



经验证的 NetApp 架构

由 NVIDIA 提供支持的 NetApp ONTAP AI 设计一款可扩展的 AI 基础架构，满足实际深度 学习用例需求

NetApp 公司的 David Arnette, Amit Borulkar 和 Robert Franz
2018 年 10 月 | NVA-1121-deploy

合作方



摘要

本文档包含由 NVIDIA® 提供支持的 NetApp® ONTAP® AI 解决方案的详细部署信息。此架构包括 [NetApp AFF A800 存储系统](#)，四个 [NVIDIA DGX-1™](#) 服务器和两个 [Cisco Nexus 3232C](#) 100Gb 以太网交换机。本文档还介绍了如何使用行业标准基准测试工具验证此系统的运行情况和性能。有关此解决方案的详细设计信息，请参见 NetApp 经验证的架构文档 [NVA-1121-design](#)。

目录

1	计划摘要	4
1.1	经验证的 NetApp 架构计划	4
1.2	NetApp ONTAP AI 解决方案	4
2	解决方案概述	5
2.1	解决方案技术	5
2.2	使用情形摘要	6
3	技术要求	7
3.1	硬件要求	7
3.2	软件要求	7
4	解决方案硬件安装和布线	8
4.1	硬件安装和基本设置	8
4.2	解决方案布线	8
5	解决方案配置详细信息	12
5.1	Cisco Nexus 3232C 配置指南	12
5.2	NetApp AFF SP 配置	18
5.3	NVIDIA DGX-1 服务器配置	21
6	解决方案验证	27
6.1	网络验证	27
6.2	存储系统验证	29
6.3	基本 TensorFlow 验证	30
7	结论	33
	致谢	33
	从何处查找其他信息	33
	版本历史记录	34

表格目录

表 1)	硬件要求。	7
表 2)	软件要求。	7
表 3)	Nexus 交换机 A 端口连接。	10
表 4)	Nexus 交换机 B 端口连接。	10

表 5) NetApp AFF A800 端口连接。	11
表 6) NVIDIA DGX-1 服务器端口连接。	11
表 7) Cisco Nexus 交换机 VLAN。	12
表 8) NetApp AFF A800 接口组配置。	19
表 9) NVIDIA DGX-1 物理接口和 VLAN。	23
表 10) NVIDIA DGX-1 绑定接口和 VLAN。	23
表 11) 经过验证的软件版本。	31
表 12) TensorFlow 常规基准设置。	31
表 13) datasets_num_private_threads TensorFlow 基准测试中的设置。	32

插图目录

图 1) NetApp ONTAP AI 解决方案机架级架构.....	5
图 2) 经 ONTAP AI 解决方案验证的架构.....	6
图 3) ONTAP AI 布线图.....	9

1 计划摘要

1.1 经验证的 NetApp 架构计划

经验证的 NetApp 架构计划为客户提供适用于 NetApp 解决方案的经过验证的架构。NVAs 为客户提供的 NetApp 解决方案架构具有以下特性：

- 已通过全面测试
- 规范性
- 尽可能降低部署风险
- 加快上市速度

本文档主要面向 NetApp 及其合作伙伴的解决方案工程师以及客户的战略决策者。其中包括经过测试的此解决方案实施的具体安装和配置信息。有关用于确定满足特定客户要求的适当配置的架构设计注意事项，请参见《[ONTAP AI 设计指南](#)》。

1.2 NetApp ONTAP AI 解决方案

NetApp ONTAP® AI 经过验证的架构由 NVIDIA DGX 超级计算机和 NetApp 云互联存储提供支持，已由 NetApp 和 NVIDIA 开发和验证。它为组织提供了一个规范化的架构，可提供以下优势：

- 它消除了设计复杂性。
- 它允许独立扩展计算和存储。
- 它可以从小规模入手，无缝扩展。
- 提供广泛的存储选项，满足各种性价比需求

NetApp ONTAP AI 将 NVIDIA DGX-1 服务器、NVIDIA Tesla® V100 图形处理单元 (GPU) 和 NetApp AFF A800 系统与一流的网络相集成。ONTAP AI 可消除设计复杂性和猜测性，帮助企业从小规模入手，实现无中断增长，同时智能地管理从边缘到核心再到云再到云的数据，从而简化 AI 部署。

显示了 NetApp ONTAP AI 解决方案的可扩展性。AFF A800 系统使用 4 台 DGX-1 服务器进行了验证，结果证明其具有足够的性能余量，可支持 5 台或更多 DGX-1 服务器，但不会影响存储吞吐量或延迟。此外，通过在 ONTAP 集群中添加更多网络交换机和存储控制器对，该解决方案可扩展到多个机架，提供极高吞吐量，从而加快训练和推理过程。这种方法可以灵活地根据数据湖的大小，使用的深度学习 (DL) 模式以及所需的性能指标单独更改计算与存储的比率。

图 1) NetApp ONTAP AI 解决方案机架级架构。



每个机架中 DGX-1 服务器和 AFF 系统的数量取决于所使用机架的电源和散热规格。系统的最终位置取决于计算流体动力学分析、气流控制 and 数据中心设计。

2 解决方案概述

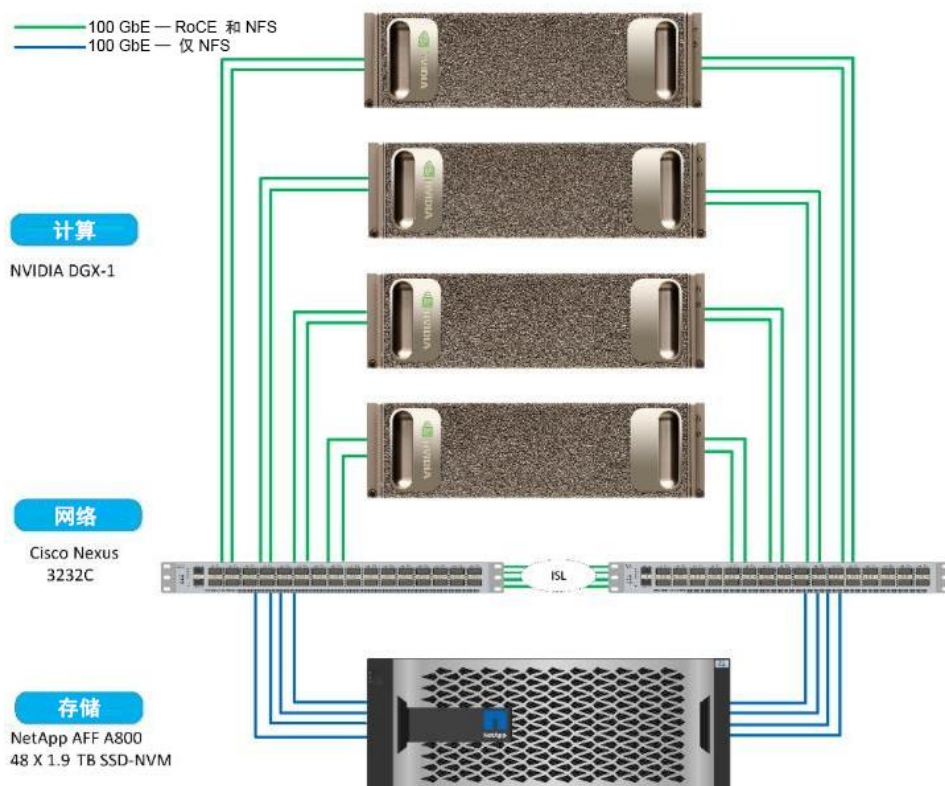
深度学习系统所采用的算法属于计算密集型算法，而且特别适合 NVIDIA GPU 的架构。深度学习算法中执行的计算涉及并行运行的巨量矩阵乘法。相较于通用中央处理单元 (CPU)，现代化 GPU 的高度并行架构可大幅提高并行处理数据的应用程序的效率。利用 DGX-1 服务器的单个和集群 NVIDIA GPU 计算架构的发展进步使其成为高性能计算 (HPC)、深度学习和分析等工作负载的首选平台。要在这些环境中将性能最大化，需要一个能够不断为 NVIDIA GPU 馈送数据的支持基础架构。因此，必须能以超低延迟和高带宽访问数据集。以太网技术已达到以前只有 InfiniBand 才可能达到的性能级别。因此，基于融合以太网的 RDMA (RDMA over Converged Ethernet, RoCE) 可以更轻松地采用这些功能，因为每个企业数据中心都对以太网技术有很好的了解并广泛部署。

2.1 解决方案技术

此解决方案采用 1 个 NetApp AFF A800 系统、4 台 NVIDIA DGX-1 服务器和 2 个 Cisco Nexus 3232C 100 Gb 以太网交换机实施。每个 DGX-1 服务器都通过四个 100GbE 连接连接到 Nexus 交换机，这些连接用于使用 RoCE 进行 GPU 间通信。用于 NFS 存储访问的传统 IP 通信也发生在这些链路上。每个存储控制器均使用 4 个 100GbE 链路连接到网络交换机。

图 2 显示了基础解决方案架构。

图 2) 经过 ONTAP AI 解决方案验证的架构。



2.2 使用情形概要

此解决方案用于支持 AI 和 DL 管道的培训和推理阶段。根据应用程序的不同，深度学习模型将处理大量不同类型的数据（结构化数据和非结构化数据）。这种不同会对底层存储系统提出各种不同的要求，包括存储数据的大小以及数据集中文件的数量。

高级存储要求包括：

- 能同时存储和检索数百万个文件
- 存储和检索多种数据对象，例如图像、音频、视频和时序数据
- 以低延迟提供并行高性能以满足 GPU 处理速度要求
- 实现跨边缘、核心和云的无缝数据管理和数据服务

在关键的深度学习训练阶段，通常会定期将数据从数据湖复制到训练集群中。然后，DL 模型会重复处理这些数据，以达到所需的机器学习能力。此阶段所用服务器使用 GPU 并行进行计算，从而形成巨大的数据处理能力。满足原始 I/O 带宽需求对于保持高 GPU 利用率至关重要。

在推理阶段，对经过训练的模型进行测试并部署到生产环境中。或者，也可以将它们反馈回数据湖，以便进一步调整输入权重。此外，在物联网（IoT）应用程序中，可以将这些型号部署到智能边缘设备上，以便进行初始边缘处理。

