NetApp

NetApp Verified Architecture

NetApp ONTAP AI with NVIDIA DGX A100 Systems and Mellanox Spectrum Ethernet Switches
NVA Deployment

David Arnette, NetApp
November 2020 | NVA-1153-DEPLOY

Abstract

This document includes storage-system deployment instructions for a NetApp Verified Architecture for machine learning (ML) and artificial intelligence (AI) workloads using NetApp® AFF A800 storage systems, NVIDIA DGX™ A100 systems, and NVIDIA® Mellanox® Spectrum™ SN3700V 200Gb Ethernet switches. It also includes instructions for executing validation benchmark tests after deployment is complete.

TABLE OF CONTENTS

NetApp ONTAP AI with NVIDIA DGX A100
Systems and Mellanox Spectrum Ethernet
Switches

Executive summary

This document contains validation information for the NetApp ONTAP® AI reference architecture for machine learning (ML) and artificial intelligence (AI) workloads. This design was implemented using a NetApp AFF A800 all-flash storage system, eight DGX A100 systems, and SN3700V switches. The operation and performance of this system was validated using industry-standard benchmark tools. Based on the validation testing results, this architecture delivers excellent training performance. Customers can also 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 ONTAP AI solution

The NetApp ONTAP 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 ONTAP AI tightly integrates DGX A100 systems and NetApp AFF A800 storage systems with state-of-the-art networking. NetApp ONTAP AI simplifies artificial intelligence deployments by eliminating design complexity and guesswork. Customers can start small and grow nondisruptively while intelligently managing data from the edge to the core to the cloud and back.

Figure 1 shows several variations in the ONTAP AI family of solutions with DGX A100 systems. The AFF A800 system performance has currently been verified with up to eight DGX A100 systems. By adding storage controller pairs to the ONTAP cluster, the architecture can scale to multiple racks to support many DGX A100 systems and petabytes of storage capacity with linear performance. 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 ONTAP AI family with NVIDIA DGX A100 systems.**

The number of DGX A100 systems and AFF 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

Deep learning 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 one NetApp AFF A800 system, eight DGX A100 systems, two NVIDIA Mellanox Spectrum SN3700V 200Gb Ethernet switches for the compute fabric, and two more SN3700V switches for storage and client access. Figure 2 shows the basic solution architecture.

NetApp ONTAP AI with NVIDIA DGX A100 Systems and Mellanox Spectrum Ethernet Switches

