



分布式 **Azure** 培训—通道检测

NetApp Solutions

NetApp
April 12, 2024

目录

- TR-4896： Azure 中的分布式培训：通道检测—解决方案设计 1
 - 解决方案概述 1
 - 车道检测—使用 Run： AI 进行分布式培训 3
 - 结论 27
 - 追加信息 27

TR-4896： Azure 中的分布式培训：通道检测—解决方案设计

NetApp Ronen Dar 的 Mameer Ahmad 和 Verron Martina，Run：AI

自 2019 年 5 月起，Microsoft 推出了 Azure 原生，这是基于 NetApp ONTAP 技术的第一方门户服务，适用于企业级 NFS 和 SMB 文件服务。这一发展由 Microsoft 和 NetApp 之间的战略合作伙伴关系推动，进一步将世界级 ONTAP 数据服务的覆盖范围扩展到 Azure。

作为领先的云数据服务提供商，NetApp 与 Run：AI 合作，这是一家对 AI 基础架构进行虚拟化的公司，可以利用全 GPU 利用率加快 AI 实验速度。通过这种合作关系，团队可以并行运行多项实验，快速访问数据并利用无限计算资源，从而加快 AI 的运行速度。Run：AI 通过自动分配资源来充分利用 GPU，而经验证的 Azure NetApp Files 架构可以消除数据管道障碍，让每个实验都能以最高速度运行。

NetApp 和 Run：AI 联手为客户在 Azure 中的人工智能之旅提供一个适应未来需求的平台。从分析和高性能计算（HPC）到自主决策（客户只需在需要时为所需的资源付费，即可优化 IT 投资），NetApp 与 Run 之间的联合：AI 在 Azure Cloud 中提供统一的体验。

解决方案概述

在此架构中，重点关注 AI 或机器学习（ML）分布式车道检测训练流程中计算最密集的部分。车道检测是自动驾驶中最重要的任务之一，它有助于通过对车道标记进行定位来引导车辆。车道标记等静态组件可引导车辆以交互方式安全地在高速公路上驾驶。

基于卷积神经网络（CNN）的方法将场景理解和分段提升到了一个新的水平。虽然对于结构较长的对象以及可能被堵塞的区域（例如，极柱，车道上的阴影等）来说，这种方法并不能很好地发挥作用。空间对流神经网络（SCNN）将 CNN 概括为丰富的空间级别。它可以在同一层的神经元之间传播信息，从而最适合结构化对象，例如通道，极或带有 occlusi 的叉车。这种兼容性是因为空间信息可以得到增强，并且可以保持平稳性和连续性。

需要在系统中注入数千个场景图像，以便模型能够学习和区分数据集中的各个组件。这些图像包括天气，日间或夜间，多层公路以及其他交通状况。

对于培训，需要高质量和高数量的数据。单个 GPU 或多个 GPU 可能需要数天到数周才能完成培训。数据分布式培训可通过使用多个和多节点 GPU 来加快此过程。Horovod 就是这样一个框架，它可以提供分布式培训，但在 GPU 集群之间读取数据可能会成为一种障碍。Azure NetApp Files 提供超快，高吞吐量和持续低延迟，可提供横向扩展 / 纵向扩展功能，从而充分利用 GPU 的计算容量。我们的实验证实，集群中的所有 GPU 平均使用 96% 以上的 GPU 来使用 SCNN 进行通道检测训练。

目标受众

数据科学在 IT 和业务领域整合了多个学科，因此我们的目标受众中包含多个角色：

- 数据科学家需要灵活地使用自己选择的工具和库。
- 数据工程师需要了解数据的流动方式及其所在位置。
- 自动驾驶用例专家。
- 云管理员和架构师设置和管理云（Azure）资源。

- 开发运营工程师需要使用工具将新的 AI/ML 应用程序集成到持续集成和持续部署（CI/CD）管道中。
- 业务用户希望能够访问 AI/ML 应用程序。

在本文档中，我们将介绍 Azure NetApp Files，Run：AI 和 Microsoft Azure 如何帮助这些角色为业务带来价值。

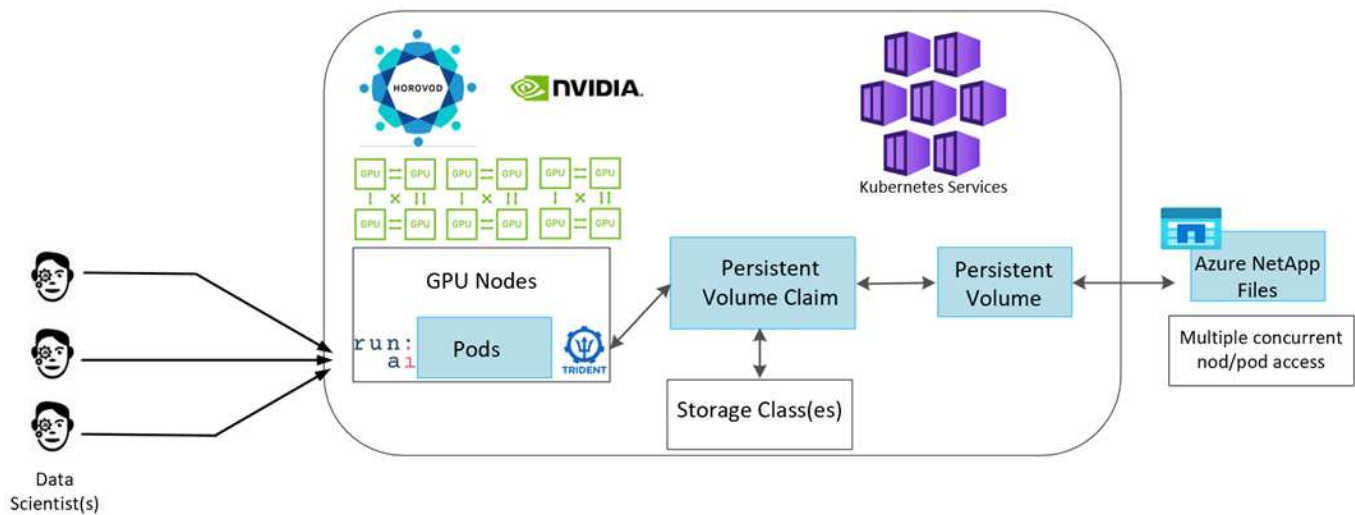
解决方案技术

本节介绍了在 Azure 云中全面运行的大规模分布式训练解决方案的通道检测用例的技术要求。下图概述了解决方案架构。

此解决方案中使用的元素包括：

- Azure Kubernetes Service （AKS）
- 采用 NVIDIA GPU 的 Azure 计算 SKU
- Azure NetApp Files
- 运行：AI
- NetApp Trident

中列出了指向此处提及的所有要素的链接 "追加信息" 部分。



云资源和服务要求

下表列出了实施解决方案所需的硬件组件。在任何解决方案实施中使用的云组件可能会因客户要求而异。

云	数量
AK	至少三个系统节点和三个 GPU 工作节点
虚拟机（VM）SKU 系统节点	三个 Standard_DS2_v2
VM SKU GPU 工作节点	三个 Standard_NC6s_v3
Azure NetApp Files	4 TB 标准层

软件要求

下表列出了实施解决方案所需的软件组件。在任何解决方案实施中使用的软件组件可能会因客户要求而异。

软件	版本或其他信息
AK — Kubernetes 版本	1.18.14
运行： AI 命令行界面	v2.2.25
运行： AI Orchestration Kubernetes Operator 版本	1.0.109
Horovod	0.21.2.
NetApp Trident	20.01.1
掌舵	3.0.0

车道检测—使用 Run : AI 进行分布式培训

本节详细介绍了如何设置平台，以便使用 run : AI orchator 大规模执行车道检测分布式培训。我们将讨论所有解决方案要素的安装以及在所述平台上运行分布式培训作业的问题。可以使用 NetApp Snapshot™ 并将其与 run : AI 实验相链接来完成 ML 版本控制，以实现数据和模型可重现性。在跟踪模型，团队成员之间共享工作，结果的可重现性，将新型号版本投入生产以及数据来源方面，ML 版本控制起着至关重要的作用。NetApp ML 版本控制（Snapshot）可以捕获与每个实验相关的数据，经过培训的模型和日志的时间点版本。它具有丰富的 API 支持，可以轻松地与运行： AI 平台集成；您只需根据训练状态触发事件即可。此外，您还必须捕获整个实验的状态，而不更改 Kubernetes（K8）上运行的代码或容器中的任何内容。

最后，本技术报告将对 AKS 中多个启用了 GPU 的节点进行性能评估。

针对使用 TuSimple 数据集的通道检测用例的分布式培训

在本技术报告中，对 TuSimple 数据集进行了分布式培训，用于检测通道。在本培训代码中，Horovod 用于通过 AKS 在 Kubernetes 集群中的多个 GPU 节点上同时执行数据分布式培训。代码作为容器映像打包，以供 TuSimple 数据下载和处理。处理后的数据存储在 NetApp Trident 插件分配的永久性卷上。在培训中，还会创建一个容器映像，并使用在下载数据期间创建的永久性卷上存储的数据。