3 技术要求

本节介绍验证此解决方案时使用的硬件和软件。[第 6 节“解决方案验证”](#)和《[ONTAP AI 设计指南](#)》中记录的所有测试均使用此处所述的硬件和软件执行。

注意： 在此参考架构中验证的配置基于实验室设备可用性，而不是基于所测试硬件的要求或限制。

3.1 硬件要求

列出了验证此解决方案所使用的硬件组件。根据客户要求，该解决方案的具体实施环境所使用的硬件组件可能会有所不同。

表 1) 硬件要求。

硬件	数量
NVIDIA DGX-1 GPU 服务器	4
NetApp AFF A800 系统	1 个高可用性 (HA) 对，包括 48 个 1.92 TB NVMe SSD
Cisco Nexus 3232C 网络交换机	2

3.2 软件要求

列出了实施此解决方案所需的软件组件。根据客户要求，该解决方案的具体实施环境所使用的软件组件可能会有所不同。

表 2) 软件要求。

软件	版本
NetApp ONTAP	9.4
Cisco NX-OS 交换机固件	7.0(3)I6(1)
NVIDIA DGX-1 操作系统	Ubuntu 16.04 LTS
Docker 容器平台	18.03.1-ce [9ee9f40]
容器版本	基于 nvcr.io/nvidia/tensorflow:18.04-py2 的 netapp_1.7.0.2
机器学习框架	TensorFlow 1.7.0
Horovod	0.11.3
打开 MPI	3.1.0
基准测试软件	TensorFlow 基准测试 [1b1ca8a]

4 解决方案硬件安装和布线

4.1 硬件安装和设置海报

所有硬件组件都应按照供应商建议的准则安装在数据中心机架中。验证此解决方案时使用的所有组件均可装入一个机架，并可容纳更多 DGX-1 服务器。特定机架电源和散热容量决定了每个机架可支持的服务器数量。

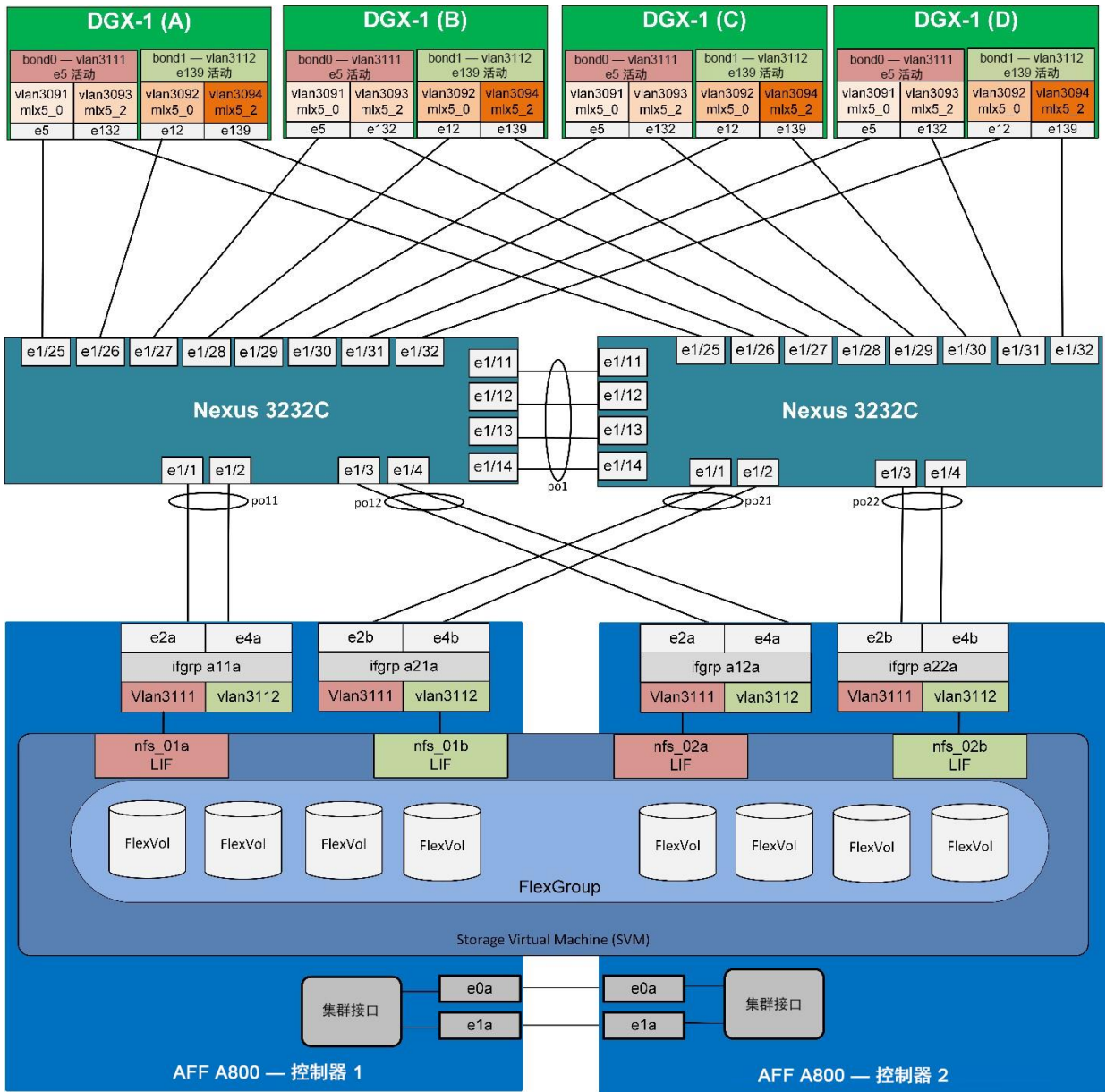
使用相应的安装文档对每个组件执行基本设置。以下配置过程假定已安装并配置所有组件以进行管理访问，并已升级到此验证中建议的软件和固件版本。有关基本安装和设置的具体详细信息，请参见相应的供应商文档。"配置详细信息"部分提供了链接。

4.2 解决方案布线

本节包含有关验证此解决方案时使用的特定布线的信息。可以修改此布线配置，以满足客户特定的实施要求。

图 3 显示了此解决方案的布线。为了简明起见，此图对 DGX-1 物理接口名称进行了缩写。

图 3) ONTAP AI 布线图。



网络交换机端口连接

表 3 显示了交换机 A 的交换机连接

表 3) Nexus 交换机 A 端口连接。

交换机名称	端口名称	已连接设备和端口
ONTAPI-SW-A	Eth1/1	A800-01: E2A
ONTAPI-SW-A	Eth1/2	A800-01: E4A
ONTAPI-SW-A	Eth1/3	A800-02: E2A
ONTAPI-SW-A	Eth1/4	A800-02: E4A
ONTAPI-SW-A	Eth1/11	ONTAPI-SW-B: Eth1/11
ONTAPI-SW-A	Eth1/12	ONTAPI-SW-B: Eth1/12
ONTAPI-SW-A	Eth1/13	ONTAPI-SW-B: Eth1/13
ONTAPI-SW-A	Eth1/14	ONTAPI-SW-B: Eth1/14
ONTAPI-SW-A	Eth1/25	DGX1-A: vp5s0
ONTAPI-SW-A	Eth1/26	DGX1-A: vp12s0
ONTAPI-SW-A	Eth1/27	DGX1-B: vp5s0
ONTAPI-SW-A	Eth1/28	DGX1-B: vp12s0
ONTAPI-SW-A	Eth1/29	DGX1-C: vp5s0
ONTAPI-SW-A	Eth1/30	DGX1-C: vp12s0
ONTAPI-SW-A	Eth1/31	DGX1-D: vp5s0
ONTAPI-SW-A	Eth1/32	DGX1-D: vp12s0

表 4 显示了交换机 B 的交换机连接

表 4) Nexus 交换机 B 端口连接。

交换机名称	端口名称	已连接设备和端口
ONTAPI-SW-B	Eth1/1	A800-01: e2b
ONTAPI-SW-B	Eth1/2	A800-01: e4b
ONTAPI-SW-B	Eth1/3	A800-02: e2b
ONTAPI-SW-B	Eth1/4	A800-02: e4b
ONTAPI-SW-B	Eth1/11	ONTAPI-SW-A: Eth1/11
ONTAPI-SW-B	Eth1/12	ONTAPI-SW-A: Eth1/12
ONTAPI-SW-B	Eth1/13	ONTAPI-SW-A: Eth1/13
ONTAPI-SW-B	Eth1/14	ONTAPI-SW-A: Eth1/14
ONTAPI-SW-B	Eth1/25	DGX1-A: vp132s0
ONTAPI-SW-B	Eth1/26	DGX1-A: vp139s0
ONTAPI-SW-B	Eth1/27	DGX1-B: vp132s0
ONTAPI-SW-B	Eth1/28	DGX1-B: enp139s0
ONTAPI-SW-B	Eth1/29	DGX1-C: vp132s0
ONTAPI-SW-B	Eth1/30	DGX1-C: vp139s0
ONTAPI-SW-B	Eth1/31	DGX1-D: vp132s0
ONTAPI-SW-B	Eth1/32	DGX1-D: vp139s0

存储控制器端口连接

表 5 显示了连接到 NetApp AFF A800 存储系统的端口。

表 5) NetApp AFF A800 端口连接。

控制器名称	端口名称	已连接设备和端口
A800-01	e0M	管理交换机
A800-01	e0a	A800-02: e0a
A800-01	e1a	A800-02: e1a
A800-01	e2a	ONTAPI-SW-A: eth1/1
A800-01	e2b	ONTAPI-SW-B: eth1/1
A800-01	E4A	ONTAPI-SW-A: eth1/2
A800-01	e4b.	ONTAPI-SW-B: eth1/2
A800-02	e0M	管理交换机
A800-02	e0a	A800-01: e0a
A800-02	e1a	A800-01: e1a
A800-02	e2a	ONTAPI-SW-A: eth1/3
A800-02	e2b	ONTAPI-SW-B: eth1/3
A800-02	E4A	ONTAPI-SW-A: eth1/4
A800-02	e4b.	ONTAPI-SW-B: eth1/4