Figure 2) NetApp ONTAP AI verified architecture.
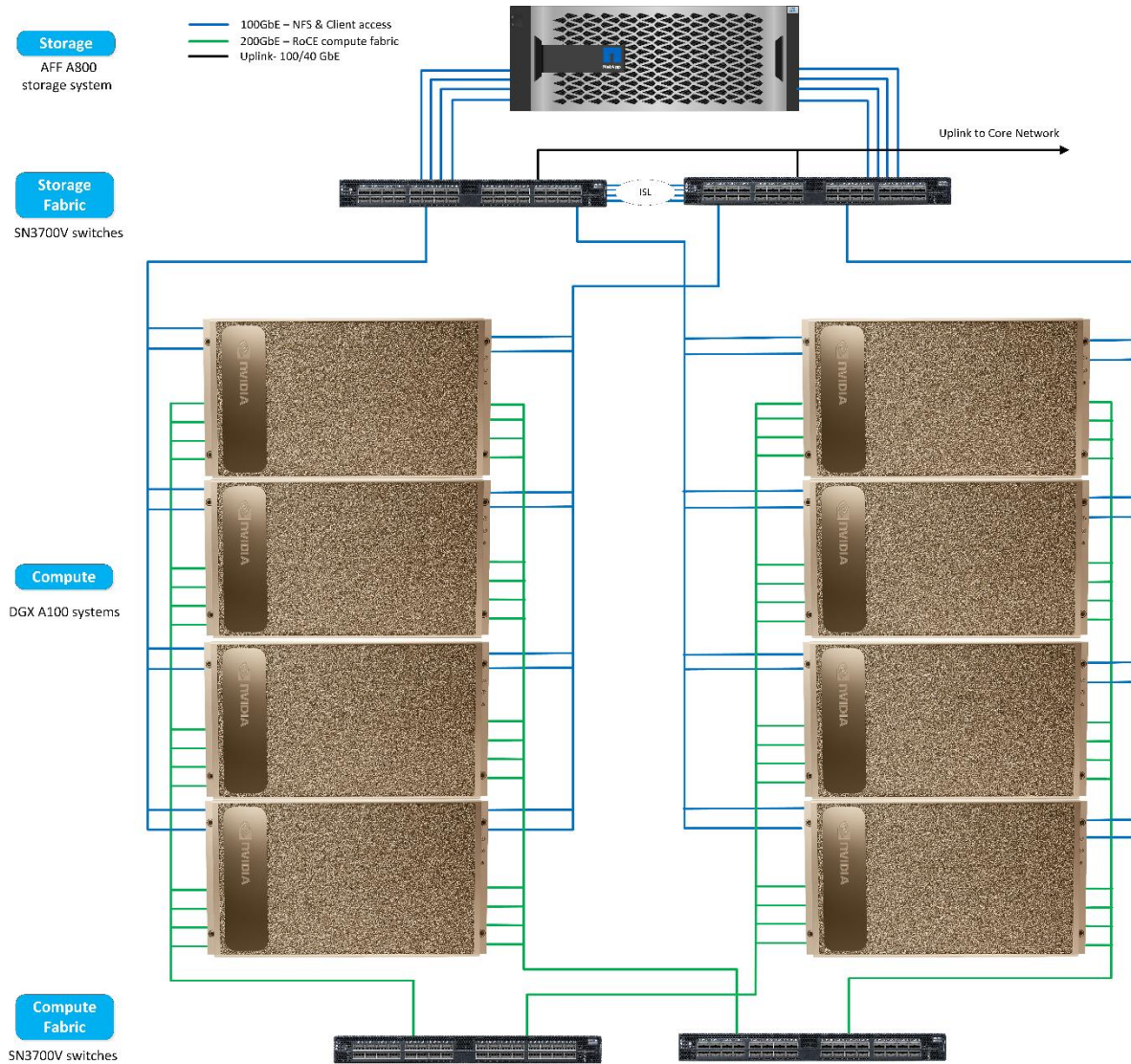
Use case summary

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 machine

NetApp ONTAP AI with NVIDIA DGX A100 Systems and Mellanox Spectrum Ethernet Switches

learning 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 below.

Hardware requirements

Table 1 lists the hardware components that were used to verify this solution.

Table 1) Hardware requirements.

Hardware	Quantity
DGX A100 systems	8
AFF A800 storage system	1 high-availability (HA) pair, includes 48x 1.92TB NVMe SSDs
SN3700V ethernet switches	2 for compute cluster interconnect
	2 for storage, client access and out-of-band management

Software requirements

Table 2 lists the software components that were used to validate the solution.

Table 2) Software requirements.

Software	Version
ONTAP storage OS	9.7P6
Network switch OS (all)	Cumulus Linux 4.2.1
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	5.0-2.1.8
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 datacenter 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 on 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 might be modified to meet customer-specific implementation requirements.

Storage system cabling for this solution is shown in Table 3.

> **Note:** This document only covers cabling of storage system components.

Storage controller port connections

Table 3 shows the ports that are connected to the NetApp AFF A800 storage system.

Table 3) NetApp AFF A800 port connections.

Controller Name	Port Name	Connected Device and Port
ontap-01	e0M	management switch
ontap-01	e0a	ontap-02:e0a
ontap-01	e1a	ontap-02:e1a
ontap-01	e3a	switch-01: <any available>
ontap-01	e3b	switch-02: <any available>
ontap-01	e5a	switch-01: <any available>
ontap-01	e5b	switch-02: <any available>
ontap-02	e0M	management switch
ontap-02	e0a	ontap-01:e0a
ontap-02	e1a	ontap-01:e1a
ontap-02	e3a	switch-01: <any available>
ontap-02	e3b	switch-02: <any available>
ontap-02	e5a	switch-01: <any available>
ontap-02	e5b	switch-02: <any available>

Deployment procedures

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

- Storage system deployment and configuration
- Optimized configuration for single-file workloads
- Network switch configuration
- Host configuration

Storage system deployment and configuration

Detailed instructions for installation and basic setup of NetApp AFF A800 storage systems can be found [here](#).

