NetApp Verified Architecture

NetApp EF-Series AI with NVIDIA DGX A100 Systems and BeeGFS

NVA Deployment

Abdel Sadek, Tim Chau, Joe McCormick, and David Arnette, NetApp
March 2021 | NVA-1156-DEPLOY

Abstract

This document describes a NetApp Verified Architecture for machine learning (ML) and artificial intelligence (AI) workloads using NetApp® EF600 NVMe storage systems, the ThinkParQ BeeGFS parallel file system, NVIDIA DGX™ A100 systems, and NVIDIA® Mellanox® Quantum™ QM8700 200Gbps InfiniBand (IB) switches. This document also includes instructions for executing validation benchmark tests after the deployment is complete.

In partnership with

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Executive summary

This document contains validation information for the NetApp EF-Series AI reference architecture for ML and AI workloads. This design was implemented using a [NetApp EF600 all-flash NVMe storage system,](#) the ThinkParQ BeeGFS parallel filesystem, eight DGX A100 systems, and QM8700 switches for both the compute cluster interconnect and storage connectivity. The operation and performance of this system was validated using industry-standard benchmark tools and proven to deliver excellent training performance. Customers can easily and independently scale compute and storage resources from half-rack to multi-rack configurations with predictable performance to meet any machine learning workload requirement.

Program summary

The NetApp Verified Architecture program provides customers with reference configurations and sizing guidance for specific workloads and use cases. These solutions are:

- Thoroughly tested
- Designed to minimize deployment risks
- Designed to accelerate time to market

This document is for NetApp and partner solutions engineers and customer strategic decision makers. It describes the architecture design considerations that were used to determine the specific equipment, cabling, and configurations required to support the validated workload.

NetApp EF-Series AI solution

The NetApp EF-Series AI reference architecture, powered by DGX A100 systems and NetApp cloud-connected storage systems, was developed and verified by NetApp and NVIDIA. It gives IT organizations an architecture that:

- Eliminates design complexities
- Allows independent scaling of compute and storage
- Enables customers to start small and scale seamlessly
- Offers a range of storage options for various performance and cost points
- NetApp EF-Series AI tightly integrates DGX A100 systems, NetApp EF600 NVMe storage systems, and the BeeGFS parallel file systems with state-of-the-art IB networking. NetApp EF600 AI simplifies artificial intelligence deployments by eliminating design complexity and guesswork. Customers can start small and scale seamlessly from science experiments and proof-of-concepts to production and beyond.
- Figure 1 shows several variations in the EF-Series AI family of solutions with DGX A100 systems. The EF600 powered BeeGFS building blocks have been verified with up to eight DGX A100 systems. By adding more of these building blocks, the architecture can scale to multiple racks supporting many DGX A100 systems and petabytes of storage capacity. This approach offers the flexibility to alter compute-to-storage ratios independently based on the size of the data lake, the deep learning (DL) models that are used, and the required performance metrics.

Figure 1) NetApp EF-Series-based BeeGFS building blocks for AI with NVIDIA DGX A100 systems.