DGX-1 网络端口连接

表 6 显示了四个 DGX-1 服务器中每个服务器上连接的网络端口。

表 6) NVIDIA DGX-1 服务器端口连接。

控制器名称	端口名称	已连接设备和端口
DGX1-A	enp1s0	管理交换机
DGX1-A	enp5s0	ONTAPI-SW-A: eth1/25
DGX1-A	enp12s0	ONTAPI-SW-A: eth1/26
DGX1-A	enp132s0	ONTAPI-SW-B: eth1/25
DGX1-A	enp139s0	ONTAPI-SW-B: eth1/26
DGX1-B	enp1s0	管理交换机
DGX1-B	enp5s0	ONTAPI-SW-A: eth1/27
DGX1-B	enp12s0	ONTAPI-SW-A: eth1/27
DGX1-B	enp132s0	ONTAPI-SW-B: eth1/28
DGX1-B	enp139s0	ONTAPI-SW-B: eth1/28
DGX1-C	enp1s0	管理交换机
DGX1-C	enp5s0	ONTAPI-SW-A: eth1/29
DGX1-C	enp12s0	ONTAPI-SW-A: eth1/29
DGX1-C	enp132s0	ONTAPI-SW-B: eth1/30
DGX1-C	enp139s0	ONTAPI-SW-B: eth1/30
DGX1-D	enp1s0	管理交换机
DGX1-D	enp5s0	ONTAPI-SW-A: eth1/31

DGX1-D	enp12s0	ONTAPI-SW-A: eth1/31
DGX1-D	enp132s0	ONTAPI-SW-B: eth1/32
DGX1-D	enp139s0	ONTAPI-SW-B: eth1/32

5 解决方案配置详细信息

本节介绍了有关完成解决方案组件配置的具体配置详细信息和说明。其中包括以下任务：

- Cisco Nexus 3232C 配置指南
- NetApp AFF SP 配置
- NVIDIA DGX-1 代理服务器

5.1 Cisco Nexus 3232C 配置 基本设置

有关 Cisco Nexus 交换机安装和基本设置的详细说明，请参见 [Cisco Nexus 3000 系列硬件安装指南](#)。

按照前面所述的布线准则配置交换机进行管理访问并进行布线后，以下配置过程将创建一个与此解决方案验证中测试的配置类似的配置。

启用功能

必须为此解决方案启用以下功能：

- 链路聚合控制协议 (Link Aggregation Control Protocol, LACP)
- 链路层发现协议 (LLDP)

要启用所需的 Nexus 功能，请完成以下步骤：

1. 以管理员身份登录。
2. 运行以下命令：

```
config t
feature lacp
feature lldp
```

创建 VLAN

表 7 中列出的虚拟 LAN (VLAN) 是为此解决方案创建的。

表 7) Cisco Nexus 交换机 VLAN 。

VLAN ID	VLAN Name
3091	RoCE VLAN 01
3092	RoCE VLAN 02
3093	RoCE VLAN 03
3094	RoCE-VLAN-04
3111	NFS-VLAN-01
3112	NFS-VLAN-02

要创建此解决方案所需的 VLAN，请在全局配置提示符处运行以下命令：

```
vlan 3091
name ROCE-VLAN-01
vlan 3092
name ROCE-VLAN-02
vlan 3093
name ROCE-VLAN-03
vlan 3094
name ROCE-VLAN-04
vlan 3111
name NFS-VLAN-01
vlan 3112
name NFS-VLAN-02
```

配置 QoS

QoS 配置设置可启用 RoCE 使用的无损以太网传输，并为每个流量类定义带宽分配。使用以下操作步骤创建 QoS 配置：

1. 创建类型为 QoS 的类映射，以便将服务类（CoS）值应用于输入流量。

```
class-map type qos match-all class_RoCE
  match cos 3
class-map type qos match-all class_control
  match cos 5-7
```

2. 创建类型为 QoS 的策略映射以设置每个类的 QoS 组。

```
policy-map type qos RoCE-marking-policy
  class class_control
    set qos-group 7
  class class_RoCE
    set qos-group 3
  class class-default
    set qos-group 0
```

3. 创建队列类型策略映射以为每个流量类分配带宽。

```
policy-map type queuing RoCE-queue-policy
  class type queuing c-out-8q-q7
    priority level 1
  class type queuing c-out-8q-q6
    bandwidth remaining percent 0
  class type queuing c-out-8q-q5
    bandwidth remaining percent 0
  class type queuing c-out-8q-q4
    bandwidth remaining percent 0
  class type queuing c-out-8q-q3
    bandwidth remaining percent 90
  class type queuing c-out-8q-q2
    bandwidth remaining percent 0
  class type queuing c-out-8q-q1
    bandwidth remaining percent 0
  class type queuing c-out-8q-q-default
    bandwidth remaining percent 10
```

4. 创建网络 QoS 策略映射，为相关队列定义最大传输单元（MTU）和基于优先级的流量控制（PFC）暂停设置。

```
policy-map type network-qos RoCE-nq-policy
  class type network-qos c-8q-nq7
    mtu 1500
  class type network-qos c-8q-nq-default
    mtu 9216
  class type network-qos c-8q-nq3
    mtu 9216
    pause pfc-cos 3
```

5. 将 network-qos 和队列策略应用为全局服务策略。

```
system qos
  service-policy type network-qos RoCE-nq-policy
  service-policy type queuing output RoCE-queue-policy
```

除了这些 QoS 配置之外，还会启用 PFC，并将 QoS 标记策略应用于连接到 DGX-1 服务器端口的每个接口。以下服务器端口配置部分包含这些配置详细信息。

配置 ISL 端口和端口通道

两个 Nexus 3232C 交换机使用四个 100 Gb 以太网链路彼此连接。这些链路配置为一个端口通道，用于带宽聚合和容错。

1. 要在交换机 A 上配置用于交换机间链路的端口和端口通道，请在全局配置提示符处运行以下命令：

```
interface Po1
  description INTER-SWITCH-LINK
  switchport mode trunk
  switchport trunk allowed vlan 3091-3094,3111-3112
  priority-flow-control mode on
  spanning-tree port type network
  mtu 9216
  service-policy type qos input qos_policy
  no shutdown
interface Eth1/11
  description ISL-NEXUS-B:Eth1/11
  switchport mode trunk
  switchport trunk allowed vlan 3091-3094,3111-3112
  priority-flow-control mode on
  mtu 9216
  channel-group 1 mode active
  no shutdown
interface Eth1/12
  description ISL-NEXUS-B:Eth1/12
  switchport mode trunk
  switchport trunk allowed vlan 3091-3094,3111-3112
  priority-flow-control mode on
  mtu 9216
  channel-group 1 mode active
  no shutdown
interface Eth1/13
  description ISL-NEXUS-B:Eth1/13
  switchport mode trunk
  switchport trunk allowed vlan 3091-3094,3111-3112
  priority-flow-control mode on
  mtu 9216
  channel-group 1 mode active
  no shutdown
interface Eth1/14
  description ISL-NEXUS-B:Eth1/14
  switchport mode trunk
  switchport trunk allowed vlan 3091-3094,3111-3112
  priority-flow-control mode on
  mtu 9216
  channel-group 1 mode active
  no shutdown
```

2. 要在交换机 B 上配置用于交换机间链路的端口和端口通道，请在全局配置提示符处运行以下命令：

```
interface Po1
  description INTER-SWITCH-LINK
  switchport mode trunk
  switchport trunk allowed vlan 3091-3094,3111-3112
  priority-flow-control mode on
```

```

spanning-tree port type network
mtu 9216
service-policy type qos input qos_policy
no shutdown
interface Eth1/11
description ISL-NEXUS-A:Eth1/11
switchport mode trunk
switchport trunk allowed vlan 3091-3094,3111-3112
priority-flow-control mode on
mtu 9216
channel-group 1 mode active
no shutdown
interface Eth1/12
description ISL-NEXUS-A:Eth1/12
switchport mode trunk
switchport trunk allowed vlan 3091-3094,3111-3112
priority-flow-control mode on
mtu 9216
channel-group 1 mode active
no shutdown
interface Eth1/13
description ISL-NEXUS-A:Eth1/13
switchport mode trunk
switchport trunk allowed vlan 3091-3094,3111-3112
priority-flow-control mode on
mtu 9216
channel-group 1 mode active
no shutdown
interface Eth1/14
description ISL-NEXUS-A:Eth1/14
switchport mode trunk
switchport trunk allowed vlan 3091-3094,3111-3112
priority-flow-control mode on
mtu 9216
channel-group 1 mode active
no shutdown

```

配置存储端口和端口通道

每个存储控制器都通过一个双端口 LACP 端口通道连接到每个 Nexus 3232C 交换机。

1. 要在交换机 A 上为存储系统连接配置 Nexus 端口，请在全局配置提示符处运行以下命令：

```

interface port-channel11
description A800-01:a11a
switchport mode trunk
switchport trunk allowed vlan 3111-3112
spanning-tree port type edge trunk
mtu 9216
interface port-channel12
description A800-02:a12a
switchport mode trunk
switchport trunk allowed vlan 3111-3112
spanning-tree port type edge trunk
mtu 9216
interface Ethernet1/1
description A800-01:e2a
switchport mode trunk
switchport trunk allowed vlan 3111-3112
spanning-tree port type edge trunk
mtu 9216
channel-group 11 mode active
interface Ethernet1/2
description A800-01:e4a
switchport mode trunk
switchport trunk allowed vlan 3111-3112
spanning-tree port type edge trunk
mtu 9216
channel-group 11 mode active

```

```

interface Ethernet1/3
description A800-02:e2a
switchport mode trunk
switchport trunk allowed vlan 3111-3112
spanning-tree port type edge trunk
mtu 9216
channel-group 12 mode active
interface Ethernet1/4
description A800-02:e4a
switchport mode trunk
switchport trunk allowed vlan 3111-3112
spanning-tree port type edge trunk
mtu 9216
channel-group 12 mode active