NetApp ONTAP AI with NVIDIA DGX A100 Systems and Mellanox Spectrum Ethernet Switches

After storage system setup is complete and the system is cabled according the cabling guidelines above, the following storage configuration procedures create the configuration as tested in this solution validation.

Validate and install licenses

This solution requires licenses for the NFS protocol. This license is included in the Premium software bundle included with all AFF storage systems. To confirm that the licenses have been installed, run the following command:

```
license show
  (system license show)

Serial Number: 1-80-000011
Owner: ontap
Package          Type      Description           Expiration
---------------- --------- --------------------- -------------------
Base             site      Cluster Base License  -
NFS              site      NFS License           -
FlexClone        site      FlexClone License     -
3 entries were displayed.
```

If necessary, additional licenses can be added using the `license add` command:

```
license add <license key>
```

Manage the default broadcast domain

Broadcast domains are used to group L2-adjacent ports together for failover purposes. Logical interfaces (LIFs) automatically fail over to another port in the same broadcast domain in the event of a link failure. LIFs attempt to fail over to another port on the same controller if possible, or to an appropriate port on another controller if necessary. Because each VLAN represents a separate L2 subnet, a storage broadcast domain is created for each VLAN.

By default, all network ports are included in the default broadcast domain. Network ports used for data services (e3a, e3b, e5a, and e5b) should be removed from the default broadcast domain, leaving just the management network ports (e0M). To perform this task, run the following commands:

```
broadcast-domain remove-ports -broadcast-domain Default -ports ontap-01:e3a,ontap-01:e3b,ontap-
01:e5a,ontap-01:e5b,ontap-02:e3a,ontap-02:e3b,ontap-02:e5a,ontap-02:e5b
```

Create data aggregates

This solution was validated using the default aggregate configuration for the AFF A800 in which each controller hosts a single data aggregate created using 47 disk partitions. If the data aggregates were not created during initial cluster setup, run the following commands to create two data aggregates:

```
aggr create -aggregate n01_data01 -node ontap-01 -diskcount 47
aggr create -aggregate n02_data02 -node ontap-02 -diskcount 47
```

Note that ONTAP partitions each SSD into two small root partitions and two larger data partitions, with one of each allocated to each controller. The root partitions are used for the root aggregate for the controller node, and the larger partitions are used for data aggregates. Root aggregates are created automatically during initial setup, and one root partition is kept as a spare on each controller. When creating data aggregates, at least one data partition should be left unused as a spare on each controller to provide redundancy in the event of a drive failure.

Create interface groups and set MTU

Interface groups are used to bond multiple storage system network ports together for bandwidth aggregation and fault tolerance. Table 4 shows the interface groups that were created on the storage system:

Table 4) NetApp A800 interface group configuration.

Controller Name	Interface Groups	Distribution Function	Mode	MTU	Ports
ontap-01	a11a	port (IP + L4 port)	multimode_lacp	9000	e3a,e5a
ontap-01	a21a	port (IP + L4 port)	multimode_lacp	9000	e3b,e5b
ontap-02	a12a	port (IP + L4 port)	multimode_lacp	9000	e3a,e5a
ontap-02	a22a	port (IP + L4 port)	multimode_lacp	9000	e3b,e5b

