Storage for a New Generation of AI/ML

Presented by
Somnath Roy – Principle Engineer – Samsung Memory Solutions Lab
Current State of AI/ML

• Focus on Large-Scale AI/ML (at least >1PB storage for training data)
  • Large-Scale Use cases:
    • Fraud prevention and risk analysis
    • Natural Language Processing
    • Real-time price optimization
    • Autonomous driving

• Compute has evolved rapidly with new algorithms and GPUs
  • In fact with the advent of GPU direct, NVIDIA is claiming bottleneck is on storage

• Can large-scale storage performance keep up with compute?
  • High read BW requirement (>1TB/s per rack) for running AI training at scale with thousands of GPUs in parallel
DSS: Performant & Scalable Object Storage

**Disaggregated Storage Solution (DSS)**

**Services**
- Samsung developed – open sourced
  - https://github.com/OpenMPDK/DSS
- NVMeoF based S3 Service
- High Read Throughput Object Storage
- Disaggregated Storage and compute
- Shared everything architecture
- Zero copy key-value transfer
- Easy Scaling at Exabytes

**Use Cases**
- Large scale high READ throughput AI training
- Image Analytics
- Audio/Video AI
- Metaverse

---

**Diagram Description**

- **App Client 1** connected to **DSS Object Store -1** and **DSS Object Store -2**
- **App Client N** connected to **DSS Object Store -3** and **DSS Object Store -4**
- **DSS Open-Mpdk** and **NVMeoF KV Driver**
- **NIC 1** - Nvidia ConnectX-5 (100G)
- **NIC 2** - Nvidia ConnectX-5 2 (100G)
- **DSS Target RDMA stack QPS on NIC-1**
- **DSS Target RDMA stack QPS on NIC-2**
- **DSS Target Subsystem -1**
- **DSS Target Subsystem -2**
- **DSS Target SPDK/DPDK layer**
- **PCIe GEN3x4 connections**
- **PM1733a 32TB (V6 TLC), U.2 x 32EA**

---

**Use Cases**

- Large scale high READ throughput AI training
- Image Analytics
- Audio/Video AI
- Metaverse
DSS Enhanced Minio Object-Store

Stock Minio Shared-nothing architecture (Compute has to grow along with storage)

DSS Minio disaggregated, Shared-everything architecture (Compute and storage can grow independently)

Distributed Locking & EC/replication traffic

KV-NVMF
Reference Minio + DSS deployment model for AMD

- VLAN1: 201.100.*
- VLAN2: 202.100.*
- VLAN3: 203.100.*
- VLAN4: 204.100.*
- VLAN5: 205.100.*
- VLAN6: 206.100.*
- VLAN7: 207.100.*

- Dist DSS Minio-1
- Dist DSS Minio-2
- Dist DSS Minio-3
- Dist DSS Minio-4

- DSS KV-API
- DSS KV-API
- DSS KV-API
- DSS KV-API

- NVMeOF-KV RDMA Driver
- NVMeOF-KV RDMA Driver
- NVMeOF-KV RDMA Driver
- NVMeOF-KV RDMA Driver

- Memory Interleaving
- Memory Interleaving
- Memory Interleaving
- Memory Interleaving

- S3 bench client-1: 202.100.*
- S3 bench client-2: 204.100.*
- S3 bench client-3: 206.100.*
- S3 bench client-4: 208.100.*

- KV-pool (SPDK subsystem with 8 drives)
  - (201.100.*), (203.100.*)
  - (205.100.*), (207.100.*)
Bucket Abstraction

Key Distribution

Cluster Expansion

Rebalance

Standard S3 Operations (Get/Put etc.)
DSS GET Performance

Setup:

- Client - 16x Dell PowerEdge R6525, 2 x DGX A100
- DSS S3 Server
  - 10x Dell PowerEdge R7525 Gen4 servers
  - Dual socket AMD EPYC 7742 64-Core
  - 1TB physical memory
  - 4x Mellanox Dual port 100/200Gb (ConnectX-6)
- SSD - 16x PM1733 4TB Gen4 NVMe SSD per DSS S3 server
- Total data set generated during test ~400TB
- Top chart is just DSS backend performance across 10 node, no S3 involved
- Tool used home grown dss test cli
- Bottom one with DSS optimized Minio
- Tool used standard S3-benchmark
AI Benchmarking Tool

- Benchmarking various storage solution based on NFS, S3 at AI training level
- Platform where developers can add their ML framework, custom data set, training method, models and storage backend
- Demo is showing a custom training with a custom data set that is only capturing data load time and BW from storage servers on NFS/S3
DSS S3 vs Standard NFS

Setup:

• Client – 12x Dell PowerEdge 740xd
• DSS S3 Server
  • 6x Dell PowerEdge R7525
  • AMD EPYC 7742 64-Core
  • Mellanox Dual port 200g (ConnectX-6)
• NFS server –
  • 6x Dell PowerEdge R6525
  • AMD EPYC 7742 64-Core
  • Mellanox Dual port 200g (ConnectX-6)
• SSD - PM1733 4TB NVMe SSD
IO flow during S3 GET request

DSS MINIO

- Select EC set
- Get Object Info
- Get object on the EC set
- Read all EC Chunks
- EC Decode
- Copy to http response writer

DSS Host

- KV Get
- Read EC Chunk 1
- Read 1 MB KV chunks
- KV Get
- Read object Meta
- Driver GET ioctl
- DSS KV Get API
- DSS KV Get

DSS Target

- Kernel Driver RDMA GET over NVMF
- SPDK NVMF RDMA configured NQNs

DSS stack IO path is one-copy

Copy to http response writer

Copy of data blocks

Zero copy when no disk failure
IO flow during S3 GET request for next Gen DSS

**End Client**

S3 GET Req → **DSS MINIO**

- Select EC set
- Read Object Info
- Get Object on the EC set
- Read all EC Chunks
- Read EC Chunk N Read

**DSS MINIO**

KV.RDD_Get()

**DSS Host**

- DSS KV Get API
- Driver GET ioctl
- Driver GET ioctl
- Driver GET ioctl

**DSS Target**

SPDK NVMF RDMA configured NQNs

GET Payload is directly RDMA into end-client memory bypassing DSS Minio server stack

Response is only success/failure, no payload back
DSS Availability

Open Source Announcement
- https://github.com/OpenMPDK/DSS
  - https://github.com/OpenMPDK/dss-sdk
  - https://github.com/OpenMPDK/dss-ansible
  - https://github.com/OpenMPDK/dss-minio
  - https://github.com/OpenMPDK/dss-ecosystem

Complete Ecosystem
- AI Benchmarking Framework supporting user preferred training and models
- Client Wrappers supporting Pytorch and Tensorflow
- Host and Target Stack
Thank You
(som.roy@samsung.com)
Please take a moment to rate this session.

Your feedback is important to us.