```

2. 要在交换机 B 上为存储系统连接配置 Nexus 端口，请在全局配置提示符处运行以下命令：

```

interface port-channel21
description A800-01:a21a
switchport mode trunk
switchport trunk allowed vlan 3111-3112
spanning-tree port type edge trunk
mtu 9216
interface port-channel22
description A800-02:a22a
switchport mode trunk
switchport trunk allowed vlan 3111-3112
spanning-tree port type edge trunk
mtu 9216
interface Ethernet1/1
description A800-01:e2b
switchport mode trunk
switchport trunk allowed vlan 3111-3112
spanning-tree port type edge trunk
mtu 9216
channel-group 21 mode active
interface Ethernet1/2
description A800-01:e4b
switchport mode trunk
switchport trunk allowed vlan 3111-3112
spanning-tree port type edge trunk
mtu 9216
channel-group 21 mode active
interface Ethernet1/3
description A800-02:e2b
switchport mode trunk
switchport trunk allowed vlan 3111-3112
spanning-tree port type edge trunk
mtu 9216
channel-group 22 mode active
interface Ethernet1/4
description A800-02:e4b
switchport mode trunk
switchport trunk allowed vlan 3111-3112
spanning-tree port type edge trunk
mtu 9216
channel-group 22 mode active

```

配置主机端口

连接到 DGX-1 服务器的端口用作单个端口，不会配置为端口通道。

1. 要配置用于将 DGX-1 连接到交换机 A 的端口，请在全局配置提示符处运行以下命令：

```

interface Ethernet1/25
description dgx1_enp5s0
switchport mode trunk

```



```

switchport trunk allowed vlan 3091-3094,3111-3112
priority-flow-control mode auto
spanning-tree port type edge trunk
mtu 9216
service-policy type qos input RoCE-marking-policy
interface Ethernet1/26
description dgx1_enp12s0
switchport mode trunk
switchport trunk allowed vlan 3091-3094,3111-3112
priority-flow-control mode on
spanning-tree port type edge trunk
mtu 9216
service-policy type qos input RoCE-marking-policy
interface Ethernet1/27
description dgx2_enp5s0
switchport mode trunk
switchport trunk allowed vlan 3091-3094,3111-3112
priority-flow-control mode on
spanning-tree port type edge trunk
mtu 9216
service-policy type qos input RoCE-marking-policy
interface Ethernet1/28
description dgx2_enp12s0
switchport mode trunk
switchport trunk allowed vlan 3091-3094,3111-3112
priority-flow-control mode on
spanning-tree port type edge trunk
mtu 9216
service-policy type qos input RoCE-marking-policy
interface Ethernet1/29
description dgx3_enp5s0
switchport mode trunk
switchport trunk allowed vlan 3091-3094,3111-3112
priority-flow-control mode on
spanning-tree port type edge trunk
mtu 9216
service-policy type qos input RoCE-marking-policy
interface Ethernet1/30
description dgx3_enp12s0
switchport mode trunk
switchport trunk allowed vlan 3091-3094,3111-3112
priority-flow-control mode on
spanning-tree port type edge trunk
mtu 9216
service-policy type qos input RoCE-marking-policy
interface Ethernet1/31
description dgx4_enp5s0
switchport mode trunk
switchport trunk allowed vlan 3091-3094,3111-3112
priority-flow-control mode on
spanning-tree port type edge trunk
mtu 9216
service-policy type qos input RoCE-marking-policy
interface Ethernet1/32
description dgx4_enp12s0
switchport mode trunk
switchport trunk allowed vlan 3091-3094,3111-3112
priority-flow-control mode on
spanning-tree port type edge trunk
mtu 9216
service-policy type qos input RoCE-marking-policy

```

2. 要配置用于将 DGX-1 连接到交换机 B 的端口，请在全局配置提示符处运行以下命令：

```

interface Ethernet1/25
description dgx1_enp132s0
switchport mode trunk
switchport trunk allowed vlan 3091-3094,3111-3112
priority-flow-control mode on
spanning-tree port type edge trunk

```

```

mtu 9216
service-policy type qos input RoCE-marking-policy
interface Ethernet1/26
description dgx1_enp139s0
switchport mode trunk
switchport trunk allowed vlan 3091-3094,3111-3112
priority-flow-control mode on
spanning-tree port type edge trunk
mtu 9216
service-policy type qos input RoCE-marking-policy
interface Ethernet1/27
description dgx2_enp132s0
switchport mode trunk
switchport trunk allowed vlan 3091-3094,3111-3112
priority-flow-control mode on
spanning-tree port type edge trunk
mtu 9216
service-policy type qos input RoCE-marking-policy
interface Ethernet1/28
description dgx2_enp139s0
switchport mode trunk
switchport trunk allowed vlan 3091-3094,3111-3112
priority-flow-control mode on
spanning-tree port type edge trunk
mtu 9216
service-policy type qos input RoCE-marking-policy
interface Ethernet1/29
description dgx3_enp132s0
switchport mode trunk
switchport trunk allowed vlan 3091-3094,3111-3112
priority-flow-control mode on
spanning-tree port type edge trunk
mtu 9216
service-policy type qos input RoCE-marking-policy
interface Ethernet1/30
description dgx3_enp139s0
switchport mode trunk
switchport trunk allowed vlan 3091-3094,3111-3112
priority-flow-control mode on
spanning-tree port type edge trunk
mtu 9216
service-policy type qos input RoCE-marking-policy
interface Ethernet1/31
description dgx4_enp132s0
switchport mode trunk
switchport trunk allowed vlan 3091-3094,3111-3112
priority-flow-control mode on
spanning-tree port type edge trunk
mtu 9216
service-policy type qos input RoCE-marking-policy
interface Ethernet1/32
description dgx4_enp139s0
switchport mode trunk
switchport trunk allowed vlan 3091-3094,3111-3112
priority-flow-control mode on
spanning-tree port type edge trunk
mtu 9216
service-policy type qos input RoCE-marking-policy

```

5.2 NetApp AFF SP 配置

有关安装和基本设置 NetApp AFF A800 存储系统的详细说明，请参见此 [AFF A800 文档](#)。

完成存储系统设置并按照前面提供的布线准则对系统进行布线后，以下存储配置过程将按照此解决方案验证中的测试创建配置。

管理默认广播域

广播域用于将 L2 相邻端口分组在一起，以便进行故障转移。如果发生链路故障，逻辑接口（LIF）会自动故障转移到同一广播域中的另一个端口。如果可能，LIF 会尝试故障转移到同一控制器上的其他端口，或者根据需要故障转移到另一控制器上的相应端口。由于每个 VLAN 代表一个单独的 L2 子网，因此会为每个 VLAN 创建一个存储广播域。

默认情况下，所有网络端口都包含在默认广播域中。应从默认广播域中删除用于数据服务的网络端口（E2A, e2b, E4A 和 e4b），而只保留管理网络端口（e0M）。要执行此任务，请运行以下命令：

```
broadcast-domain remove-ports -broadcast-domain Default -ports A800-01:e2a,A800-01:e2b,A800-01:e4a,A800-01:e4b, A800-02:e2a,A800-02:e2b,A800-02:e4a,A800-02:e4b
```

创建数据聚合

此解决方案已使用 AFF A800 的默认聚合配置进行验证，其中每个控制器托管一个使用 47 个磁盘分区创建的数据聚合。如果在初始集群设置期间未创建数据聚合，请运行以下命令创建两个数据聚合：

```
aggr create -aggregate aggr1_node01 -node A800-01 -diskcount 47
aggr create -aggregate aggr1_node02 -node A800-02 -diskcount 47
```

请注意，NetApp ONTAP 会将每个固态驱动器（SSD）分区为两个小型根分区和两个较大的数据分区，每个分区中有一个分配给每个控制器。根分区用于控制器节点的根聚合，较大的分区用于数据聚合。在初始设置期间会自动创建根聚合，并且每个控制器上会保留一个根分区作为备用分区。创建数据聚合时，请在每个控制器上至少保留一个未使用的数据分区作为备用分区，以便在驱动器发生故障时提供冗余。

创建接口组并设置 MTU

接口组用于将多个存储系统网络端口绑定在一起，以实现带宽聚合和容错。表 8 显示了在存储系统上创建的接口组。

表 8) NetApp AFF A800 接口组配置。

控制器名称	接口组	分发功能	模式	MTU	端口
A800-01	a11a	顺序（轮循）	multimode_lacp	9000	E2A, E4A
A800-01	a21a	顺序（轮循）	multimode_lacp	9000	e2b, e4b
A800-02	a12a	顺序（轮循）	multimode_lacp	9000	E2A, E4A
A800-02	a22a	顺序（轮循）	multimode_lacp	9000	e2b, e4b

运行以下命令以创建接口组（ifgrp）并配置 MTU：

```
ifgrp create -node A800-01 -ifgrp a11a -distr-func sequential -mode multimode_lacp
ifgrp add-port -node A800-01 -ifgrp a11a -port e2a
ifgrp add-port -node A800-01 -ifgrp a11a -port e4a
ifgrp create -node A800-01 -ifgrp a21a -distr-func sequential -mode multimode_lacp
ifgrp add-port -node A800-01 -ifgrp a21a -port e2b
ifgrp add-port -node A800-01 -ifgrp a21a -port e4b
ifgrp create -node A800-02 -ifgrp a12a -distr-func sequential -mode multimode_lacp
ifgrp add-port -node A800-02 -ifgrp a12a -port e2a
ifgrp add-port -node A800-02 -ifgrp a12a -port e4a
ifgrp create -node A800-02 -ifgrp a22a -distr-func sequential -mode multimode_lacp
ifgrp add-port -node A800-02 -ifgrp a22a -port e2b
ifgrp add-port -node A800-02 -ifgrp a22a -port e4b
```

```
network port modify -node A800-01 -port a11a -mtu 9000
network port modify -node A800-01 -port a21a -mtu 9000
network port modify -node A800-02 -port a12a -mtu 9000
network port modify -node A800-02 -port a22a -mtu 9000
```