Run the following commands to create the interface groups (ifgrps) and configure the MTU:

```
ifgrp create -node ontap-01 -ifgrp a11a -distr-func port -mode multimode_lacp
ifgrp add-port -node ontap-01 -ifgrp a11a -port e3a
ifgrp add-port -node ontap-01 -ifgrp a11a -port e5a
ifgrp create -node ontap-01 -ifgrp a21a -distr-func port -mode multimode_lacp
ifgrp add-port -node ontap-01 -ifgrp a21a -port e3b
ifgrp add-port -node ontap-01 -ifgrp a21a -port e5b
ifgrp create -node ontap-02 -ifgrp a12a -distr-func port -mode multimode_lacp
ifgrp add-port -node ontap-02 -ifgrp a12a -port e3a
ifgrp add-port -node ontap-02 -ifgrp a12a -port e5a
ifgrp create -node ontap-02 -ifgrp a22a -distr-func port -mode multimode_lacp
ifgrp add-port -node ontap-02 -ifgrp a22a -port e3b
ifgrp add-port -node ontap-02 -ifgrp a22a -port e5b

network port modify -node ontap-01 -port a11a -mtu 9000
network port modify -node ontap-01 -port a21a -mtu 9000
network port modify -node ontap-02 -port a12a -mtu 9000
network port modify -node ontap-02 -port a22a -mtu 9000
```

Create storage VLANs

A single VLAN is used for storage connectivity to the hosts. This VLAN was created on each of the two interface groups on each controller, providing a total of four possible failover targets for each VLAN. The MTU must be set on each VLAN in addition to the ports configured above. To configure the storage VLANs and set the MTU, run the following commands:

```
network port vlan create –node ontap-01 -vlan-name a11a-3111
network port vlan create –node ontap-01 -vlan-name a21a-3111
network port vlan create –node ontap-02 -vlan-name a12a-3111
network port vlan create –node ontap-02 -vlan-name a22a-3111

network port modify -node ontap-01 -port a11a-3111 -mtu 9000
network port modify -node ontap-01 -port a21a-3111 -mtu 9000
network port modify -node ontap-02 -port a12a-3111 -mtu 9000
network port modify -node ontap-02 -port a22a-3111 -mtu 9000
```

Create/modify broadcast domains

To create broadcast domains for the newly created VLANs and assign the appropriate ports, run the following commands:

```
broadcast-domain create -broadcast-domain vlan3111 -mtu 9000
```

```
broadcast-domain add-ports -broadcast-domain vlan3111 -ports ontap-01:a11a-3111, ontap-01:a21a-
3111, ontap-02:a12a-3111, ontap-02:a22a-3111
```

Create SVM

Data access for ONTAP systems is provided by a storage virtual machine (SVM). A data SVM must be created with the appropriate protocols before hosts can access the storage system. To create an SVM with the NFS protocol and enable access to the data aggregates, run the following commands:

```
vserver create -vserver data01 -rootvolume data01_rootvol -aggregate n01_data01 -rootvolume-
security-style unix
vserver modify -vserver data01 -aggr-list n01_data01,n02_data01 -allowed-protocols nfs
```

Create NFS service

For clients to access the storage system, a logical NFS server is created on the SVM. To enable maximum throughput, the maximum transfer size should be changed from the default to 256k. To create the NFS service and configure the transfer size, run the following commands:

```
vserver nfs create -vserver data01 -access true -v3 enabled
set advanced

Warning: These advanced commands are potentially dangerous; use them only when directed to do so
by NetApp personnel.
Do you want to continue? {y|n}: y

nfs modify -vserver data01 -tcp-max-xfer-size 262144 -v3-64bit-identifiers enabled
```