要提交数据和培训作业，请使用 run : ai 编排资源分配和管理。Run : AI 允许您执行 Horovod 所需的消息传递接口（Message Passing Interface，MPI）操作。此布局允许多个 GPU 节点彼此通信，以便在每次训练迷你批处理后更新训练权重。此外，它还可以通过 UI 和 CLI 监控训练，从而轻松监控实验进度。

NetApp Snapshot 集成在培训代码中，可捕获每个实验的数据状态和经过培训的模型。通过此功能，您可以跟踪所用数据和代码的版本以及生成的相关培训模型。

AK 设置和安装

要设置和安装 AKS 集群，请转至 ["创建 AKS 集群"](#)。然后，按照以下一系列步骤进行操作：

1. 选择节点类型（无论是系统（CPU）节点还是辅助（GPU）节点）时，请选择以下项：

- 以 Standard_DS2_v2 大小添加名为 agentpool 的主系统节点。使用默认的三个节点。
- 添加工作节点 gpupool 并使用 Standard_Nc6s_v3 池大小。至少为 GPU 节点使用三个节点。

+ Add node pool Delete				
Name	Mode	OS type	Node count	Node size
<input type="checkbox"/> agentpool	System	Linux	3	Standard_DS2_v2
<input type="checkbox"/> gpupool	User	Linux	3	Standard_NC6s_v



部署需要 5 – 10 分钟。

- 部署完成后，单击 Connect to Cluster 。要连接到新创建的 AKS 集群，请从本地环境（笔记本电脑 /PC ）安装 Kubernetes 命令行工具。请访问 ["安装工具"](#) 以根据您的操作系统进行安装。
- ["在本地环境中安装 Azure CLI"](#)。
- 要从终端访问 AKS 集群，请先输入 az login 并输入凭据。
- 运行以下两个命令：

```
az account set --subscription xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxxxxxx
aks get-credentials --resource-group resourcegroup --name aksclustername
```

- 在 Azure 命令行界面中输入此命令：

```
kubectl get nodes
```



如果所有六个节点均按此处所示启动并运行，则 AKS 集群已准备就绪并连接到本地环境。

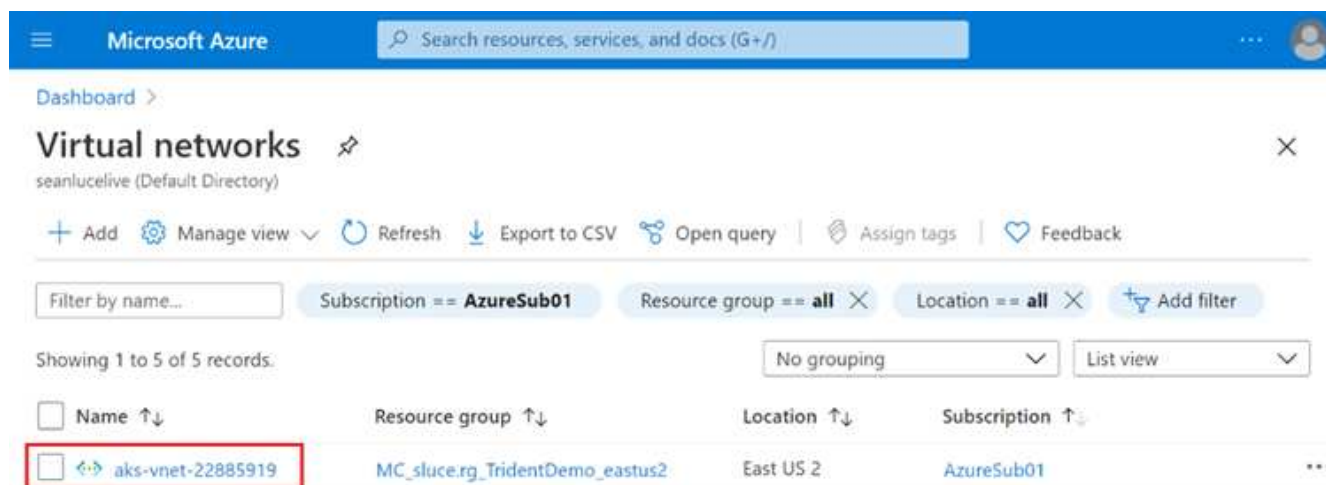
```
verronmartina@verron-mac-0 ~ % kubectl get nodes
NAME                                STATUS    ROLES    AGE   VERSION
aks-agentpool-34613062-vmss000000  Ready    agent    22m   v1.18.14
aks-agentpool-34613062-vmss000001  Ready    agent    22m   v1.18.14
aks-agentpool-34613062-vmss000002  Ready    agent    22m   v1.18.14
aks-gpupool-34613062-vmss000000     Ready    agent    20m   v1.18.14
aks-gpupool-34613062-vmss000001     Ready    agent    20m   v1.18.14
aks-gpupool-34613062-vmss000002     Ready    agent    20m   v1.18.14
verronmartina@verron-mac-0 ~ %
```

为 Azure NetApp Files 创建委派子网

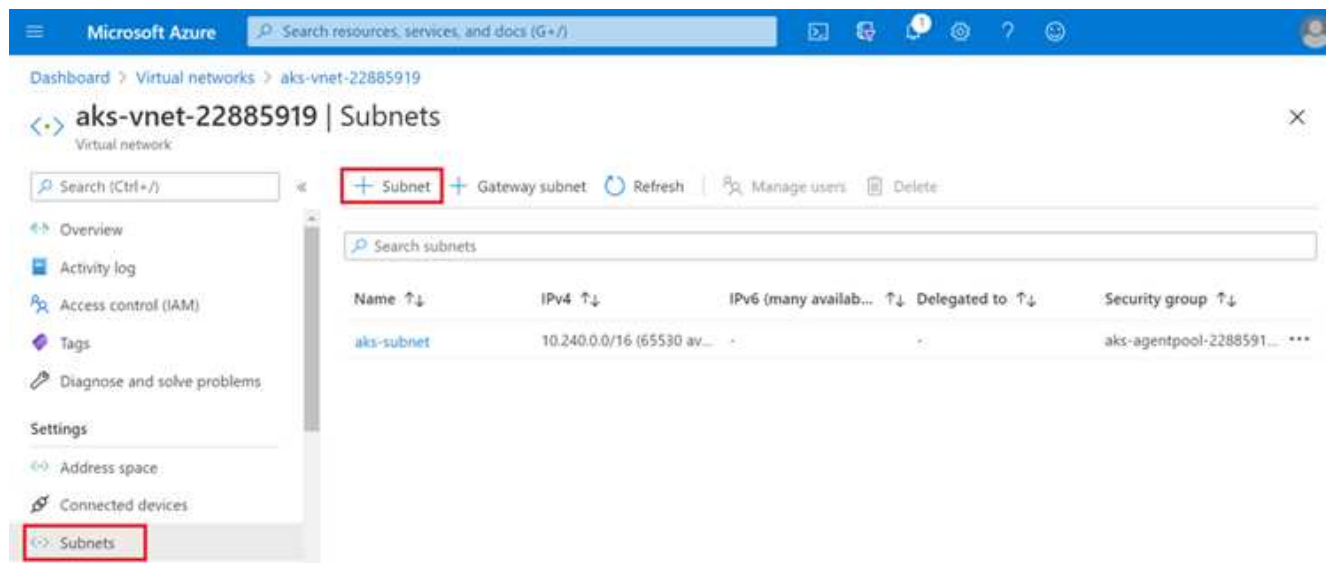
要为 Azure NetApp Files 创建委派子网，请执行以下一系列步骤：

- 导航到 Azure 门户中的虚拟网络。查找新创建的虚拟网络。它应具有前面板，如 AK vnet ，如此处所示。单

击虚拟网络的名称。



2. 单击子网，然后从顶部工具栏中选择 +Subnet 。



3. 为子网提供名称，例如 ANF.SN，然后在 Subnet delegation 标题下选择 Microsoft.NetApp/volumes。请勿更改任何其他内容。单击确定。

Add subnet



Name * ✓

Subnet address range * ⓘ
10.0.0.0 - 10.0.0.255 (251 + 5 Azure reserved addresses)

☐ Add IPv6 address space ⓘ

NAT gateway ⓘ ▼

Network security group ▼

Route table ▼

SERVICE ENDPOINTS

Create service endpoint policies to allow traffic to specific azure resources from your virtual network over service endpoints. [Learn more](#)

Services ⓘ ▼

SUBNET DELEGATION

Delegate subnet to a service ⓘ ▼

Azure NetApp Files 卷将分配给应用程序集群，并在 Kubernetes 中用作永久性卷声明（Persistent Volume Claim，PVC）。反过来，这种分配也为我们提供了将卷映射到不同服务的灵活性，包括 Jupyter 笔记本电脑，无服务器功能等