Create VLAN ports and set MTU

创建存储 VLAN

此解决方案中使用了两个单独的 **NFS VLAN**，以便根据需要通过网络为每个主机创建两个独立的路径。每个 **VLAN** 都在每个控制器上的两个接口组中的每个接口组上创建，从而为每个 **VLAN** 总共提供四个可能的故障转移目标。要配置用于验证此解决方案的存储 **VLAN**，请运行以下命令：

```
network port vlan create -node A800-01 -vlan-name a11a-3111
network port vlan create -node A800-01 -vlan-name a21a-3111
network port vlan create -node A800-01 -vlan-name a11a-3112
network port vlan create -node A800-01 -vlan-name a21a-3112
network port vlan create -node A800-02 -vlan-name a12a-3111
network port vlan create -node A800-02 -vlan-name a22a-3111
network port vlan create -node A800-02 -vlan-name a12a-3112
network port vlan create -node A800-02 -vlan-name a22a-3112
```

创建或修改广播域

要为新创建的 **VLAN** 创建广播域并分配适当的端口，请运行以下命令：

```
broadcast-domain create -broadcast-domain NFS-A -mtu 9000
broadcast-domain create -broadcast-domain NFS-B -mtu 9000
broadcast-domain add-ports -broadcast-domain NFS-A -ports A800-01:a11a-3111, A800-01:a21a-3111,
A800-02:a12a-3111, A800-02:a22a-3111
broadcast-domain add-ports -broadcast-domain NFS-B -ports A800-01:a11a-3112, A800-01:a21a-3112,
A800-02:a12a-3112, A800-02:a22a-3112
```

创建 SVM。

ONTAP 系统的数据访问由 **Storage Virtual Machine (SVM)** 提供。主机必须先使用适当的协议创建数据 **SVM**，然后才能访问存储系统。要使用 **NFS** 协议创建 **SVM** 并允许访问数据聚合，请运行以下命令：

```
vserver create -vserver AI-SVM -rootvolume AI_SVM_rootvol -aggregate aggr1_node01 -
rootvolume- security-style unix
vserver modify -vserver AI-SVM -aggr-list aggr1_node01,aggr1_node02 -allowed-protocols nfs
```

创建 LIF。

LIF 是客户端服务器用于挂载 **NFS** 导出的 **IP** 地址。对于此解决方案，会在每个控制器的每个接口组上创建一个 **LIF**，以提供离散的潜在挂载点。要创建此解决方案中使用的 **LIF**，请运行以下命令：

```
network interface create -vserver AI-SVM -lif NFS-01-A -role data -data-protocol nfs -home-node
A800-01 -home-port a11a-3111 -address 172.31.11.11 -netmask 255.255.255.0 -status-admin up
network interface create -vserver AI-SVM -lif NFS-01-B -role data -data-protocol nfs -home-node
A800-01 -home-port a21a-3112 -address 172.31.12.11 -netmask 255.255.255.0 -status-admin up
network interface create -vserver AI-SVM -lif NFS-02-A -role data -data-protocol nfs -home-node
A800-02 -home-port a12a-3111 -address 172.31.11.12 -netmask 255.255.255.0 -status-admin up
network interface create -vserver AI-SVM -lif NFS-02-B -role data -data-protocol nfs -home-node
A800-02 -home-port a22a-3112 -address 172.31.12.12 -netmask 255.255.255.0 -status-admin up
```

FlexGroup 卷创建

创建了一个每个节点包含四个成分卷的 NetApp ONTAP FlexGroup 卷，用于托管此验证中使用的 ImageNet 数据。要创建 FlexGroup 卷，请运行以下命令：

```
volume show -vserver AI-SVM -junction
flexgroup deploy -size 10TB -type RW -space-guarantee none -foreground true -vserver AI-SVM -
volume imagenet_dataset
```

创建导出策略。

导出策略用于确定哪些客户端主机可以访问存储系统。要配置导出策略以允许 DGX-1 服务器使用 NFS 挂载数据集，请运行以下命令：

```
vserver export-policy rule create -vserver AI-SVM -policyname default -protocol nfs -clientmatch
172.31.11.0/24 -rorule sys -rwrule sys -superuser sys -allow-suid false
vserver export-policy rule create -vserver AI-SVM -policyname default -protocol nfs -clientmatch
172.31.12.0/24 -rorule sys -rwrule sys -superuser sys -allow-suid false

vserver nfs create -access true -v3 enabled
```

5.3 NVIDIA DGX-1 代理服务器

有关首次安装 DGX-1 服务器的详细说明，请参见《[DGX-1 用户指南](#)》。完成服务器的初始安装和设置后，以下操作步骤将完成此解决方案所需的配置。

1. 使用默认凭据（或在 DGX_1 安装步骤中设置的身份验证凭据）登录到 DGX-1 服务器，并为 root 用户设置安全 Shell (SSH) 访问权限。

```
sudo passwd root
[sudo] password for dgxuser:
Enter new UNIX password:
Retype new UNIX password:
passwd: password updated successfully
```

2. 编辑 /etc/ssh/sshd_config 此文件以允许 root 用户使用 SSH 登录。

```
PermitRootLogin yes
#PermitRootLogin prohibit-password
```

3. 验证是否已配置默认的 Canonical 和 NVIDIA 存储库。

```
/etc/apt/sources.list.d/dgx.list
```

```
deb http://international.download.nvidia.com/dgx/repos/ xenial main multiverse restricted
universe
```

```
/etc/apt/sources.list.d/docker.list
```

```
deb [arch=amd64] https://download.docker.com/linux/ubuntu xenial stable
```

```
/etc/apt/sources.list
```

```
deb http://security.ubuntu.com/ubuntu xenial-security main
deb http://security.ubuntu.com/ubuntu xenial-security universe
deb http://security.ubuntu.com/ubuntu xenial-security multiverse
deb http://archive.ubuntu.com/ubuntu/ xenial main multiverse universe
deb http://archive.ubuntu.com/ubunsudop tu/ xenial-updates main multiverse universe
```

注意：在有气流的环境 /etc/apt/sources.list sources.list.d 中，请确保这些存储库已镜像并进行了适当配置。

4. 从存储库更新软件包。

```
sudo apt-get update
```

5. 验证 DGX-1 会话

```
grep VERSION /etc/dgx-release  
DGX_SWBUILD_VERSION="3.1.6"
```

6. 配置 NTP 服务器以确保所有组件都处于时间同步状态。

```
sudo apt install ntp
```

7. 在 /etc/ntp.conf 中配置适合您环境的时间服务器。

```
server ntp1.lab.netapp.com  
server ntp2.lab.netapp.com  
server ntp3.lab.netapp.com
```

8. 重新启动 NTP 服务器服务。

```
systemctl restart ntp
```

将 InfiniBand 适配器转换为 100GbE

1. 下载 [Mellanox 软件工具包](#)。

2. 解压缩已下载的软件包。

```
tar -xvf mft-4.6.0-48-x86_64-deb.gz
```

3. install.sh 从 mft-4.6.0-48-x86_64-deb 目录运行。

```
root@wdl-dgx4:/home/dgxuser/mft-4.6.0-48-x86_64-deb# ./install.sh  
-I- Removing all installed mft packages: mft kernel-mft-dkms  
-I- Installing package: /home/dgxuser/mft-4.6.0-48-x86_64-deb/SDEBS/kernel-mft-dkms_4.6.0-48_all.deb  
-I- Installing package: /home/dgxuser/mft-4.6.0-48-x86_64-deb/DEBS/mft-4.6.0-48.amd64.deb  
-I- In order to start mst, please run "mst start".
```

4. 启动 Mellanox 工具。

```
root@wdl-dgx4:/home/dgxuser/mft-4.6.0-48-x86_64-deb# mst start  
Starting MST (Mellanox Software Tools) driver set  
Loading MST PCI module - Success  
Loading MST PCI configuration module - Success  
Create devices  
Unloading MST PCI module (unused) - Success
```

5. 将 InfiniBand 端口转换为以太网。

```
mlxconfig -y -d /dev/mst/mt4115_pciconf0 set LINK_TYPE_P1=2  
mlxconfig -y -d /dev/mst/mt4115_pciconf1 set LINK_TYPE_P1=2  
mlxconfig -y -d /dev/mst/mt4115_pciconf2 set LINK_TYPE_P1=2  
mlxconfig -y -d /dev/mst/mt4115_pciconf3 set LINK_TYPE_P1=2
```

6. 验证是否已应用配置更改。

```
mlxconfig query |grep -e "LINK_TYPE\|PCI\ device"  
PCI device: /dev/mst/mt4115_pciconf3  
LINK_TYPE_P1 ETH(2)  
PCI device: /dev/mst/mt4115_pciconf2  
LINK_TYPE_P1 ETH(2)  
PCI device: /dev/mst/mt4115_pciconf1  
LINK_TYPE_P1 ETH(2)  
PCI device: /dev/mst/mt4115_pciconf0  
LINK_TYPE_P1 ETH
```

7. 重新启动服务器以使更改保持不变。

```
sudo reboot
```

注： 有关将端口特性从 InfiniBand 更改为以太网的全面说明，请参见 [DGX 系统文档](#)。

配置 100GbE 接口

100GbE 接口配置为在 DGX-1 服务器之间实现互连，以传输 RoCE 流量并连接到 AFF A800 存储系统。表 9 显示了在相应接口上标记的 VLAN 。

表 9) NVIDIA DGX-1 物理接口和 VLAN 。

物理接口	VLAN
enp5s0	RoCE VLAN 01
enp12s0	RoCE VLAN 02
enp132s0	RoCE VLAN 03
enp139s0	RoCE-VLAN-04

两个 100GbE 接口在主动 - 被动配置中绑定在一起，并使用 NFS VLAN 进行标记，如表 10 所示。

表 10) NVIDIA DGX-1 绑定接口和 VLAN 。

绑定接口	"interfaces": [VLAN
bond0	enp5s0, enp132s0	NFS-VLAN-01
绑 1	enp12s0, enp139s0	NFS-VLAN-02

要配置主机网络，请完成以下步骤：

1. 安装相应的软件包。

```
sudo apt-get install vlan ifenslave
```

2. 确保在启动期间加载 VLAN 和绑定模块。

```
echo "8021q" >> /etc/modules  
echo "bonding" >> /etc/modules
```