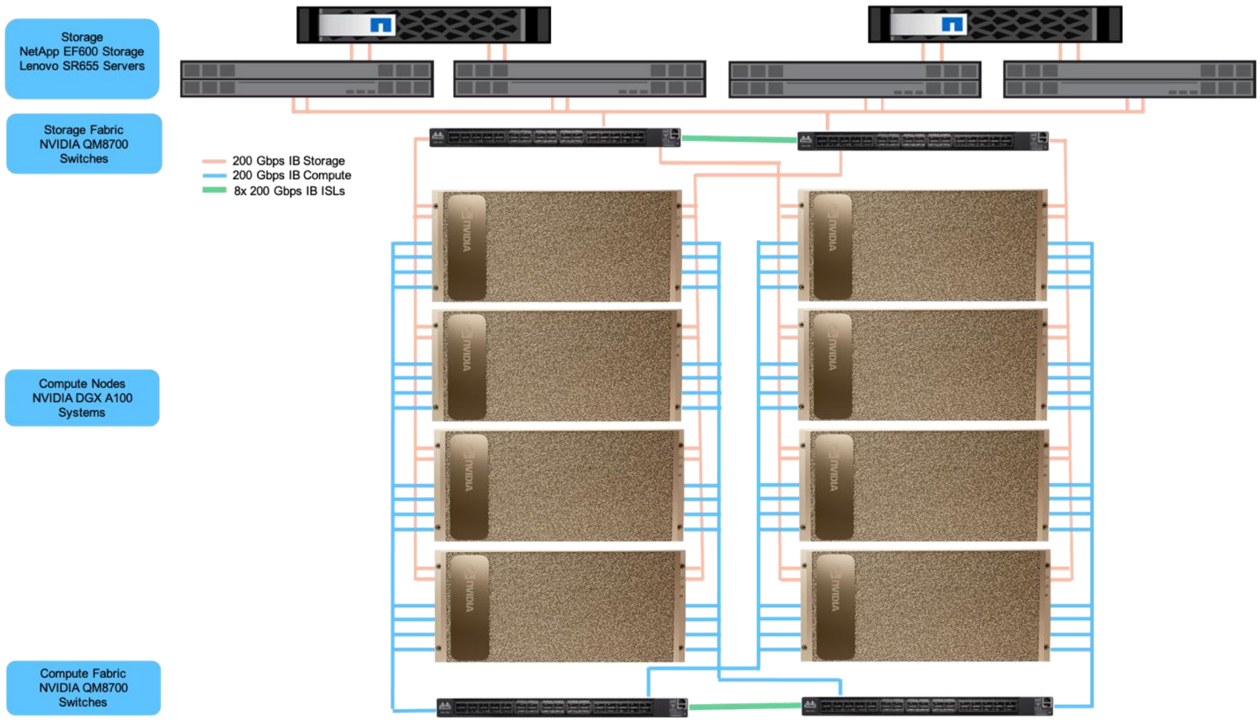
The number of DGX A100 systems and EF600 systems per rack depends on the power and cooling specifications of the rack in use. Final placement of the systems is subject to computational fluid dynamics analysis, airflow management, and data center design.

Solution overview

DL systems leverage algorithms that are computationally intensive and that are uniquely suited to the architecture of GPUs. Computations that are performed in DL algorithms involve an immense volume of matrix multiplications running in parallel. Advances in individual and clustered GPU computing architectures leveraging DGX systems have made them the preferred platform for workloads such as high-performance computing (HPC), DL, video processing, and analytics. Maximizing performance in these environments requires a supporting infrastructure, including storage and networking, that can keep GPUs fed with data. Dataset access must therefore be provided at ultra-low latencies with high bandwidth.

This reference architecture was validated with two BeeGFS building blocks. Each block is comprised of one NetApp EF600 system connected using NVMe/IB to two Lenovo x86 servers running BeeGFS server services. Eight DGX A100 systems running BeeGFS client services connected to two NVIDIA Mellanox QM8700 200Gb IB switches for compute and storage operations. Figure 2 shows the basic solution architecture.

Figure 2) NetApp EF-Series with BeeGFS AI verified architecture.



Use case summary

This solution is intended to support the training and inference phases of the AI and DL pipeline. Depending on the application, DL models work with large amounts of different types of data (both structured and unstructured). This difference imposes a varied set of requirements on the underlying storage system, both in terms of size of the data that is being stored and the number of files in the dataset.

The high-level storage requirements include the following:

- The ability to store and to retrieve millions of files concurrently.
- Storage and retrieval of diverse data objects such as images, audio, video, and time-series data.
- Delivery of high parallel performance at low latencies to meet the GPU processing speeds.
- Seamless data management and data services that span the edge, the core, and the cloud.

For the critical training phase of DL, data is typically copied from the data lake into the training cluster at regular intervals. That data is then processed repeatedly by the DL model to achieve the desired ML proficiency. The servers that are used in this phase use GPUs to parallelize computations, creating a tremendous appetite for data. Meeting the raw I/O bandwidth needs is crucial for maintaining high GPU utilizations.

Technology requirements

This section covers the hardware and software that was used for all the testing described in the “Solution verification” section.

Hardware requirements

Table 1 lists the hardware components that were used to verify this solution. Table 2 lists the specifications for the Lenovo SR655 servers used to run BeeGFS services.

Table 1) Hardware requirements.