Create LIFs

LIFs are the IP addresses that client servers use to mount NFS exports. For this solution, two LIFs are created on each interface group on each controller, providing eight discreet potential mount points leveraging all of the available physical connections. To create the LIFs used in this solution, run the following commands:

```
network interface create -vserver data01 -lif data01-n01-a -role data -data-protocol nfs -home-
node ontap-01 -home-port a11a-3111 -address 192.168.0.101 -netmask 255.255.255.0 -status-admin up
network interface create -vserver data01 -lif data01-n01-b -role data -data-protocol nfs -home-
node ontap-01 -home-port a11a-3111 -address 192.168.0.102 -netmask 255.255.255.0 -status-admin up
network interface create -vserver data01 -lif data01-n01-c -role data -data-protocol nfs -home-
node ontap-01 -home-port a21a-3111 -address 192.168.0.103 -netmask 255.255.255.0 -status-admin up
network interface create -vserver data01 -lif data01-n01-d -role data -data-protocol nfs -home-
node ontap-01 -home-port a21a-3111 -address 192.168.0.104 -netmask 255.255.255.0 -status-admin up

network interface create -vserver data01 -lif data01-n02-a -role data -data-protocol nfs -home-
node ontap-02 -home-port a12a-3111 -address 192.168.0.105 -netmask 255.255.255.0 -status-admin up
network interface create -vserver data01 -lif data01-n02-b -role data -data-protocol nfs -home-
node ontap-02 -home-port a12a-3111 -address 192.168.0.106 -netmask 255.255.255.0 -status-admin up
network interface create -vserver data01 -lif data01-n02-c -role data -data-protocol nfs -home-
node ontap-02 -home-port a22a-3111 -address 192.168.0.107 -netmask 255.255.255.0 -status-admin up
network interface create -vserver data01 -lif data01-n02-d -role data -data-protocol nfs -home-
node ontap-02 -home-port a22a-3111 -address 192.168.0.108 -netmask 255.255.255.0 -status-admin up
```

Create FlexGroup volume

A single NetApp FlexGroup volume with eight constituents per node was created to support multi-file workloads. The FIO benchmarks used in this validation, as well as TensorFlow and most other ML/DL frameworks typically use multiple data files to parallelize the data ingest process, and FlexGroup volumes were designed to maximize storage throughput for these workloads. This configuration can scale across multiple nodes by extending the FlexGroup to the new nodes, providing a single namespace up to the maximum size determined by the number and model of controllers in the cluster. To create the FlexGroup volume, run the following command:

```
volume create -vserver data01 -volume data_flexgroup_01 -auto-provision-as flexgroup -size 20TB -
space-guarantee none -junction-path /data_flexgroup_01
```

For single-file workloads such as the MXNet benchmark tested in this paper, see the Single-File Workload Configuration section below.

Create export policy

Export policies determine which client hosts can access the storage system. To configure an export policy to allow the DGX A100 systems to mount the dataset using NFS, run the following command:

```
vserver export-policy rule create -vserver data01 -policyname default -protocol nfs -clientmatch
192.168.0.0/24 -rorule sys -rwrule sys -superuser sys
Network Switch configuration
```

Complete network switch installation and configuration of the Mellanox SN3700V switches used in this validation is beyond the scope of this document. To enable connectivity for the A800 storage system after initial configuration is complete, the switches must be configured with two LACP LAG groups on each switch. Table 5 shows the required LAG group configuration to match the storage system Interface Group configuration above.

Table 5) Network switch configuration for storage

Switch Name	LAG Name	Load Balancing Algorithm	Mode	MTU	Ports
switch01	storage_one	L4 port	LACP	9216	swp1,swp2
switch01	storage_two	L4 port	LACP	9216	swp11,swp12
switch02	storage_one	L4 port	LACP	9216	swp1,swp2
switch02	storage_two	L4 port	LACP	9216	swp11,swp12

This is an example of the network switch configuration commands for the storage network:

```
net del all
net add dns nameserver ipv4 10.150.224.204 vrf mgmt
net add time ntp server 0.cumulusnetworks.pool.ntp.org iburst
net add time ntp server 1.cumulusnetworks.pool.ntp.org iburst
net add time ntp server 2.cumulusnetworks.pool.ntp.org iburst
net add time ntp server 3.cumulusnetworks.pool.ntp.org iburst
net add time ntp source eth0
net add snmp-server listening-address localhost
net add routing defaults datacenter
net add routing log syslog informational
net add routing service integrated-vtysh-config
net add time zone Etc/UTC
net add interface swp1-32 breakout 1x
net add ptp global slave-only no
net add ptp global priority1 255
net add ptp global priority2 255
net add ptp global domain-number 0
net add ptp global logging-level 5
net add ptp global path-trace-enabled no
net add ptp global use-syslog yes
net add ptp global verbose no
net add ptp global summary-interval 0
net add ptp global time-stamping
net add bond bond_eight bond slaves swp31
net add bond bond_eleven bond slaves swp4
net add bond bond_fifteen bond slaves swp8
net add bond bond_five bond slaves swp28
net add bond bond_four bond slaves swp27
net add bond bond_fourteen bond slaves swp7
net add bond bond_nine bond slaves swp32
```