注意： 您可能需要切换到 root 用户才能运行这些命令。

3. /etc/network/interfaces 修改

```
# This file describes the network interfaces available on your system  
# and how to activate them. For more information, see interfaces(5).  
  
source /etc/network/interfaces.d/*  
  
# The loopback network interface  
auto lo  
iface lo inet loopback  
  
# The primary network interface  
auto enp1s0f0  
iface enp1s0f0 inet static  
    address 172.17.213.110  
    netmask 255.255.252.0  
    network 172.17.212.0  
    broadcast 172.17.215.255  
    gateway 172.17.212.1  
    # dns-* options are implemented by the resolvconf package, if installed  
    dns-nameservers 172.19.2.30 172.19.3.32
```

```
dns-search lab.netapp.com

auto bond0
iface bond0 inet manual
    bond-mode active-backup
    bond-miimon 100
    bond-slaves none

auto enp5s0
iface enp5s0 inet manual
    bond-master bond0
    bond-primary enp5s0

auto enp132s0
iface enp132s0 inet manual
    bond-master bond0

auto bond1
iface bond1 inet manual
    bond-mode active-backup
    bond-miimon 100
    bond-slaves none

auto enp12s0
iface enp12s0 inet manual
    bond-master bond1
    bond-primary enp12s0

auto enp139s0
iface enp139s0 inet manual
    bond-master bond1

auto enp5s0.3091
iface enp5s0.3091 inet static
address 172.31.91.24
netmask 255.255.255.0
mtu 9000
vlan-raw-device enp5s0

auto enp12s0.3092
iface enp12s0.3092 inet static
address 172.31.92.24
netmask 255.255.255.0
mtu 9000
vlan-raw-device enp12s0

auto enp132s0.3093
iface enp132s0.3093 inet static
address 172.31.93.24
netmask 255.255.255.0
mtu 9000
vlan-raw-device enp132s0

auto enp139s0.3094
iface enp139s0.3094 inet static
address 172.31.94.24
netmask 255.255.255.0
mtu 9000
vlan-raw-device enp139s0

auto bond0.3111
iface bond0.3111 inet static
address 172.31.11.24
netmask 255.255.255.0
mtu 9000

auto bond1.3112
iface bond1.3112 inet static
address 172.31.12.24
```



```
netmask 255.255.255.0
mtu 9000
```

4. 重新启动网络以使更改生效。

```
systemctl restart networking
```

5. 在所有 100GbE 接口的优先级 3 上启用 PFC 。

```
mlnx_qos -i enp5s0 --pfc 0,0,0,1,0,0,0,0
mlnx_qos -i enp12s0 --pfc 0,0,0,1,0,0,0,0
mlnx_qos -i enp132s0 --pfc 0,0,0,1,0,0,0,0
mlnx_qos -i enp139s0 --pfc 0,0,0,1,0,0,0,0
```

安装 NFS 存储系统

为了提供确定性的性能和故障转移行为，每个主机在 AFF A800 存储系统上使用一个特定的 IP 地址来挂载 FlexGroup 数据卷。要在每个主机上挂载卷，请运行以下命令：

```
# On DGX-1 server 1
sudo mount -t nfs -o vers=3 172.31.11.11:/fgvolume [path in DGX-1 server 1]
# On DGX-1 server 2
sudo mount -t nfs -o vers=3 172.31.11.12:/fgvolume [path in DGX-1 server 2]
# On DGX-1 server 3
sudo mount -t nfs -o vers=3 172.31.12.11:/fgvolume [path in DGX-1 server 3]
# On DGX-1 server 4
sudo mount -t nfs -o vers=3 172.31.12.12:/fgvolume [path in DGX-1 server 4]
```

注意： 添加其他 DGX-1 系统后，挂载点将以轮循方式进行设置。

安装和升级 Docker

DGX-1 系统预安装了 Docker，但可能不是最新版本。

1. 要卸载现有版本的 Docker，安装最新版本并将当前用户添加到 Docker 组，请运行以下命令：

```
sudo apt-get remove docker docker-engine docker.io
sudo apt-get install docker-ce
sudo usermod -a -G docker $USER
```

2. 重新启动 Docker 引擎。

```
systemctl restart docker
```

注意： 如果未将用户添加到 Docker 组，sudo 则需要提升权限才能运行任何 Docker 命令。

安装和升级 NVIDIA 软件

DGX-1 服务器还随附安装了 NVIDIA-Docker 扩展软件包，但也应在安装期间进行更新。

1. 要删除现有版本的 NVIDIA-Docker，请运行以下命令：

```
docker volume ls -q -f driver=nvidia-docker | xargs -r -I{} -n1 docker ps -q -a -f volume={} |
xargs -r docker rm -f
sudo apt-get purge -y nvidia-docker
```

2. 要添加 nvidia-Docker 存储库并下载并安装最新版本，请运行以下命令：

```
curl -s -L https://nvidia.github.io/nvidia-docker/gpgkey | \
sudo apt-key add -
distribution=$(. /etc/os-release;echo $ID$VERSION_ID)
```

```
curl -s -L https://nvidia.github.io/nvidia-docker/$distribution/nvidia-docker.list | sudo tee
/etc/apt/sources.list.d/nvidia-docker.list
sudo apt-get update
sudo apt-get install -y nvidia-docker2
```

3. 将存储驱动程序修改为 overlay2 。

```
sudo vi /etc/systemd/system/docker.service.d/docker-override.conf
[Service]
ExecStart=
ExecStart=/usr/bin/dockerd -H fd:// -s overlay2 Limit
MEMLOCK=infinity
LimitSTACK=67108864
```

4. 安装新版本的 NVIDIA-Docker 后，运行以下命令重新启动 Docker 守护进程：

```
sudo systemctl restart docker
```

5. 通过运行最新的 NVIDIA/CUDA 映像来验证 NVIDIA 运行时。

```
docker run --runtime=nvidia --rm nvidia/cuda nvidia-smi
```

从 NVIDIA GPU Cloud 下载 ML Framework 容器

借助 NVIDIA GPU Cloud (NGC)，研究人员和数据科学家可以访问一个全面的 GPU 优化软件工具目录，这些工具适用于深度学习和 HPC，可充分利用 NVIDIA GPU。NGC 容器注册表采用经过 NVIDIA 调整，测试，认证和维护的容器，适用于顶级 DL 框架。要获取官方 NVIDIA 容器，请参见 [《DGX 容器注册表用户指南》](#)。

为了提高使用多个 DGX-1 服务器的分布式培训的性能，容器中还安装了一个名为 Horovod 的附加软件包。Horovod 通过 NVIDIA Collective Communications Library (NCCL) 更新变量，并使用 Open MPI 在 DGX-1 服务器之间进行通信。Open MPI 会通过 SSH 发送命令，但默认 NVIDIA 容器未安装 openssh-server 和 openssh-client。要添加 Horovod，请更新 Open MPI 并添加 SSH 客户端和服务器组件。以下 Docker 文件用于创建在此验证中使用的容器，并以默认 NVIDIA TensorFlow 容器作为起点。

```
FROM nvcr.io/nvidia/tensorflow:18.04-py2

# Install required packages
RUN apt-get update && apt-get install -y --no-install-recommends \
    libibverbs1 \
    libibverbs-dev \
    libmlx5-1 \
    librdmacm-dev \
    librdmacm1 \
    openssh-client \
    openssh-server \
    infiniband-diags \
    rdmacm-utils \
    libmthca-dev \
    ibutils \
    ibverbs-utils \
    && \
    rm -rf /var/lib/apt/lists/*

# (Optional) Install a newer version of OpenMPI
ENV OPENMPI_VERSION 3.1.0
ENV OPENMPI_URL https://download.open-mpi.org/release/open-mpi/v3.1/openmpi-
${OPENMPI_VERSION}.tar.gz
RUN wget -q -O - ${OPENMPI_URL} | tar -xzf - && \
    cd openmpi-${OPENMPI_VERSION} && \
    ./configure --enable-orterun-prefix-by-default \
    --with-cuda --with-verbs \
```

```

    --prefix=/usr/local/mpi --disable-getpuid && \
    make -j"$(nproc)" install && \
cd .. && rm -rf openmpi-${OPENMPI_VERSION}
ENV PATH /usr/local/mpi/bin:$PATH

# Create SSH authentication and Disable SSH host key checking
RUN mkdir -p /var/run/ssh && \
    mkdir -p /root/.ssh && \
    echo "    StrictHostKeyChecking no" >> /etc/ssh/ssh_config && \
    echo "    UserKnownHostsFile /dev/null" >> /etc/ssh/ssh_config && \
    echo "    LogLevel quiet" >> /etc/ssh/ssh_config && \
    mkdir -p /root/.ssh && \
    echo "HOST *" > /root/.ssh/config && \
    ssh-keygen -t rsa -b 4096 -f /root/.ssh/id_rsa -N "" && \
    cp /root/.ssh/id_rsa.pub /root/.ssh/authorized_keys && \
    chmod 700 /root/.ssh && \
    chmod 600 /root/.ssh/*

# Update libraries
RUN ldconfig

```

6 解决方案验证

本节介绍了用于验证是否已正确安装和配置基础架构的测试信息。应执行本节中的每个测试，以确保系统按预期运行。

6.1 网络验证

执行以下测试，确保已为所使用的每种流量类型正确配置网络。

RoCE 网络验证测试

此测试用于验证 DGX-1 服务器是否可以使用 RoCE 网络连接彼此通信。

1. 确保已使用本文档前面 "配置 100GbE 接口" 一节中的操作步骤配置所有 100 GB 网络接口。
2. 运行 Mellanox 带宽验证工具，对用于 RoCE 通信的每个 100 Gb 以太网链路执行 I/O:
 - a. 在一台服务器上运行以下命令:

```
ib_send_bw -d mlx5_0 -i 1 -S 3 -F --report_gbits
```

- b. 在第二台服务器上运行以下命令:

```
ib_send_bw -d mlx5_0 -i 1 -S 3 -F --report_gbits 172.31.91.24
```

- c. 确保达到平均带宽超过 95%。

```

-----
-----
Send BW Test
Dual-port QPS      :          Device      : mlx5_0
数量              : OFF                Transport type : IB
连接类型          :                    Using SRQ    : OFF
TX depth          : 128
CQ Moderation     : 100
Mtu               : 4096[B]
Link type         : Ethernet
GID index         : 3
Max inline data   : 0[B]
rdma_cm QPs      : OFF