Hardware	Quantity
DGX A100 systems	Eight
EF600 storage systems	Two each with two RAID controllers (128Gb memory each) setup in a high-availability (HA) pair with 24x 3.8TB NVMe SSDs in a 2U chassis
QM8700 IB switches	Two for the compute cluster interconnect and BeeGFS
Lenovo SR655 servers	Four running BeeGFS management, metadata, and storage services

Table 2) Lenovo SR655 Server Specifications.

Component	Description
CPU	AMD EPYC 7302 16C 3.0GHz
Memory	8x16GB (128GB total)
PCIe slots	PCIe Gen4 (Riser 1: x16/x16 / Riser 2: x16/x16)

Software requirements

Table 3 lists the software components that were used to validate the solution.

Table 3) Software requirements.

Software	Version
SANtricity OS	11.62.00.9012
BeeGFS	7.2
BeeGFS servers OS	RedHat Enterprise Linux RHEL 7.8
BeeGFS servers OFED	5.1-0.6.6
BeeGFS servers multipathing	Device Mapper MultiPath (DMMP)
IB switch OS	3.9.0606
DGX OS	4.99.10
Docker container platform	19.03.8
Container version	nvcr.io/nvidia/mxnet:20.06-py3 – MLPerf test tensorflow:20.05-tf2-py3 – other tests
OFED version	5.3.0-59
NCCL test version	https://github.com/NVIDIA/nccl-tests/tree/ec1b5e22e618d342698fda659efdd5918da6bd9f
FIO version	3.1

Solution hardware installation and cabling

Hardware installation and basic setup

All hardware components should be installed in the data center racks according to the vendor's recommended guidelines. All components used in the validation of this solution fit into a single rack with room for additional DGX A100 systems. Specific rack power and cooling capacities determine exactly how many servers can be supported in each rack.

Perform basic setup for each component using the appropriate installation documentation. The following configuration procedures assume that all components have been installed and configured for management access and have been upgraded to the software and/or firmware versions recommended in this validation. For specific details about basic installation and setup, see to the appropriate vendor documentation. Links are provided in the deployment procedures section for reference.

Solution cabling

This section contains information about the specific cabling used in the validation of this solution. This cabling configuration can be modified to meet customer-specific implementation requirements.

Note: This document only covers cabling of BeeGFS servers and EF600 storage systems.

Storage controller port connections

Table 4 shows the port connections to the NetApp EF600 storage system and Table 5 shows the port connections to the Lenovo SR655 servers.

Table 4) NetApp EF600 port connections.

Controller name	Port name	Connected device and port
nv-c1-a	e0M	management switch
nv-c1-a	2a	nv-h1:ib0
nv-c1-a	2b	nv-h2:ib0
nv-c1-b	e0M	management switch
nv-c1-b	2a	nv-h1:ib2
nv-c1-b	2b	nv-h2:ib2
nv-c2-a	e0M	management switch
nv-c2-a	2a	nv-h3:ib0
nv-c2-a	2b	nv-h4:ib0
nv-c2-b	e0M	management switch
nv-c2-b	2a	nv-h3:ib2
nv-c2-b	2b	nv-h4:ib2

Table 5) Lenovo SR655 port connections.

Host name	Port name	Connected device and port
nv-h1	eth0	management switch
nv-h1	ib0	nv-c1-a:2a
nv-h1	ib1	DGX A100 switch (any port).
nv-h1	ib2	nv-c1-b:2a
nv-h1	Ib3	DGX A100 switch (any port).
nv-h2	eth0	management switch
nv-h2	ib0	nv-c1-a:2b
nv-h2	ib1	DGX A100 switch (any port).
nv-h2	ib2	nv-c1-b:2b
nv-h2	Ib3	DGX A100 switch (any port).

Host name	Port name	Connected device and port
nv-h3	eth0	management switch
nv-h3	ib0	nv-c2-a:2a
nv-h3	ib1	DGX A100 switch (any port).
nv-h3	ib2	nv-c2-b:2a
nv-h3	Ib3	DGX A100 switch (any port).
nv-h4	eth0	management switch
nv-h4	ib0	nv-c2-a:2b
nv-h4	ib1	DGX A100 switch (any port).
nv-h4	ib2	nv-c2-b:2b
nv-h4	Ib3	DGX A100 switch (any port).