NetApp ONTAP AI with NVIDIA DGX A100 Systems and Mellanox Spectrum Ethernet Switches

```
net add bond bond_seven bond slaves swp30
net add bond bond_seventeen bond slaves swp10
net add bond bond_six bond slaves swp29
net add bond bond_sixteen bond slaves swp9
net add bond bond_ten bond slaves swp3
net add bond bond_thirteen bond slaves swp6
net add bond bond_three bond slaves swp26
net add bond bond_twelve bond slaves swp5
net add bond bond_two bond slaves swp25
net add bond peerlink bond slaves swp15,swp16,swp17,swp18
net add bond storage_one bond slaves swp1,swp2
net add bond storage_two bond slaves swp11,swp12
net add bond bond_eight clag id 8
net add bond bond_eleven clag id 11
net add bond bond_fifteen clag id 15
net add bond bond_five clag id 5
net add bond bond_four clag id 4
net add bond bond_fourteen clag id 14
net add bond bond_nine clag id 9
net add bond bond_seven clag id 7
net add bond bond_seventeen clag id 17
net add bond bond_six clag id 6
net add bond bond_sixteen clag id 16
net add bond bond_ten clag id 10
net add bond bond_thirteen clag id 13
net add bond bond_three clag id 3
net add bond bond_twelve clag id 12
net add bond bond_two clag id 2
net add bridge bridge ports
bond_eight,bond_eleven,bond_fifteen,bond_five,bond_four,bond_fourteen,bond_nine,bond_seven,bond_s
eventeen,bond_six,bond_sixteen,bond_ten,bond_thirteen,bond_three,bond_twelve,bond_two,peerlink,st
orage_one,storage_two
net add bridge bridge vids 3000,3111-3112
net add bridge bridge vlan-aware
net add interface eth0 ip address 10.150.33.7/26
net add interface eth0 ip gateway 10.150.33.1
net add interface eth0 vrf mgmt
net add interface peerlink.4094 clag backup-ip 10.150.33.8
net add interface peerlink.4094 clag peer-ip linklocal
net add interface peerlink.4094 clag priority 1000
net add interface peerlink.4094 clag sys-mac 44:38:39:FF:07:07
net add interface swp1-32 link speed 100000
net add vrf mgmt ip address 127.0.0.1/8
net add vrf mgmt ipv6 address ::1/128
net add vrf mgmt vrf-table auto
net add hostname RL-SN37V-C23-U47
net add dot1x radius accounting-port 1813
net add dot1x eap-reauth-period 0
net add dot1x default-dacl-preauth-filename default_preauth_dacl.rules
net add dot1x radius authentication-port 1812
net add dot1x mab-activation-delay 30
net commit
```

DGX A100 system configuration

Complete installation and configuration of DGX A100 systems is outside the scope of this document. To enable connectivity to the A800, two 100GbE ports are configured as an LACP LAG group for the storage VLAN and another two ports are configured as a bond for in-band management and client access VLANs. An example of the network interface configuration is used in the following validation:

```
# This file describes the network interfaces available on your system
# For more information, see netplan(5).
network:
  version: 2
  renderer: networkd
  ethernets:
    enp225s0f0:
      match:
        macaddress: 0c:42:a1:0a:2e:ba
```

```
      mtu: 9000
  enp225s0f1:
    match:
      macaddress: 0c:42:a1:0a:2e:bb
    mtu: 9000
  enp97s0f0:
    match:
      macaddress: 0c:42:a1:0a:35:fa
    mtu: 9000
  enp97s0f1:
    match:
      macaddress: 0c:42:a1:0a:35:fb
    mtu: 9000
 bonds:
   storage:
     mtu: 9000
     dhcp4: false
     dhcp6: false
     interfaces: [enp97s0f0, enp225s0f0]
     parameters:
       mode: 802.3ad
       mii-monitor-interval: 1
       transmit-hash-policy: layer3+4
 vlans:
   storage.3111:
     id: 3111
     link: storage
     addresses: [192.168.0.13/24]
     mtu: 9000
```