```

Data ex. method : Ethernet

local address: LID 0000 QPN 0x00d7 PSN 0x7a569d
GID: 00:00:00:00:00:00:00:00:00:00:255:255:172:31:91:23
remote address: LID 0000 QPN 0x01ca PSN 0x5578f6
GID: 00:00:00:00:00:00:00:00:00:00:255:255:172:31:91:24

#bytes #iterations BW peak[Gb/sec] BW average[Gb/sec] MsgRate[Mpps]
65536 1000 96.89 96.88 0.184786

d. 在相应的网络交换机上，确保在 RoCE 流量通过网络时，QoS 组 3 的 QoS 计数器正在递增。

```
Sh queuing interface eth1/25
x6-3232-sw1# sh queuing interface eth1/25
```

```
slot 1
=====
```

```
Egress Queuing for Ethernet1/25 [System]
```

QoS-Group#	Bandwidth%	PrioLevel	Min	Shape Max	Units	QLimit
7	-	1	-	-	-	6 (D)
6	0	-	-	-	-	6 (D)
5	0	-	-	-	-	6 (D)
4	0	-	-	-	-	6 (D)
3	90	-	-	-	-	- (U)
2	0	-	-	-	-	6 (D)
1	0	-	-	-	-	6 (D)
+	0	10	-	-	-	6 (D)

```
-----
|                               QOS GROUP 0                               |
+-----+-----+-----+-----+
|                               | Unicast | OOBFC Unicast | Multicast |
+-----+-----+-----+-----+
| Tx Pkts | 5380880 | 0 | 9531 |
| Tx Byts | 393869894 | 0 | 5695330 |
| Dropped Pkts | 0 | 0 | 0 |
| Dropped Byts | 0 | 0 | 0 |
| Q Depth Byts | 0 | 0 | 0 |
+-----+-----+-----+-----+
```

```
-----
|                               QOS GROUP 1                               |
+-----+-----+-----+-----+
|                               | Unicast | OOBFC Unicast | Multicast |
+-----+-----+-----+-----+
| Tx Pkts | 0 | 0 | 0 |
| Tx Byts | 0 | 0 | 0 |
| Dropped Pkts | 0 | 0 | 0 |
| Dropped Byts | 0 | 0 | 0 |
| Q Depth Byts | 0 | 0 | 0 |
+-----+-----+-----+-----+
```

```
-----
|                               QOS GROUP 2                               |
+-----+-----+-----+-----+
|                               | Unicast | OOBFC Unicast | Multicast |
+-----+-----+-----+-----+
| Tx Pkts | 0 | 0 | 0 |
| Tx Byts | 0 | 0 | 0 |
| Dropped Pkts | 0 | 0 | 0 |
| Dropped Byts | 0 | 0 | 0 |
| Q Depth Byts | 0 | 0 | 0 |
+-----+-----+-----+-----+
```

```
-----
|                               QOS GROUP 3                               |
+-----+-----+-----+-----+
|                               | Unicast | OOBFC Unicast | Multicast |
+-----+-----+-----+-----+
```

```

+-----+
| Tx Pkts | 81936594 | 0 | 0 |
| Tx Byts | 246652519884 | 0 | 0 |
| Dropped Pkts | 0 | 0 | 0 |
| Dropped Byts | 0 | 0 | 0 |
| Q Depth Byts | 0 | 0 | 0 |
+-----+
| QOS GROUP 4 |
+-----+
<output truncated>

```

e. 对每个服务器和接口组合重复上述步骤，以验证每个服务器的 GPU 间带宽。

- DGX1-A mlx5_0 -> DGX1-B mlx5_0
- DGX1-A mlx5_0 -> DGX1-C mlx5_0
- DGX1-A mlx5_0 -> DGX1-D mlx5_0
- DGX1-A mlx5_1 -> DGX1-B mlx5_1
- DGX1-A mlx5_1 -> DGX1-C mlx5_1
- DGX1-A mlx5_1 -> DGX1-D mlx5_1
- DGX1-A mlx5_2 -> DGX1-B mlx5_2
- DGX1-A mlx5_2 -> DGX1-C mlx5_2
- DGX1-A mlx5_2 -> DGX1-D mlx5_2
- DGX1-A mlx5_3 -> DGX1-B mlx5_3
- DGX1-A mlx5_3 -> DGX1-C mlx5_3
- DGX1-A mlx5_3 -> DGX1-D mlx5_3

网络 QoS 验证测试

1. 验证是否已根据先前的操作步骤配置网络交换机。
2. 验证是否已根据先前的操作步骤配置所有 100 GB 网络接口。在每台服务器上安装并运行 iperf3 实用程序，以生成 TCP/IP 网络流量。
3. 用程序，以生成 TCP/IP 网络流量。
 - a. 安装 iperf3 。

```
apt-get install iperf3
```

b. 在一台服务器上运行以下命令以启动侦听器守护进程：

```
iperf3 -s -p 7501 -D
```

c. 在第二台服务器上运行以下命令：

```
iperf3 -c 172.31.94.23 -t 30 -l 1M -p 7501
```

d. 验证是否已达到大约 45 Gbps 的平均带宽。

```

[ ID] Interval           Transfer    Bandwidth    Retr
[ 4]  0.00-30.00 sec    161 GBytes  46.1 Gbits/sec  1956  sender
[ 4]  0.00-30.00 sec    161 GBytes  46.1 Gbits/sec                receiver

```

4. 同时运行 iperf3 测试和 ib_send_bw 测试。

6.2 存储系统验证

本测试旨在验证 DGX-1 服务器是否能够以合适的性能访问 NetApp AFF A800 存储系统上的数据。此测试使用开源 I/O 生成器 FIO 来执行与培训 workload 所产生的 I/O 操作类似的 I/O 操作。要执行此测试，请完成以下步骤。

1. 在 DGX-1 上安装 FIO 软件。

```
Apt-get install fio
```

2. 创建名为 test.fio 的 FIO 作业文件，其中包含以下内容：

```
[global]
rw=read
size=51200m
directory=/fgvvolume #This should be set to a directory on the external storage NFS mount
bs=64k

[job1]
```

此文件将创建一个读取 50 GB 的作业，其中一个线程以 64 k 块大小执行 100% 顺序读取，并对指定的目录执行读取。有关 FIO 的追加信息，请参见 [FIO文档](#)。

3. 在每个 DGX-1 服务器上，运行以下 FIO 命令。

4. 同时所有四个服务器上运行相同的 FIO 命令，以验证每个服务器的并发带宽。

此测试的预期结果是，每个 DGX-1 服务器的存储带宽约为 1100 MBps 到 1200 MBps，如以下输出所示，所有四个 DGX-1 服务器的存储带宽合在一起为 4 Gbps 到 5 Gbps 。

```
root@wdl-dgx3:~# fio test.fio
job1: (g=0): rw=read, bs=64K-64K/64K-64K/64K-64K, ioengine=sync, iodepth=1
fio-2.2.10
Starting 1 process
job1: Laying out IO file(s) (1 file(s) / 51200MB)
Jobs: 1 (f=1): [R(1)] [100.0% done] [2003MB/0KB/0KB /s] [32.4K/0/0 iops] [eta 00m:00s]
job1: (groupid=0, jobs=1): err= 0: pid=29804: Thu Sep 6 07:53:08 2018
 read : io=51200MB, bw=1944.3MB/s, iops=31108, runt= 26334msec
    clat (usec): min=5, max=2614, avg=30.81, stdev=60.21
      lat (usec): min=5, max=2614, avg=30.85, stdev=60.21
    clat percentiles (usec):
       | 1.00th=[  5], 5.00th=[  5], 10.00th=[  6], 20.00th=[  6],
       | 30.00th=[ 10], 40.00th=[ 10], 50.00th=[ 11], 60.00th=[ 11],
       | 70.00th=[ 18], 80.00th=[ 32], 90.00th=[ 65], 95.00th=[ 116],
       | 99.00th=[ 306], 99.50th=[ 338], 99.90th=[ 386], 99.95th=[ 398],
       | 99.99th=[ 1048]
    bw (MB /s): min=  0, max= 2012, per=100.00%, avg=1962.81, stdev=277.60
    lat (usec) : 10=28.50%, 20=44.52%, 50=13.40%, 100=8.17%, 250=2.47%
    lat (usec) : 500=2.92%, 750=0.01%, 1000=0.01%
    lat (msec)  : 2=0.01%, 4=0.01%
    cpu        : usr=2.32%, sys=57.60%, ctx=94831, majf=0, minf=37
  IO depths   : 1=100.0%, 2=0.0%, 4=0.0%, 8=0.0%, 16=0.0%, 32=0.0%, >=64=0.0%
    submit    : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
    complete  : 0=0.0%, 4=100.0%, 8=0.0%, 16=0.0%, 32=0.0%, 64=0.0%, >=64=0.0%
    issued   : total=r=819200/w=0/d=0, short=r=0/w=0/d=0, drop=r=0/w=0/d=0
    latency   : target=0, window=0, percentile=100.00%, depth=1

Run status group 0 (all jobs):
  READ: io=51200MB, aggrbw=1944.3MB/s, minb=1944.3MB/s, maxb=1944.3MB/s, mint=26334msec,
maxt=26334msec

Disk stats (read/write):
sdb: ios=203422/8, merge=0/1, ticks=30712/0, in_queue=30704, util=78.70%
```

6.3 基本 TensorFlow 验证

本节介绍如何通过 TensorFlow 上四种常见的卷积神经网络（CNN）模型验证解决方案。表 11 介绍了此测试中使用的软件版本，表 12 和表 13 显示了所使用的 TensorFlow 设置。

表 11) 经过验证的软件版本。

软件	版本
NetApp ONTAP	9.4
Cisco NX-OS 交换机固件	7.0(3)I6(1)
NVIDIA DGX-1 操作系统	Ubuntu 16.04 LTS
Docker 容器平台	18.03.1-ce [9ee9f40]
容器版本	基于 nvcr.io/nvidia/tensorflow:18.04-py2 的 netapp_1.7.0.2
机器学习框架	TensorFlow 1.7.0
Horovod	0.11.3
打开 MPI	3.1.0
基准测试软件	TensorFlow 基准测试 [1b1ca8a]