Deployment procedures

Deploying NetApp storage systems for use with DGX A100 systems involves the following tasks:

- EF600 storage system deployment and configuration

  Note: For more information, see NVA-1156-DESIGN: NetApp EF-Series AI with NVIDIA DGX A100 Systems and BeeGFS Design Guide.

- BeeGFS deployment and configuration
- IB switch configuration

  Note: For more information, see NVA-1156-DESIGN: NetApp EF-Series AI with NVIDIA DGX A100 Systems and BeeGFS Design Guide.

- DGX A100 configuration

BeeGFS deployment and configuration

Detailed instructions for the installation and basic setup of NetApp E-Series storage systems with BeeGFS are described in TR-4755: BeeGFS with NetApp E-Series. This section captures any additional tuning or optimizations that you should perform to create the configuration tested in this solution validation.

Lenovo SR655 server configuration

The following BIOS settings were applied to maximize performance:

- NUMA nodes per socket = NPS1
- Determinism Control = Manual
- Determinism Slider = Performance
- SVM Mode = Disabled
- IOMMU = Disabled
- DF Cstates = Disabled
- SMT = Disabled

BeeGFS server service configuration

Each Lenovo SR655 server is configured to run multiple BeeGFS services to maximize utilization of the EF600 storage systems and 200Gbps IB interfaces and reduce rack footprint. Each server runs two

BeeGFS storage services and two BeeGFS metadata services. One server also runs the lightweight BeeGFS management service.

Note: Running multiple BeeGFS services on the same server requires using [Multi Mode](#).

BeeGFS services are configured using files in `/etc/beegfs/` corresponding to each running service. The following changes to the defaults were found to have the best performance.

Metadata service configuration

- connMaxInternodeNum = 32 > 128
- TuneNumWorkers = 0 > 32
- TuneTargetChooser = randomized > roundrobin

Storage service configuration

- connMaxInternodeNum = 12 > 128
- TuneNumStreamListeners = 1 > 2
- TuneNumWorkers = 12 > 14
- TuneUseAggressiveStreamPoll = False > True

Common service configuration

To balance the services across available IB ports, the BeeGFS `connInterfacesFile` parameter was used to set services with an odd BeeGFS node ID to prefer ib1 then ib3, and services with even numbered node IDs to prefer ib3, then ib1. As we are running multiple services on the same node, unique TCP and UDP ports were set using the `connMetaPortTCP` and `connMetaPortUDP` parameters.

To optimize BeeGFS target performance, the following settings were applied to the device mapper and underlying NVMe devices representing the corresponding E-Series volumes in Linux:

- max_sectors_kb = 512
- read_ahead_kb = 4096
- nvme_scheduler = "none"
- dm_scheduler = "noop"
- nr_requests = 1024
- nomerges = 2
- rqaffinity = 2

These settings are located under `/sys/block/dm-X/queue/ or /sys/block/nvmeXnY/queue/` and can be modified by using `echo` (for example, `echo 512 > /sys/block/dm-1/max_sectors_kb`). In order for changes to persist through reboot, add the appropriate settings to `rc.local` or configure a udev rule (recommended).

DGX A100 system configuration

Complete installation and configuration of DGX A100 systems is outside the scope of this document. Two 200Gbps IB ports were configured on each node for connectivity to BeeGFS. The following sections document the setup required to mount BeeGFS to the DGX A100 systems.

BeeGFS client installation and configuration

Mounting BeeGFS to the DGX A100 systems requires installing the BeeGFS client package by using the apt package manager. The BeeGFS client automatically handles the compiling of the BeeGFS kernel module that provides a normal mountpoint, which ensures applications can access BeeGFS without modification. This module can be installed on all supported Linux kernels without the need for patches.