服务用户可以通过多种方式使用平台中的存储。Azure NetApp Files 的主要优势包括：

- 使用户能够使用快照。
- 允许用户在 Azure NetApp Files 卷上存储大量数据。
- 在一组大型文件上运行 Azure NetApp Files 卷的型号时，可以获得这些卷的性能优势。

Azure NetApp Files 设置

要完成 Azure NetApp Files 的设置，必须先按照中所述对其进行配置 "[快速入门：设置 Azure NetApp Files 并创](#)

建 NFS 卷"。

但是，您可以省略为 Azure NetApp Files 创建 NFS 卷的步骤，因为您将通过 Trident 创建卷。在继续操作之前，请确保您已：

- 1. "注册 Azure NetApp Files 和 NetApp 资源提供商（通过 Azure Cloud Shell ）"。
- 2. "已在 Azure NetApp Files 中创建帐户"。
- 3. "设置容量池"（最低 4 TiB 标准版或高级版，具体取决于您的需求）。

建立 AKS 虚拟网络和 Azure NetApp Files 虚拟网络对等关系

接下来，按照以下步骤将 AKS 虚拟网络（vNet）与 Azure NetApp Files vNet 建立对等关系：

- 1. 在 Azure 门户顶部的搜索框中，键入虚拟网络。
- 2. 单击 vNet AK - vnet-name，然后在搜索字段中输入 Peeids。
- 3. 单击 +Add，然后输入下表中提供的信息：

字段	值或问题描述
对等链路名称	aps-vnet-name_to_anf
subscriptionId	订阅要与之建立对等关系的 Azure NetApp Files vNet
vNet 对等配对节点	Azure NetApp Files vNet



保留所有非星号部分的默认设置

- 4. 单击添加或确定将对等添加到虚拟网络。

有关详细信息，请访问 "创建，更改或删除虚拟网络对等关系"。

Trident

Trident 是 NetApp 为应用程序容器永久性存储维护的一个开源项目。Trident 已作为外部配置程序控制器实施，该控制器本身作为 POD 运行，可监控卷并完全自动化配置过程。

NetApp Trident 通过创建和附加永久性卷来存储培训数据集和经过培训的模型，可以与 K8 平稳集成。借助此功能，数据科学家和数据工程师可以更轻松地使用 K8，而无需手动存储和管理数据集。Trident 还可以通过逻辑 API 集成将数据管理相关任务集成在一起，因此数据科学家无需学习管理新的数据平台。

安装 Trident

要安装 Trident 软件，请完成以下步骤：

- 1. "首先安装 Helm"。
- 2. 下载并解压缩 Trident 21.01.1 安装程序。

```
wget
https://github.com/NetApp/trident/releases/download/v21.01.1/trident-
installer-21.01.1.tar.gz
tar -xf trident-installer-21.01.1.tar.gz
```

3. 将目录更改为 trident 安装程序。

```
cd trident-installer
```

4. 将 tridentctl 复制到系统中的目录 `$path`。

```
cp ./tridentctl /usr/local/bin
```

5. 使用 Helm 在 K8s 集群上安装 Trident：

- a. 将目录更改为 helm 目录。

```
cd helm
```

- b. 安装 Trident。

```
helm install trident trident-operator-21.01.1.tgz --namespace trident
--create-namespace
```

- c. 按照通常的 K8s 方式检查 Trident Pod 的状态：

```
kubectl -n trident get pods
```

- d. 如果所有 Pod 均已启动且正在运行，则会安装 Trident，您可以继续操作。

设置 Azure NetApp Files 后端和存储类

要设置 Azure NetApp Files 后端和存储类，请完成以下步骤：

1. 切换回主目录。

```
cd ~
```

2. 克隆 ["项目存储库"](#) lan-detect-scnn-horovod。

3. 转至 `trident - config` 目录。

```
cd ./lane-detection-SCNN-horovod/trident-config
```

4. 创建 Azure 服务原则（服务原则是 Trident 如何与 Azure 通信以访问 Azure NetApp Files 资源）。

```
az ad sp create-for-rbac --name
```

输出应类似于以下示例：

```
{
  "appId": "xxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx",
  "displayName": "netapptrident",
  "name": "http://netapptrident",
  "password": "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx",
  "tenant": "xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx"
}
```

5. 创建 Trident backend json 文件。
6. 使用您的首选文本编辑器，填写 `anf-backend.json` 文件中下表中的以下字段。

字段	价值
subscriptionId	您的 Azure 订阅 ID
tenantId	您的 Azure 租户 ID （上一步 AZ AD sp 的输出）
clientId	您的应用程序 ID （来自上一步 AZ AD sp 的输出）
客户端机密	您的密码 （上一步 AZ AD sp 的输出）

此文件应类似于以下示例：

```
{
  "version": 1,
  "storageDriverName": "azure-netapp-files",
  "subscriptionID": "fakeec765-4774-fake-ae98-a721add4fake",
  "tenantID": "fakef836-edc1-fake-bff9-b2d865eefake",
  "clientID": "fake0f63-bf8e-fake-8076-8de91e57fake",
  "clientSecret": "SECRET",
  "location": "westeurope",
  "serviceLevel": "Standard",
  "virtualNetwork": "anf-vnet",
  "subnet": "default",
  "nfsMountOptions": "vers=3,proto=tcp",
  "limitVolumeSize": "500Gi",
  "defaults": {
    "exportRule": "0.0.0.0/0",
    "size": "200Gi"
  }
}
```

7. 指示 Trident 在 `trident` 命名空间中创建 Azure NetApp Files 后端，使用 `anf-backend.json` 作为配置文件，如下所示：

```
tridentctl create backend -f anf-backend.json -n trident
```

8. 创建存储类：

- a. K8 用户使用按名称指定存储类的 PVC 配置卷。指示 K8s 使用以下命令创建一个存储类 `azurenetafiles`，该存储类将引用上一步中创建的 Azure NetApp Files 后端：

```
kubectl create -f anf-storage-class.yaml
```

- b. 使用以下命令检查是否已创建存储类：

```
kubectl get sc azurenetafiles
```

输出应类似于以下示例：

NAME	PROVISIONER	RECLAIMPOLICY	VOLUMEBINDINGMODE	ALLOWVOLUMEEXPANSION	AGE
azurenetafiles	csi.trident.netapp.io	Delete	Immediate	false	98s

在 AKS 上部署和设置卷快照组件

如果集群未预安装正确的卷快照组件，则可以通过运行以下步骤手动安装这些组件：



1. 使用以下命令安装 Snapshot 测试版 CRD：

```
kubectl create -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-3.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshotclasses.yaml
kubectl create -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-3.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshotcontents.yaml
kubectl create -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-3.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshots.yaml
```

2. 使用 GitHub 中的以下文档安装 Snapshot 控制器：

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-3.0/deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-3.0/deploy/kubernetes/snapshot-controller/setup-snapshot-controller.yaml
```

3. 设置 K8s volumesnapshotclass：创建卷快照之前，请先执行 ["卷快照类"](#) 必须已设置。为 Azure NetApp Files 创建卷快照类，并使用它通过 NetApp Snapshot 技术实现 ML 版本控制。create volumesnapshotclass netapp-csi-snapclass 并将其设置为 default `volumesnapshotclass`，如下所例：

```
kubectl create -f netapp-volume-snapshot-class.yaml
```

输出应类似于以下示例：

```
volumesnapshotclass.snapshot.storage.k8s.io/netapp-csi-snapclass created
```

4. 使用以下命令检查是否已创建卷 Snapshot 副本类：

```
kubectl get volumesnapshotclass
```

输出应类似于以下示例：

NAME	DRIVER	DELETIONPOLICY	AGE
netapp-csi-snapclass	csi.trident.netapp.io	Delete	63s

运行：AI 安装

要安装 run : ai ，请完成以下步骤：

1. "在 AKS 上安装 run : ai 集群"。
2. 转至 app.runai.ai ，单击创建新项目，然后将其命名为 LAN-detection 。它将在 K8s 集群上创建一个命名空间，其开头为 runai - ，后跟项目名称。在这种情况下，创建的命名空间将为 runai-lane 检测。

New Project

Basics

Node Affinity

Time Limit

Basics

Project Name ⓘ

lane-detection

Assigned GPUs

3

Over-quota for project

☒ Allow over-quota

Save Cancel

3. "安装 run : ai 命令行界面"。
4. 在您的终端上，使用以下命令将通道检测设置为默认运行： AI project :

```
`runai config project lane-detection`
```

输出应类似于以下示例：

```
Project lane-detection has been set as default project
```

5. 为项目命名空间创建 ClusterRole 和 ClusterRoleBinding（例如，LANE-detection）因此，属于 runai-lan-detection namespace 的默认服务帐户有权在作业执行期间执行 volumesnapshot 操作：

- a. 使用以下命令列出命名空间以检查 runai-lan-detection 是否存在：

```
kubectl get namespaces
```

输出应类似于以下示例：

NAME	STATUS	AGE
default	Active	130m
kube-node-lease	Active	130m
kube-public	Active	130m
kube-system	Active	130m
runai	Active	4m44s
runai-lan-detection	Active	13s
trident	Active	102m

6. 使用以下命令创建 ClusterRole netappsnapshot 和 ClusterRoleBinding`netappsnapshot`：