表 12) TensorFlow 常规基准设置。

参数	值
batch_group_size	20
update_method	Horovod
forward_only	true (仅当执行推理测试时; 否则为 false)
distortions	False
num_intra_threads	1
num_inter_threads	10
datasets_use_prefetch	True
prefetch_buffer_size	20
use_dataset	True
data_name	ImageNet
use_fp16	true (仅当在 Tensor 核心上测试时)
use_tf_layers	false (仅当在 Tensor 核心上测试时)
force_gpu_compatible	True
data_format	网络信息
device	GPU
num_gpu	1 (对于 Horovod 为必需项, Open MPI 用于控制 GPU 的总数)
datasets_num_private_threads	请参见表 13

表 13) datasets_num_private_threads TensorFlow 基准测试中的设置。

类型:	NVIDIA 类型:	型号	值
培训	CUDA 核心	ResNet-50	1
		ResNet-152	1
		Inception-v3	1
		VGG16	1
	Tensor 核心	ResNet-50	2
		ResNet-152	2
		Inception-v3	2
		VGG16	2
推理	CUDA 核心	ResNet-50	5
		ResNet-152	2
		Inception-v3	5
		VGG16	2
	Tensor 核心	ResNet-50	10
		ResNet-152	5
		Inception-v3	9
		VGG16	5

设置并运行基本测试

以下示例展示了在四个 DGX-1 服务器上批大小为 256 的 RESNET-50 CNN 型号。在这四台服务器中，选择一台作为初始化命令的主要工作服务器。另外三个是二级员工，他们只侦听特定端口并等待主员工发出命令。

1. 在选定为主服务器上运行以下命令：

```
docker run -it --runtime=nvidia --privileged --network=host \
--ulimit memlock=-1 --shm-size=1g --ulimit stack=67108864 \
-v ~/tensorflow:/tensorflow -v /mnt:/mnt --rm netapp_tf_1.7.0.2 \
mpirun -allow-run-as-root -bind-to none -map-by slot \
-np 32 -H wdl-dgx1:8,wdl-dgx2:8,wdl-dgx3:8,wdl-dgx4:8 \
-x NCCL_DEBUG=INFO -x LD_LIBRARY_PATH -x PATH -mca pml ob1 \
-mca btl ^openib -mca btl_tcp_if_include enpls0f0 \ # use TCP for MPI
-x NCCL_IB_HCA=mlx5 \ # assign the interface for NCCL
-x NCCL_NET_GDR_READ=1 \ # RoCE receive to memory directly
-x NCCL_IB_SL=3 -x NCCL_IB_GID_INDEX=3 \ # RoCE CoS priority
-mca plm_rsh_agent ssh -mca plm_rsh_args "-p 12345" \ # ssh port 12345
python tf_cnn_benchmarks.py --model=resnet50 --batch_size=256 \
--batch_group_size=20 --num_epochs=4 --nodistortions \
--num_gpus=1 --device=gpu --force_gpu_compatible=True \
--data_format=NCHW --use_fp16=True --use_tf_layers=False \
--data_name=imagenet --use_datasets=True \
--data_dir=/mnt/mldata/mldata_single/imagenet_1 \
--num_intra_threads=1 --num_inter_threads=10 \
--variable_update=horovod --horovod_device=gpu \
--datasets_prefetch_buffer_size=20 --datasets_num_private_threads=2
```


2. 在二级服务器上运行以下命令:

```
docker run -it --runtime=nvidia --privileged --network=host --ulimit memlock=-1 --shm-size=1g --ulimit stack=67108864 -v ~/tensorflow:/tensorflow -v /mnt:/mnt --rm netapp_tf_1.7.0.2 bash -c "/usr/sbin/sshd -p 12345; sleep infinity"
```

7 结论

这一经过 NetApp 验证的架构为使用 NVIDIA DGX-1 GPU 服务器, NetApp AFF A800 全闪存存储系统和 Cisco Nexus 3000 100GbE 网络交换机构建 AI 培训基础架构提供了参考设计。通过使用标准数据中心组件, 此解决方案可以消除 InfiniBand 解决方案的成本和复杂性, 并为企业 IT 部门提供部署标准化 AI 培训基础架构的蓝图。

此解决方案表明, AFF A800 可以轻松支持八个或更多 DGX-1 服务器的工作负载。客户还可以使用 NetApp AFF A220, AFF A300 或 AFF A700 存储系统以及更少的 DGX-1 服务器来实施此架构, 或者使用存储集群来支持大型 DGX-1 场。

本文档包含在实验室中测试的 ONTAP AI 解决方案的详细部署和验证说明。尽管具体结果可能因多种因素而异, 但客户可以确信, 使用本指南部署基础架构可以获得与《[ONTAP AI 设计指南](#)》中所示类似的性能。

声明

我们要对来自 NVIDIA 令人尊敬的同行 Darrin Johnson、Tony Paikeday、Robert Sohigian 和 James Mauro 对这个经验证的 NetApp 架构所做的贡献表示衷心的感谢。没有 NetApp 关键团队成员 Robert Franz 和 Kesari Mishra 的支持和指导, 我们就无法完成此项研究。

我们要对自己的真知灼见为本白皮书的研究带来巨大帮助的所有人士表示真诚的赞赏和谢意。

从何处查找其他信息

如需详细了解本文档所述的信息, 请参见以下文档和/或网站:

- NVIDIA DGX-1 服务器
<https://www.nvidia.com/en-us/data-center/dgx-1/>
- NVIDIA Tesla V100 Tensor 核心 GPU
<https://www.nvidia.com/en-us/data-center/tesla-v100/>
- NVIDIA GPU Cloud
<https://www.nvidia.com/en-us/gpu-cloud/>
- 《NVIDIA 连接 DGX-1 用户指南》
<https://docs.nvidia.com/dgx/dgx1-user-guide/index.html>
- 《NVIDIA 容器注册表用户指南》
<https://docs.nvidia.com/dgx/dgx-registry-user-guide/index.html>
- 为 100 Gb 以太网配置 Mellanox 适配器
<https://docs.nvidia.com/dgx/dgx1-user-guide/configuring-managing-dgx1.html#infiniband-port-正在更改>
- NetApp AFF 系统

- AFF 产品规格
<https://www.netapp.com/cn/media/ds-3582.pdf>
- NetApp 借助 AFF 展现闪存优势
<https://www.netapp.com/us/media/ds-3733.pdf>
- ONTAP 9.x 文档
<http://mysupport.netapp.com/documentation/productlibrary/index.html?productID=62286>
- NetApp FlexGroup 技术报告
<https://www.netapp.com/cn/media/tr-4557.pdf>
- NetApp AFF A800 安装说明
https://library.netapp.com/ecm/ecm_download_file/ECMLP2842669
- NetApp 互操作性表工具
<http://support.netapp.com/matrix>
- Cisco Nexus 网络
 - Cisco Nexus 3232C 系列交换机
<https://www.cisco.com/c/en/us/products/switches/nexus-3232c-switch/index.html>
 - Cisco Nexus 3232C 配置指南
<https://www.cisco.com/c/en/us/support/switches/nexus-3000-series-switches/products-installation-and-configuration-guides-list.html>
 - Cisco Nexus 3232C 硬件安装指南
https://www.cisco.com/c/en/us/td/docs/switches/datacenter/nexus3000/hw/installation/guide/b_n3000_hardware_install_guide.html
 - Cisco Nexus 3232C 命令行参考
<https://www.cisco.com/c/en/us/support/switches/nexus-3000-series-switches/products-command-reference-list.html>
- 机器学习框架：
 - TensorFlow: 适合所有人的开源机器学习框架
<https://www.tensorflow.org/>
 - Horovod: Uber 适用于 TensorFlow 的开源分布式深度学习框架
<https://eng.uber.com/horovod/>
 - 在容器运行时生态系统中启用 GPU
<https://devblogs.nvidia.com/gpu-containers-runtime/>
- 数据集与基准测试：
 - ImageNet
<http://www.image-net.org/>
 - TensorFlow 基准测试
<https://www.tensorflow.org/performance/benchmarks>
 - FIO 磁盘 I/O 延迟

版本历史

版本	日期	文档版本历史
1.0 版	10 月	初始版本

要验证您的特定环境是否支持本文档所述的确切产品和功能版本，请参见 NetApp 支持站点上的[互操作性表工具 \(IMT\)](#)。NetApp IMT 中定义的产品组件和版本可用于构建 NetApp 所支持的配置。具体的配置结果取决于每个客户如何依照所发布规格进行安装。

版权信息

版权所有 © 2018 NetApp, Inc. 和 NVIDIA。保留所有权利。中国印刷。未经版权所有者事先书面许可，本档中受版权保护的任部分不得以任何形式或通过任何手段（图片、电子或机械方式，包括影印、录音、录像或存储在电子检索系统中）进行复制。

从受版权保护的 NetApp 资料派生的软件受以下许可和免责声明的约束：

本软件由 NetApp 按“原样”提供，不含任何明示或暗示担保，包括但不限于适销性以及针对特定用途的适用性的隐含担保，特此声明不承担任何责任。在任何情况下，对于因使用本软件而以任何方式造成的任何直接性、间接性、偶然性、特殊性、惩罚性或后果性损失（包括但不限于购买替代商品或服务；使用、数据或利润方面的损失；或者业务中断），无论原因如何以及基于何种责任理论，无论出于合同、严格责任或侵权行为（包括疏忽或其他行为），NetApp 均不承担责任，即使已被告知存在上述损失的可能性。

NetApp 保留在不另行通知的情况下随时对本文档所述的任何产品进行更改的权利。除非 NetApp 以书面形式明确同意，否则 NetApp 不承担因使用本文档所述产品而产生的任何责任或义务。使用或购买本产品不表示获得 NetApp 的任何专利权、商标权或任何其他知识产权许可。

本手册中描述的产品可能受一项或多项美国专利、外国专利或正在申请的专利的保护。

有限权利说明：美国政府使用、复制或公开本软件受 DFARS 252.277-7103（1988 年 10 月）和 FAR 52-227-19（1987 年 6 月）中“技术数据和计算机软件权利”条款第 (c)(1)(ii) 条规定的限制条件的约束。

商标信息

NetApp、NetApp 标识和 <http://www.netapp.com/TM> 上所列的商标是 NetApp, Inc. 的商标。其他公司和产品名称可能是其各自所有者的商标。