believe that we have demonstrated how the use of high-speed storage can shorten machine learning workflow processing time, as initially envisioned. This verification proved that high-speed storage is no longer too technical to use in a machine learning environment. The future of GPUDirect Storage is expected to bring further time reductions of storage intensive workflow with machine learning as various tools in workflows become GPUDirect Storage compatible in the future.



Let us introduce KIOXIA products that contribute to improve processing time improvements of machine learning workflow as demonstrated above. In addition to NVMe SSDs as local storage, which actually confirmed the effectiveness of GPUDirect Storage in this verification, we also have different NVMe SSD model with lower latency for same purpose. Also KumoScale is an NVMe-oF software product that enables storage disaggregation to improve machine learning performance as verified above.

Furthermore, we have an Ethernet-attached SSD that supports NVMe-oF as a stand-alone SSD. Although it has different construction from KumoScale, these SSDs can provide a scalable NVMe-oF high-speed storage environment. This Ethernet-attached SSD is also GPUDirect Storage compatible.

KIOXIA Products for GPUDirect Storage

KumoScale: https://business.kioxia.com/ja-jp/ssd/kumoscale-software.html
SSD: https://business.kioxia.com/ja-jp/ssd.html

*AMD and EPYC is a trademark of Advanced Micro Devices, Inc.
*Intel and Xeon are trademarks of Intel Corporation or its subsidiaries.
*Linux is a trademark of Linus Torvalds in the U.S. and other countries.
*Supermicro is a trademark or registered trademark of Super Micro Computer, Inc. or its subsidiaries in the United States and other countries.
*Other company names, product names, and service names may be trademarks of their respective companies.