```
`kubectl create -f runai-project-snap-role.yaml`  
`kubectl create -f runai-project-snap-role-binding.yaml`
```

下载并将 TuSimple 数据集作为 run : ai 作业处理

下载并处理运行时的 TuSimple 数据集的过程：AI 作业是可选的。其中包括以下步骤：

1. 构建并推送 Docker 映像，或者如果要使用现有 Docker 映像（例如，muneer7589/download-tusimple : 1.0），则省略此步骤

- a. 切换到主目录：

```
cd ~
```

- b. 转到项目的数据目录 lan-detect-scnn-horovod：

```
cd ./lane-detection-SCNN-horovod/data
```

- c. 修改 `build_image.sh` shell 脚本并将 Docker 存储库更改为您的。例如，将 `muneer7589` 替换为 Docker 存储库名称。您还可以更改 Docker 映像名称和标记（例如 `download-tusimple` 和 `1.0`）：

```
#!/bin/bash
#
# A simple script to build the Docker image.
#
# $ build_image.sh
set -ex

IMAGE=muneer7589/download-tusimple
TAG=1.0

# Build image
echo "Building image: "$IMAGE
docker build . -f Dockerfile \
  --tag "${IMAGE}:${TAG}"
echo "Finished building image: "$IMAGE

# Push image
echo "Pushing image: "$IMAGE
docker push "${IMAGE}:${TAG}"
echo "Finished pushing image: "$IMAGE
```

- d. 运行脚本以构建 Docker 映像，并使用以下命令将其推送到 Docker 存储库：

```
chmod +x build_image.sh
./build_image.sh
```

2. 提交运行：AI 作业，以下载，提取，预处理并将 Tuple 通道检测数据集存储在一个 PVC 中，该 PVC 由 NetApp Trident 动态创建：

- a. 使用以下命令提交运行：AI 作业：

```
runai submit
--name download-tusimple-data
--pvc azurenetappfiles:100Gi:/mnt
--image muneer7589/download-tusimple:1.0
```

- b. 输入下表中的信息以提交运行：AI 作业：

字段	值或问题描述
name	作业的名称

字段	值或问题描述
-pvc	PVC 格式为 [StorageClassName] : size : ContainerMountPath 在上述作业提交中，您正在使用具有存储类 azurenetappfiles 的 Trident 根据需要创建 PVC 。此处的永久性卷容量为 100Gi ，并挂载在路径 /mnt 处。
图像	创建此作业的容器时要使用的 Docker 映像

输出应类似于以下示例：

```
The job 'download-tusimple-data' has been submitted successfully
You can run `runai describe job download-tusimple-data -p lane-detection` to check the job status
```

- c. 列出已提交的运行：AI 作业。

```
runai list jobs
```

```
Showing jobs for project lane-detection
NAME          STATUS      SERVICE URL(S)  AGE  NODE          IMAGE                                     TYPE  PROJECT      USER          GPUs Allocated (Requested)
PODs Running (Pending)
download-tusimple-data  ContainerCreating  1m   aks-agentpool-34613862-vmss00000a  muneer7589/download-tusimple:1.0  Train  lane-detection  veronmartina  0 (0)
```

- d. 检查提交的作业日志。

```
runai logs download-tusimple-data -t 10
```

```
751150K ..... 6% 16.2M 20m37s
751200K ..... 6% 11.1M 20m37s
751250K ..... 6% 12.5M 20m36s
751300K ..... 6% 11.3M 20m36s
751350K ..... 6% 15.2M 20m36s
751400K ..... 6% 10.5M 20m36s
751450K ..... 6% 15.2M 20m36s
751500K ..... 6% 14.1M 20m36s
751550K ..... 6% 24.3M 20m36s
751600K ..... 6% 26.3M 20m36s
```

- e. 列出已创建的 PVC 。在下一步中使用此 PVC 命令进行培训。

```
kubectl get pvc | grep download-tusimple-data
```

输出应类似于以下示例：

```
pvc-download-tusimple-data-0   Bound          pvc-bb03b74d-2c17-40c4-a445-79f3de8d16d5   100Gi   RWO          azurenetappfiles   4m47s
```

- a. 在 run : ai UI （或 app.run.ai ）中检查作业。

run: a.1 Jobs										Cluster: runaiaks	Muneer.Ahmad@netapp.com	NetApp
Filter and Search												
Job Name	Status	User	Project	Total Run Time	Creation Time	Type	GPU Utilization	Used CPU				
download-tusimple-data	Running	verronma...	lane-detection	00:07:11	03/03/21, 2:51PM	Train	-	0.00	0			
build1	Deleted	root	lane-detection	00:01:56	03/01/21, 10:18...	Interactive	-	-	-			
download-tusimple-data	Deleted	root	lane-detection	-	03/01/21, 9:58AM	Train	-	-	-			
download-tusimple-data	Deleted	root	lane-detection	-	03/01/21, 10:03...	Train	-	-	-			
download-tusimple-data	Deleted	root	lane-detection	00:02:55	03/01/21, 10:24...	Train	-	-	-			
download-tusimple-data	Deleted	root	lane-detection	-	03/01/21, 10:30...	Train	-	-	-			
download-tusimple-data	Deleted	root	lane-detection	00:13:17	03/01/21, 11:41...	Train	-	-	-			
download-tusimple-data-1	Deleted	verronma...	lane-detection	-	02/26/21, 5:30PM	Train	-	-	-			

使用 Horovod 执行分布式通道检测培训

使用 Horovod 执行分布式通道检测培训是一个可选过程。但是，需要执行以下步骤：

1. 构建并推送 Docker 映像，或者如果要使用现有 Docker 映像（例如，muneer7589/dist-lan-detection : 3.1 ），请跳过此步骤：

- a. 切换到主目录。

```
cd ~
```

- b. 转到项目目录 lan-detect-scann-horovod.

```
cd ./lane-detection-SCNN-horovod
```

- c. 修改 build_image.sh shell 脚本并将 Docker 存储库更改为您的（例如，将 muneer7589 替换为您的 Docker 存储库名称）。您也可以更改 Docker 映像名称和标记（例如，dist-lan-detection 和 3.1 ）。

```
#!/bin/bash
#
# A simple script to build the distributed Docker image.
#
# $ build_image.sh
set -ex

IMAGE=muneer7589/dist-lane-detection
TAG=3.0

# Build image
echo "Building image: "$IMAGE
docker build . -f Dockerfile \
  --tag "${IMAGE}:${TAG}"
echo "Finished building image: "$IMAGE

# Push image
echo "Pushing image: "$IMAGE
docker push "${IMAGE}:${TAG}"
echo "Finished pushing image: "$IMAGE
```

- d. 运行脚本以构建 Docker 映像并推送到 Docker 存储库。

```
chmod +x build_image.sh
./build_image.sh
```

2. 提交 Run：AI 作业以执行分布式培训（MPI）：

- 使用提交运行：AI 在上一步中自动创建 PVC（用于下载数据）仅允许您访问 RW，这样不允许多个 Pod 或节点在分布式培训中访问同一 PVC。将访问模式更新为 ReadWriteMany，然后使用 Kubernetes 修补程序执行此操作。
- 首先，运行以下命令以获取 PVC 的卷名称：

```
kubectl get pvc | grep download-tusimple-data
```

```
root@ai-w-gpu-2:/mnt/ai_data/anf_runai/lane-detection-SCNN-horovod# kubectl get pvc | grep download-tusimple-data
pvc-download-tusimple-data-0   Bound          pvc-bb03b74d-2c17-40c4-a445-79f3de8d16d5   100Gi   RWX           azurenetappfiles   2d4h
```

- 修补卷并将访问模式更新为 ReadWriteMany（在以下命令中将卷名称替换为您的）：

```
kubectl patch pv pvc-bb03b74d-2c17-40c4-a445-79f3de8d16d5 -p
'{"spec":{"accessModes":["ReadWriteMany"]}}'
```

- 使用下表中的信息提交运行：AI MPI 作业以执行分布式培训`作业：