Before you start the BeeGFS client, make the following minor changes:

1. Update the `buildArgs` line in `/etc/beegfs/beegfs-client-autobuild.conf` to ensure the client module is built with support for remote direct memory access (RDMA). If the OpenFabrics Enterprise Distribution (OFED) kernel modules are in use, you must supply the path to the OFED kernel modules. For the OFED version used on the DGX A100 systems, the following configuration is a result:

```
buildArgs=-j8 BEEGFS_OPENTK_IBVERBS=1 OFED_INCLUDE_PATH /usr/src/ofa_kernel/5.3.0-59-
generic/include/
```

2. In `/etc/beegfs/beegfs-client.conf`, set `sysMgmtdHost` to the IP address of the server running the BeeGFS management service. The following parameters are overridden:
   - `connMaxInternodeNum = 12 > 128`
   - `connRDMABufNum = 70 > 32`
   - `connRDMABufSize = 8192 > 65536`
3. Start or enable BeeGFS `beegfs-helperd` and `beegfs-client services` by using `systemctl`:

```
systemctl start beegfs-helperd beegfs-client
systemctl enable beegfs-helperd beegfs-client
```

The BeeGFS file system is now accessible at the default `/mnt/beegfs` mount point.

Optimizing performance with BeeGFS

One of the benefits of BeeGFS is the ability to optimize for multiple types of workloads that might need to share the same global file system. Specifically, BeeGFS allows configuration of the "stripe pattern" used to split each file to be written/read in parallel across all of the BeeGFS storage services and targets (in this instance, NetApp volumes). According to the BeeGFS Documentation website:

> Striping in BeeGFS can be configured on a per-directory and per-file basis. Each directory has a specific stripe pattern configuration, which will be derived to new subdirectories and applied to any file created inside a directory. There are currently two basic parameters that can be configured for stripe patterns: the desired number of storage targets for each file and the chunk size (or block size) for each file stripe.

Striping can be easily applied and updated post deployment, therefore, different patterns were used to optimize each benchmark test. For more information about the stripe pattern used for each test ID, see NVA-1156-DESIGN: NetApp EF-Series AI with NVIDIA DGX A100 Systems and BeeGFS Design Guide. To configure striping in BeeGFS, run the following command:

```
beegfs-ctl -setpattern -numtargets=16 -chunksize=2m /mnt/beegfs
```

Solution verification

This solution was validated by using both synthetic storage benchmarks and MLPerf v0.7 training benchmark. For more information about the validation test results, see NVA-1156-DESIGN: NetApp EF-Series AI with NVIDIA DGX A100 Systems and BeeGFS Design Guide.

FIO storage throughput benchmark test

To perform the FIO throughput benchmark test, run the following commands to create the test files and perform I/O to them:

```
/usr/bin/fio --create_only=1 --rw=write --direct=1 --ioengine=posixaio -- iodepth=32 --
create_serialize=0 --fallocate=none --group_reporting=1 -- disable_lat=1 --disable_clat=1 --
disable_slat=1 --startdelay=5 --ramp_time=3 --runtime=180 --time_based=1 --invalidate=1 --
blocksize=1024k -- size=4194304k --numjobs=120 --
```

```
directory=/mnt/fs_under_test/fiodir.20200629205829 /usr/bin/fio --rw=write --direct=1 --
ioengine=posixaio --iodepth=32 -- create_serialize=0 --fallocate=none --group_reporting=1 --
disable_lat=1 -- disable_clat=1 --disable_slat=1 --startdelay=5 --ramp_time=3 --runtime=180 - -
time_based=1 --invalidate=1 --blocksize=1024k --size=4194304k --numjobs=120 --
directory=/mnt/fs_under_test/fiodir.20200629205829 /usr/bin/fio --rw=read --direct=1 --
ioengine=posixaio --iodepth=32 -- create_serialize=0 --fallocate=none --group_reporting=1 --
disable_lat=1 -- disable_clat=1 --disable_slat=1 --startdelay=5 --ramp_time=3 --runtime=180 - -
time_based=1 --invalidate=1 --blocksize=1024k --size=4194304k --numjobs=120 --
directory=/mnt/fs_under_test/fiodir.20200629205829
```