After network connectivity to the storage network is complete, the NetApp volumes created above must be mounted on each DGX A100 system. Several mount options are required on the host to achieve maximum performance. In addition, each DGX system uses a dedicated IP address to ensure optimal load balancing throughout the network. To mount the FlexGroup volume used for FIO benchmark testing, run the following commands:

On DGX A100 system #1:

```
mount -o nvonnect=16,rsize=262144,wsize=262144 192.168.0.101:/data_flexgroup_01
```

On DGX A100 system #2:

```
mount -o nvonnect=16,rsize=262144,wsize=262144 192.168.0.102:/data_flexgroup_01
```

On DGX A100 system #3:

```
mount -o nvonnect=16,rsize=262144,wsize=262144 192.168.0.103:/data_flexgroup_01
```

On DGX A100 system #4:

```
mount -o nvonnect=16,rsize=262144,wsize=262144 192.168.0.104:/data_flexgroup_01
```

On DGX A100 system #5:

```
mount -o nvonnect=16,rsize=262144,wsize=262144 192.168.0.105:/data_flexgroup_01
```

On DGX A100 system #6:

```
mount -o nvonnect=16,rsize=262144,wsize=262144 192.168.0.106:/data_flexgroup_01
```

On DGX A100 system #7:

```
mount -o nvonnect=16,rsize=262144,wsize=262144 192.168.0.107:/data_flexgroup_01
```

On DGX A100 system #8:

```
mount -o nvonnect=16,rsize=262144,wsize=262144 192.168.0.108:/data_flexgroup_01
```

Solution verification

This solution was validated using both synthetic storage benchmarks and the MLPerf v0.7 training benchmark. Validation test results can be found in the NVA-1153 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, but for more information on 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 AFF A800 storage systems and DGX A100 systems. The DGX A100 system is a next-generation deep learning platform that requires equally advanced storage and data management capabilities. By combining DGX A100 with NetApp AFF systems, this verified architecture can be implemented at almost any scale, from a single DGX A100 paired with an AFF A400 storage system up to potentially 96 DGX A100 systems on a 24-node AFF A800 cluster. Combined with the

superior cloud integration and software-defined capabilities of NetApp ONTAP, AFF enables a full range of data pipelines that spans 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:

- NVA-1153-DESIGN: NetApp ONTAP AI with NVIDIA DGX A100 Systems and Mellanox Spectrum Ethernet Switches
  www.netapp.com/pdf.html?item=/media/21793-nva-1153-design.pdf

NetApp AFF systems:

- AFF datasheet
  https://www.netapp.com/us/media/ds-3582.pdf
- NetApp Flash Advantage for AFF
  https://www.netapp.com/us/media/ds-3733.pdf
- ONTAP 9.x documentation
  http://mysupport.netapp.com/documentation/productlibrary/index.html?productID=62286
- NetApp FlexGroup technical report
  https://www.netapp.com/us/media/tr-4557.pdf

NetApp Interoperability Matrix:

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

NetApp Trident:

- https://netapp.io/persistent-storage-provisioner-for-kubernetes/
- https://netapp-trident.readthedocs.io/en/stable-v19.04/kubernetes/index.html
- https://github.com/NetApp/trident

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 Spectrum SN3000 series switches
  https://www.mellanox.com/products/ethernet-switches/sn3000

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/

NetApp ONTAP AI with NVIDIA DGX A100 Systems and Mellanox Spectrum Ethernet Switches

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.