```

runai submit-mpi
--name dist-lane-detection-training
--large-shm
--processes=3
--gpu 1
--pvc pvc-download-tusimple-data-0:/mnt
--image muneer7589/dist-lane-detection:3.1
-e USE_WORKERS="true"
-e NUM_WORKERS=4
-e BATCH_SIZE=33
-e USE_VAL="false"
-e VAL_BATCH_SIZE=99
-e ENABLE_SNAPSHOT="true"
-e PVC_NAME="pvc-download-tusimple-data-0"

```

字段	值或问题描述
name	分布式培训作业的名称
大型 shm	挂载大型 /dev/shm 设备这是一个挂载在 RAM 上的共享文件系统，可为多个 CPU 工作人员提供足够大的共享内存来处理批处理并将其加载到 CPU RAM 中。
流程	分布式培训流程的数量
GPU	要为此作业中的作业分配的 GPU/ 进程数，有三个 GPU 工作进程（-processes=3），每个进程都分配有一个 GPU（-GPU 1）
PVC	使用由先前作业（download-tusimple 数据）创建并挂载到路径 /mnt 的现有永久性卷（vpvc 下载 -tusimple 数据 0）
图像	创建此作业的容器时要使用的 Docker 映像
定义要在容器中设置的环境变量	
use_works.	如果将参数设置为 true，则会启用多进程数据加载
num_works.	数据加载程序工作进程的数量
batch_size	训练批大小
使用 VAL	如果将参数设置为 true，则可以进行验证
Val_batch_size	验证批处理大小
enable_snapshot	如果将参数设置为 true，则可以为 ML 版本控制创建数据和经过培训的模型快照
PVC_NAME	要为其创建快照的 PVC 的名称。在提交的上述作业中，您将创建由数据集和经过培训的模型组成的 PVC-download-tusimple data-0 的快照

输出应类似于以下示例：

```
The job 'dist-lane-detection-training' has been submitted successfully
You can run 'runai describe job dist-lane-detection-training -p lane-detection' to check the job status
```

e. 列出已提交的作业。

```
runai list jobs
```

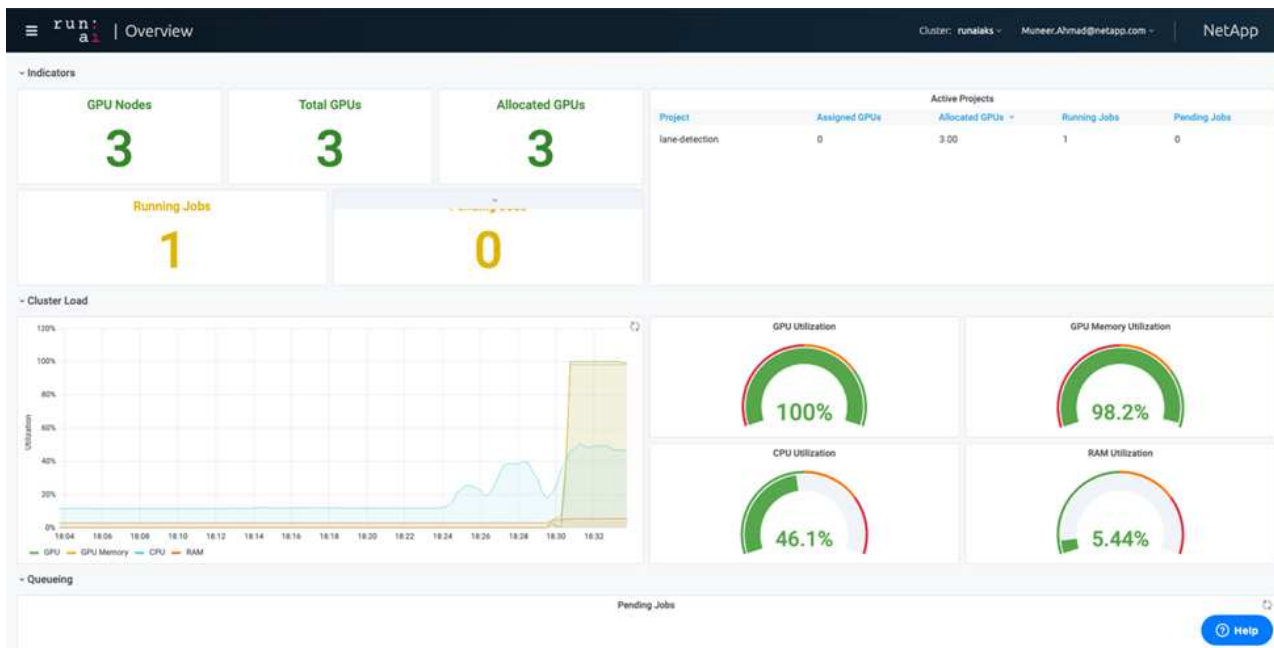
NAME	SERVICE URL(S)	STATUS	AGE	NODE	IMAGE	TYPE	PROJECT	USER	GPUs Allocated (Requested)	PODs
download-tusimple-data		Succeeded	1d		muneer7589/download-tusimple:1.0	Train	lane-detection	verronmartina	0 (0)	0 (0)
dist-lane-detection-training		Init:0/1	2m	<multiple>	muneer7589/dist-lane-detection:3.1	Train	lane-detection	root	3 (3)	4 (0)

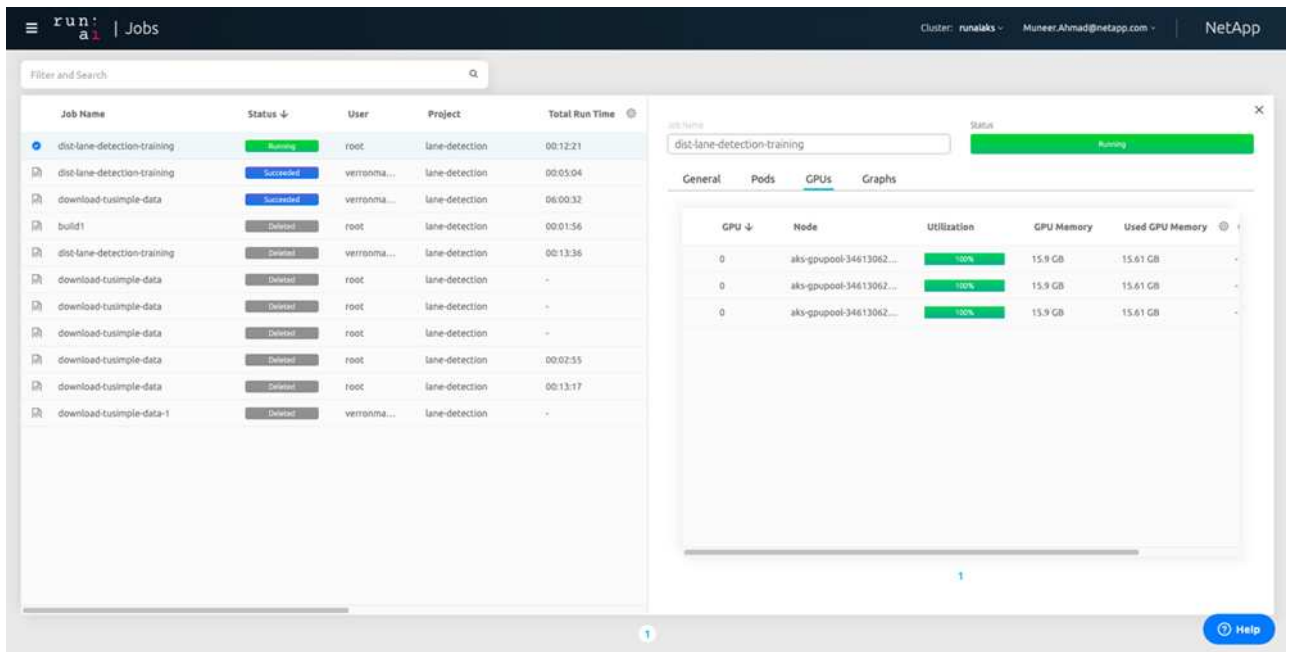
f. 已提交作业日志：

```
runai logs dist-lane-detection-training
```

```
root@ai-w-gpu-2:~/runai# runai logs dist-lane-detection-training
Running with 3 workers
2021-03-04 17:29:23.158449: I tensorflow/stream_executor/platform/default/dso_loader.cc:48] Successfully opened dynamic library libcudart.so.10.1
+ POD_NAME=dist-lane-detection-training-worker-0
+ [ d = - ]
+ shift
+ /opt/kube/kubect1 cp /opt/kube/hosts dist-lane-detection-training-worker-0:/etc/hosts_of_nodes
+ POD_NAME=dist-lane-detection-training-worker-2
+ [ d = - ]
+ shift
+ /opt/kube/kubect1 cp /opt/kube/hosts dist-lane-detection-training-worker-2:/etc/hosts_of_nodes
+ POD_NAME=dist-lane-detection-training-worker-1
```

g. 查看 Run 中的培训作业：AI GUI（或 app.runai.ai): run : AI Dashboard，如下图所示。第一个图详细介绍了为分布在 AKS 三个节点上的分布式培训作业分配的三个 GPU，以及第二个运行：AI 作业：





h. 完成培训后，请检查创建的 NetApp Snapshot 副本，并将其与 run : ai 作业链接在一起。

```
runai logs dist-lane-detection-training --tail 1
```

```
[1,0]<stdout>:Snapshot snap-pvc-download-tusimple-data-0-dist-lane-detection-training-launcher-2021-03-05-16-23-42 created in namespace runai-lane-detection
```

```
kubectl get volumesnapshots | grep download-tusimple-data-0
```

从 NetApp Snapshot 副本还原数据

要从 NetApp Snapshot 副本还原数据，请完成以下步骤：

1. 切换到主目录。