FIO storage IOPS benchmark test

To perform the FIO IOPS benchmark test, run the following commands:

```
/usr/bin/fio --create_only=1 --rw=write --direct=1 --ioengine=posixaio -- iodepth=32 --
create_serialize=0 --fallocate=none --group_reporting=1 -- disable_lat=1 --disable_clat=1 --
disable_slat=1 --startdelay=5 --ramp_time=3 --runtime=180 --time_based=1 --invalidate=1 --
blocksize=1024k -- size=4194304k --numjobs=180 --
directory=/mnt/fs_under_test/fiodir.20200629213502 /usr/bin/fio --rw=write --direct=1 --
ioengine=posixaio --iodepth=32 -- create_serialize=0 --fallocate=none --group_reporting=1 --
disable_lat=1 -- disable_clat=1 --disable_slat=1 --startdelay=5 --ramp_time=3 --runtime=180 - -
time_based=1 --invalidate=1 --blocksize=4k --size=4194304k --numjobs=180 --
directory=/mnt/fs_under_test/fiodir.20200629213502 /usr/bin/fio --rw=read --direct=1 --
ioengine=posixaio --iodepth=32 -- create_serialize=0 --fallocate=none --group_reporting=1 --
disable_lat=1 -- disable_clat=1 --disable_slat=1 --startdelay=5 --ramp_time=3 --runtime=180 - -
time_based=1 --invalidate=1 --blocksize=4k --size=4194304k --numjobs=180 --
directory=/mnt/fs_under_test/fiodir.20200629213502
```

MLPerf Training v0.7 benchmark test

This test was performed according to the configuration defined in the MLPerf Training v0.7 benchmark specifications. Configuration of this benchmark is outside the scope of this document. For more information about using MLPerf to perform this and other DL benchmark tests, see the MLPerf training overview.

Conclusion

This document provides the storage implementation details for a high-performance ML/DL training infrastructure using NetApp EF600 storage systems, the BeeGFS parallel file system, and DGX A100 systems. The DGX A100 system is a next-generation DL platform that requires equally advanced storage and data management capabilities. By combining DGX A100 with BeeGFS and EF600 storage systems, this verified architecture can be implemented at almost any scale, from a single DGX A100 system paired with a single BeeGFS building block, up to potentially 120 DGX A100 systems with a scalable number of BeeGFS building blocks presenting a single storage namespace. Combined with the superior cloud integration and software-defined capabilities of the NetApp product portfolio, NetApp storage solutions enable a full range of data pipelines that span the edge, the core, and the cloud for successful DL projects.

Where to find additional information

To learn more about the information that is described in this document, review the following documents and/or websites:

NetApp EF-Series systems

- NVA-1156-DESIGN: NetApp EF-Series AI with NVIDIA DGX A100 Systems and BeeGFS Design Guide:
https://www.netapp.com/pdf.html?item=/media/25445-nva-1156-design.pdf

- NetApp EF-Series product page
https://www.netapp.com/us/products/storage-systems/all-flash-array/ef-series.aspx
- EF600 datasheet
https://www.netapp.com/pdf.html?item=/media/19339-DS-4082.pdf
- NetApp AI and HPC solutions
https://www.netapp.com/artificial-intelligence/high-performance-computing/
- BeeGFS with NetApp E-Series Solution Deployment
https://www.netapp.com/pdf.html?item=/media/17132-tr4755pdf.pdf

NetApp Interoperability Matrix

- NetApp Interoperability Matrix Tool
http://support.netapp.com/matrix

NVIDIA DGX A100 systems

- NVIDIA DGX A100 systems
https://www.nvidia.com/en-us/data-center/dgx-a100/
- NVIDIA A100 Tensor core GPU
https://www.nvidia.com/en-us/data-center/a100/
- NVIDIA GPU Cloud
https://www.nvidia.com/en-us/gpu-cloud/

NVIDIA Mellanox networking

- NVIDIA Mellanox Quantum QM8700 series IB switches
https://www.nvidia.com/en-us/networking/infiniband/qm8700/

Machine learning frameworks

- TensorFlow: An Open-Source Machine Learning Framework for Everyone
https://www.tensorflow.org/
- Horovod: Uber's Open-Source Distributed Deep Learning Framework for TensorFlow
https://eng.uber.com/horovod/
- Enabling GPUs in the Container Runtime Ecosystem
https://devblogs.nvidia.com/gpu-containers-runtime/

Dataset and benchmarks

- ImageNet
http://www.image-net.org/
- MLPerf training and inference benchmarks
https://mlperf.org/

Refer to the Interoperability Matrix Tool (IMT) on the NetApp Support site to validate that the exact product and feature versions described in this document are supported for your specific environment. The NetApp IMT defines the product components and versions that can be used to construct configurations that are supported by NetApp. Specific results depend on each customer's installation in accordance with published specifications.

NVA-1156-DEPLOY-0321

NetApp