```
cd ~
```

2. 转到项目目录 lane-detect-scnn-horovod。

```
cd ./lane-detection-SCNN-horovod
```

3. 修改 restore-snapshot-vc.yaml 并将 dataSource name 字段更新到要从中还原数据的 Snapshot 副本。您也可以更改要将数据还原到的 PVC 名称，在此示例中为其 restored-tusimple。

```

apiVersion: v1
kind: PersistentVolumeClaim
metadata:
  name: restored-tusimple
spec:
  storageClassName: azurenetappfiles
  dataSource:
    name: snap-pvc-download-tusimple-data-0-dist-lane-detection-training-launcher-2021-03-05-16-23-42
    kind: VolumeSnapshot
    apiGroup: snapshot.storage.k8s.io
  accessModes:
    - ReadWriteMany
  resources:
    requests:
      storage: 100Gi

```

4. 使用 `restore-snapshot-vc.yaml` 创建新的 PVC。

```
kubectl create -f restore-snapshot-pvc.yaml
```

输出应类似于以下示例：

```
persistentvolumeclaim/restored-tusimple created
```

5. 如果您要使用刚刚还原的数据进行培训，则作业提交将保持不变；在提交培训作业时，只需将 `vc_name` 替换为已还原的 `vc_name`，如以下命令所示：

```

runai submit-mpi
--name dist-lane-detection-training
--large-shm
--processes=3
--gpu 1
--pvc restored-tusimple:/mnt
--image muneer7589/dist-lane-detection:3.1
-e USE_WORKERS="true"
-e NUM_WORKERS=4
-e BATCH_SIZE=33
-e USE_VAL="false"
-e VAL_BATCH_SIZE=99
-e ENABLE_SNAPSHOT="true"
-e PVC_NAME="restored-tusimple"

```

性能评估

为了显示解决方案的线性可扩展性，我们对以下两种情形进行了性能测试：一个 GPU 和三个 GPU。在有关 TuSimple 通道检测数据集的培训中，我们捕获了 GPU 分配，GPU 和内存利用率，不同的单节点和三节点指标。为了分析培训过程中的资源利用率，数据增加了五倍。

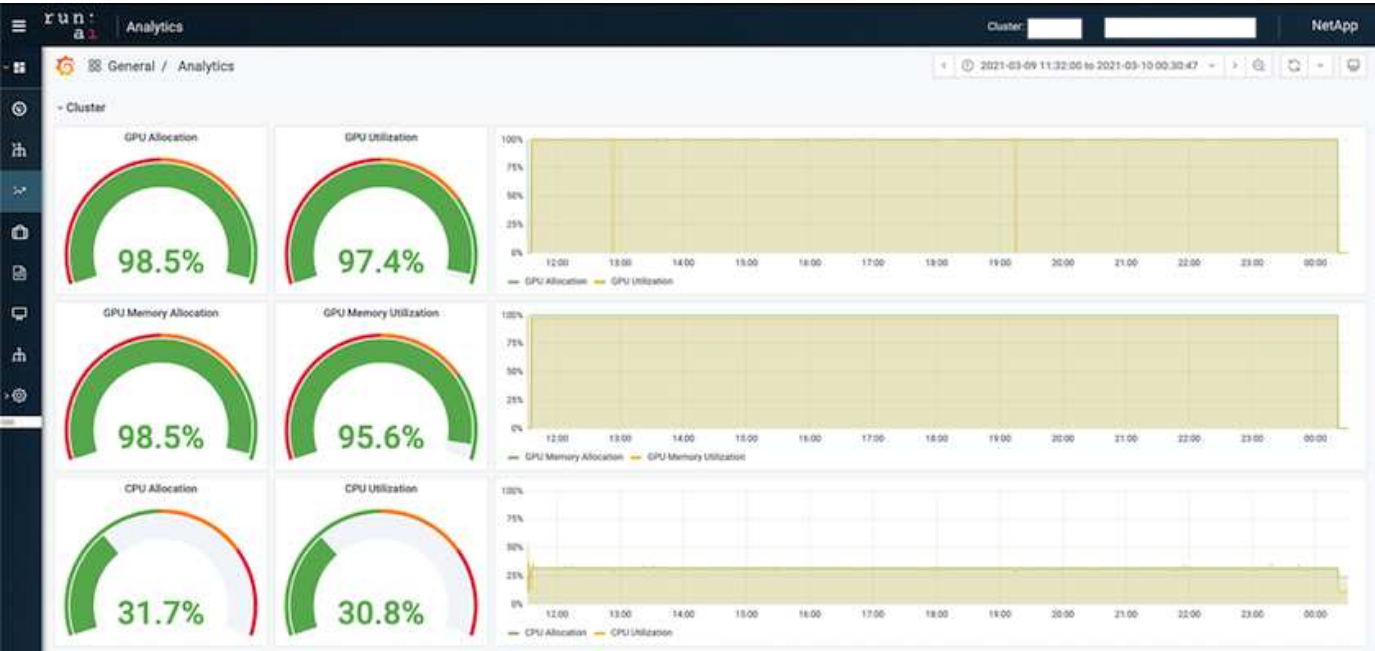
借助解决方案，客户可以从一个小型数据集和几个 GPU 入手。当数据量和 GPU 需求增加时，客户可以动态地横向扩展标准层中的 TB，并快速扩展到高级层，从而在不移动任何数据的情况下获得每 TB 吞吐量的四倍。本

节将进一步介绍此过程。"Azure NetApp Files 服务级别"。

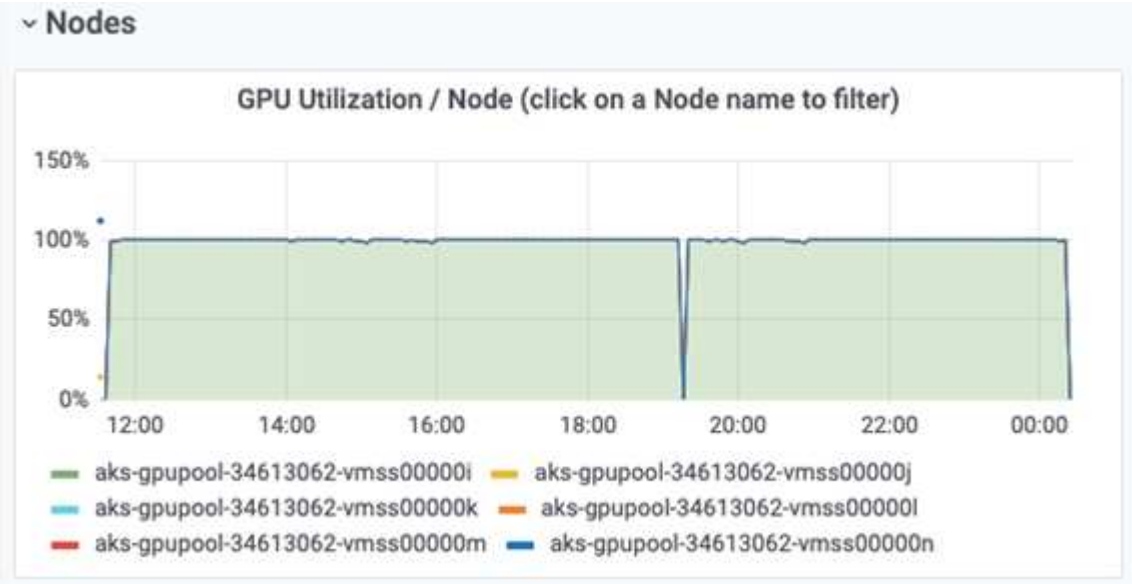
一个 GPU 的处理时间为 12 小时 45 分钟。三个节点上的三个 GPU 的处理时间约为 4 小时 30 分钟。

本文档其余部分中显示的图说明了根据各个业务需求提供的性能和可扩展性示例。

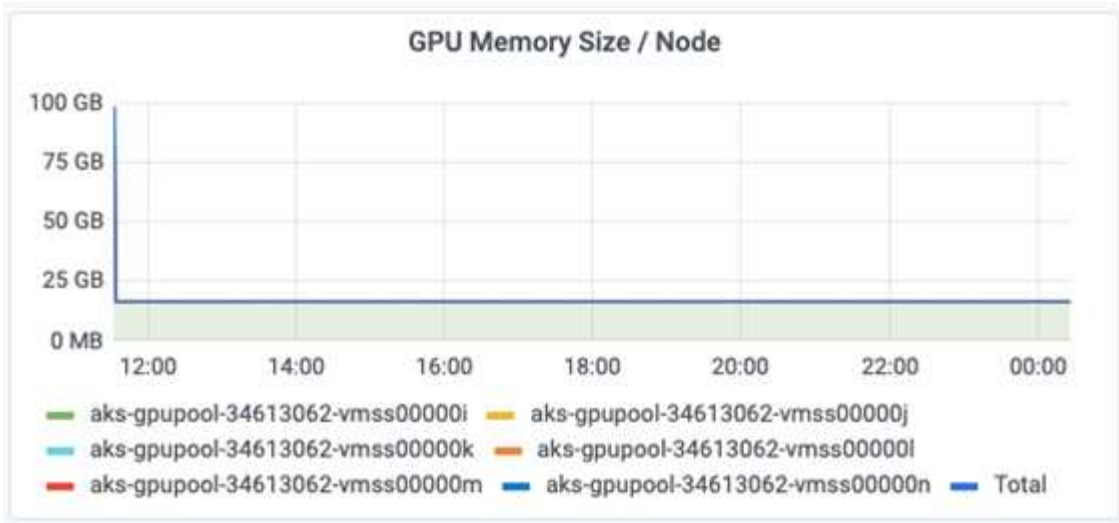
下图显示了 1 个 GPU 分配和内存利用率。



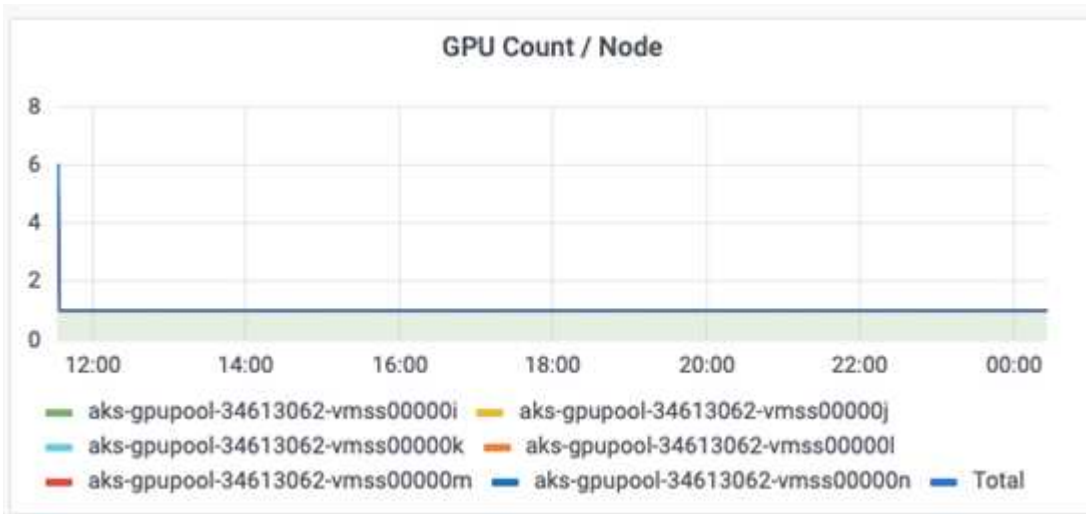
下图显示了单节点 GPU 利用率。



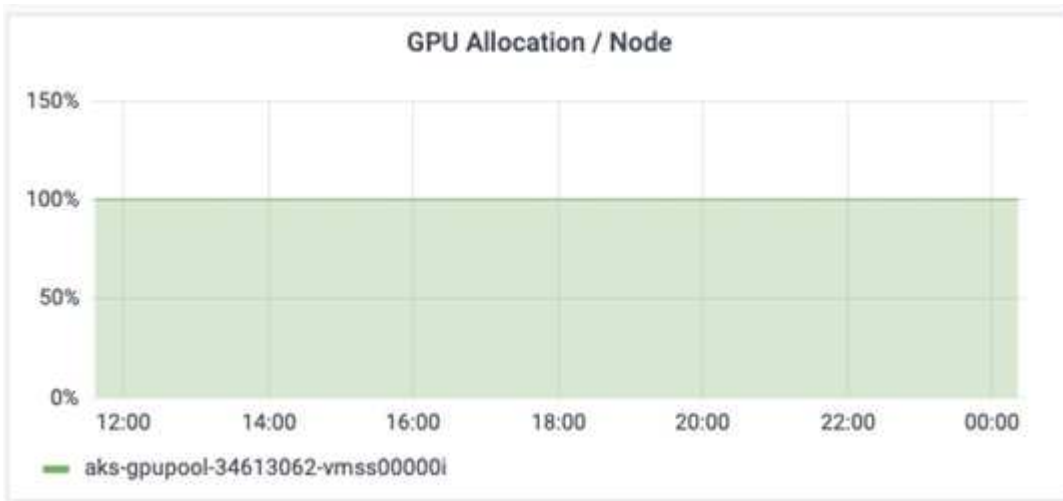
下图显示了单节点内存大小（16 GB）。



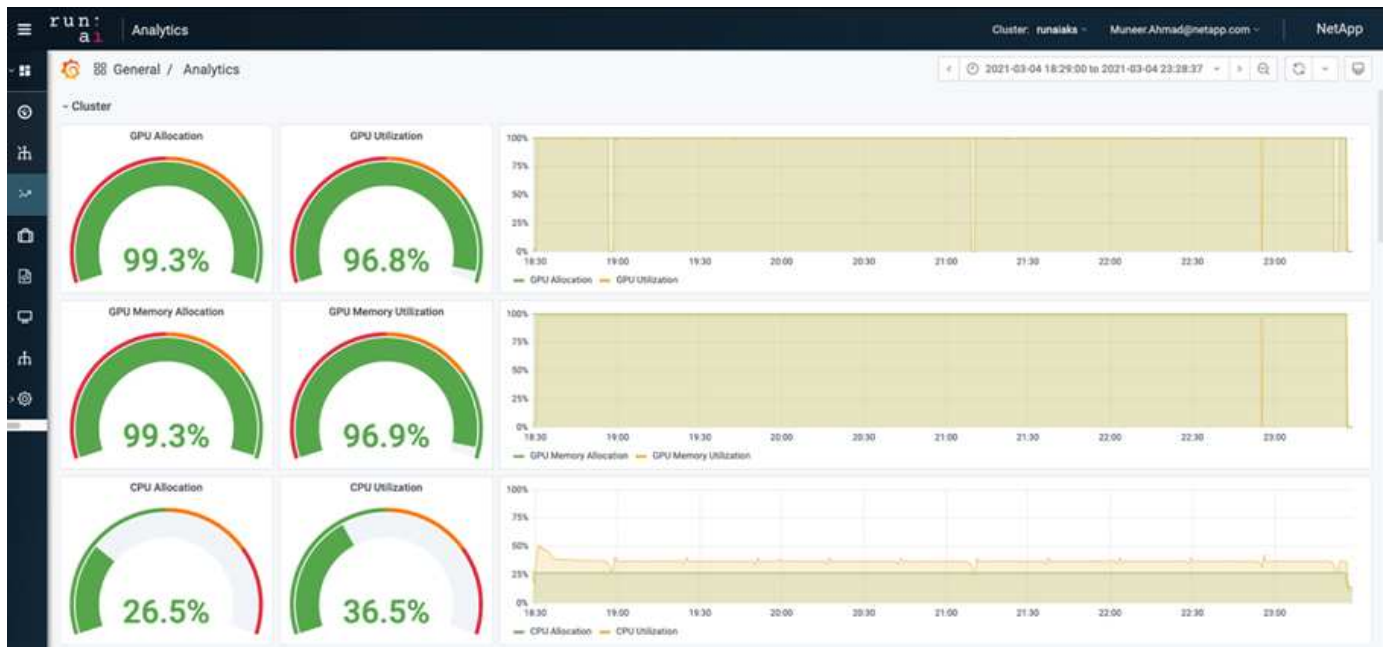
下图显示了单节点 GPU 计数（1）。



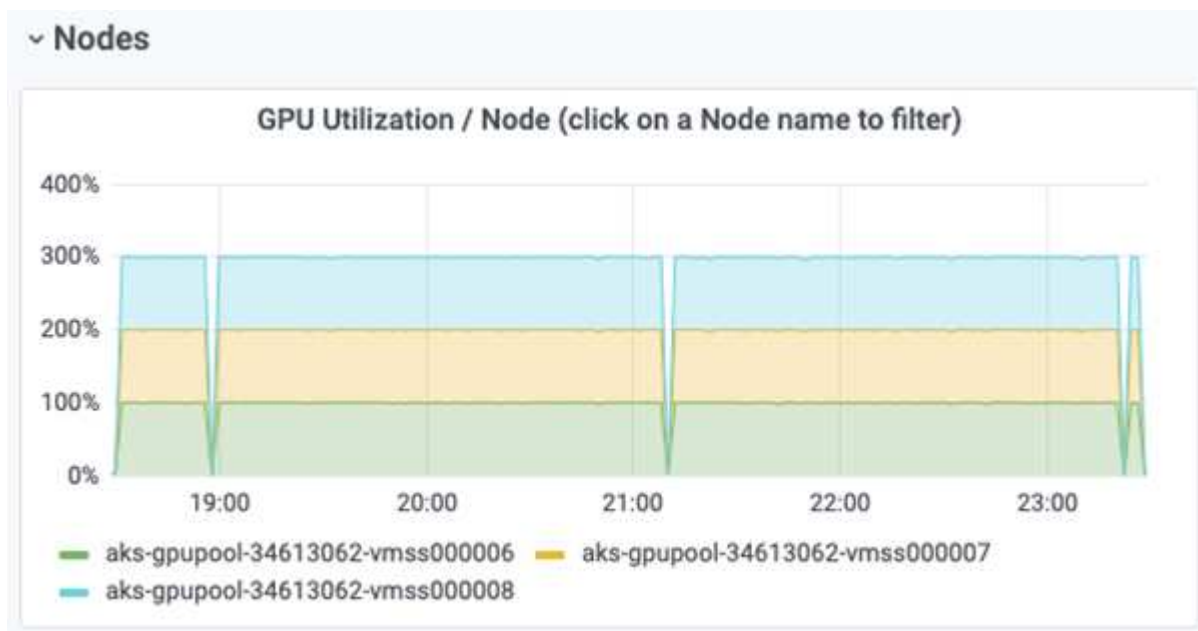
下图显示了单节点 GPU 分配（%）。



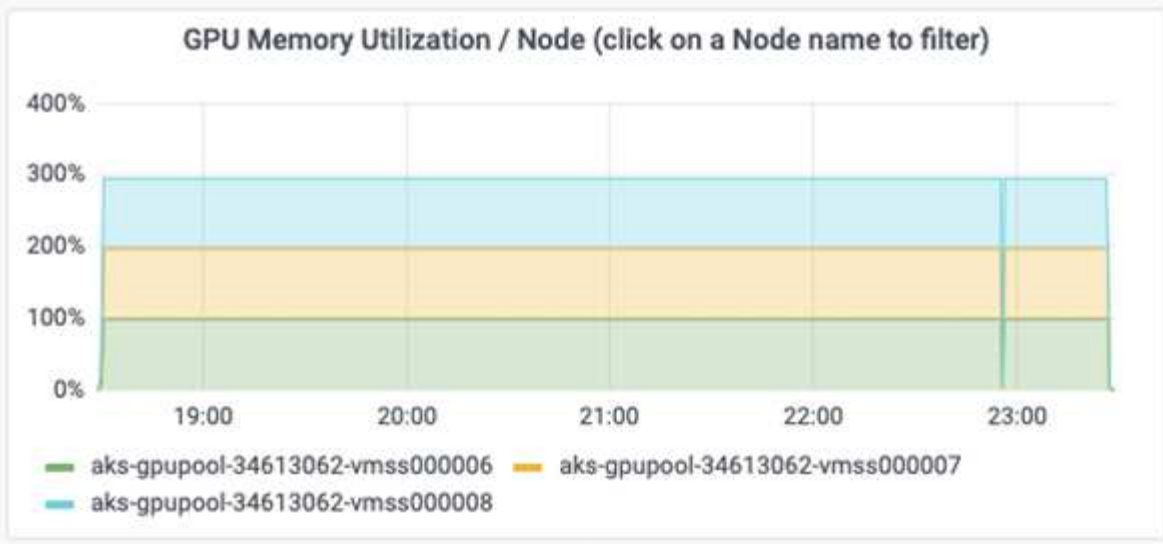
下图显示了三个节点上的三个 GPU — GPU 分配和内存。



下图显示了三个节点的三个 GPU 利用率（%）。



下图显示了三个节点的三个 GPU 内存利用率（%）。



Azure NetApp Files 服务级别

您可以通过将现有卷移动到使用的另一个容量池来更改此卷的服务级别 "服务级别" 所需的卷。此卷的现有服务级别更改不需要迁移数据。它也不会影响对卷的访问。

动态更改卷的服务级别

要更改卷的服务级别，请执行以下步骤：

1. 在卷页面上，右键单击要更改其服务级别的卷。选择更改池。

NFSv3	10.28.254.4:/norootfor	Standard	pool0	...
NFSv4.1	NAS-735a.docs.lab:/fo	Premium		...
NFSv4.1	NAS-735a.docs.lab:/krt	Premium		...
NFSv3	10.28.254.4:/moveme0	Premium		...
NFSv3	10.28.254.4:/placeholder	Premium		...

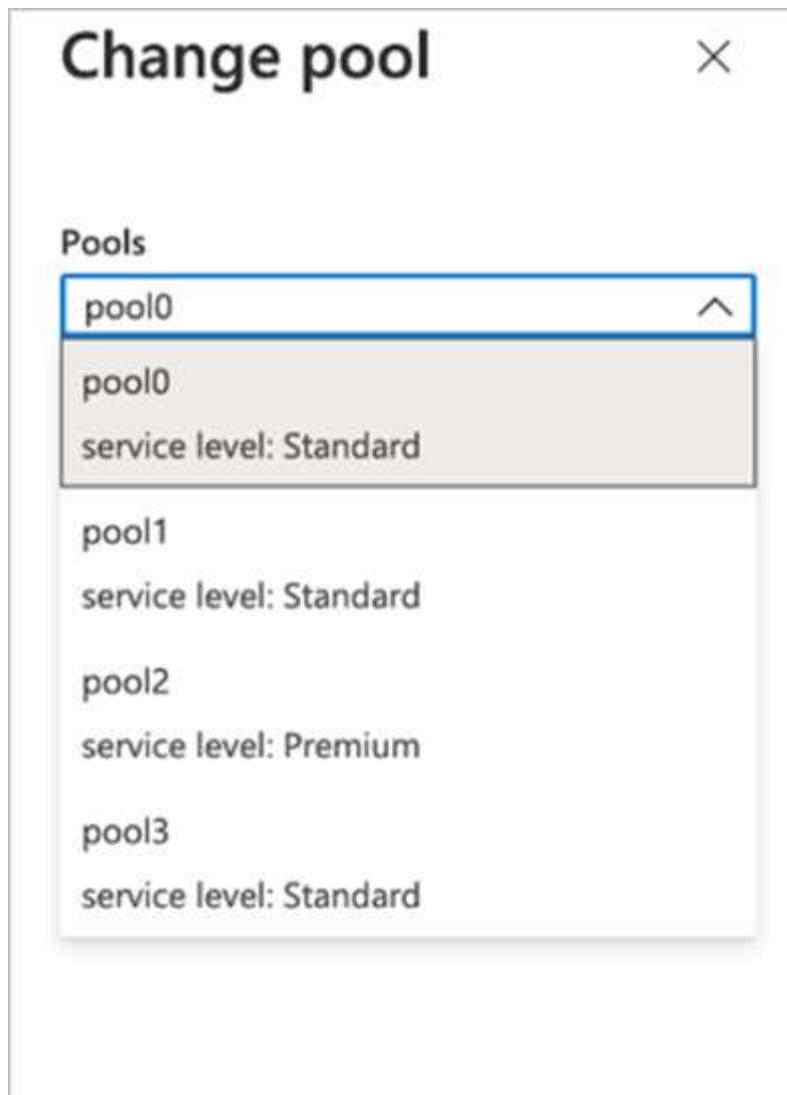
Resize

Edit

Change pool

Delete

2. 在更改池窗口中，选择要将卷移动到的容量池。然后，单击确定。



自动执行服务级别更改

动态服务级别更改当前仍在公有预览中，但默认情况下不会启用。要在 Azure 订阅上启用此功能，请按照文档 [中提供的步骤进行操作 "动态更改卷的服务级别"](#)。"

- 您还可以对 Azure 使用以下命令：CLI。有关更改 Azure NetApp Files 的池大小的详细信息，请访问 ["AZ netappfiles volume：管理 Azure NetApp Files（ANF）卷资源"](#)。

```
az netappfiles volume pool-change -g mygroup
--account-name myaccname
-pool-name mypoolname
--name myvolname
--new-pool-resource-id mynewresourceid
```

- 此处显示的 `set-aznetappfilesvolumepool cmdlet` 可更改 Azure NetApp Files 卷的池。有关更改卷池大小和 Azure PowerShell 的详细信息，请访问 ["更改 Azure NetApp Files 卷的池"](#)。

```
Set-AzNetAppFilesVolumePool
-ResourceGroupName "MyRG"
-AccountName "MyAnfAccount"
-PoolName "MyAnfPool"
-Name "MyAnfVolume"
-NewPoolResourceId 7d6e4069-6c78-6c61-7bf6-c60968e45fbf
```

结论

NetApp 与 Run： AI 合作创建了本技术报告，展示了 Azure NetApp Files 的独特功能以及用于简化 AI 工作负载流程的 Run： AI 平台。本技术报告提供了一个参考架构，用于简化分布式通道检测培训的数据管道和工作负载流程编排流程。

总之，对于大规模分布式培训（尤其是在公有云环境中），资源编排和存储组件是解决方案的重要组成部分。确保数据管理不会妨碍多个 GPU 处理，从而实现 GPU 周期的最佳利用率。这样，就可以使该系统尽可能地经济高效地用于大规模分布式培训。

NetApp 提供的 Data Fabric 可以帮助数据科学家和数据工程师在内部和云中相互连接，以实现同步数据，而无需执行任何手动干预，从而克服了这一挑战。换言之，Data Fabric 可以平稳地管理分布在多个位置的 AI 工作流。此外，它还可以将数据贴近计算，并在需要时随时随地执行分析，培训和验证，从而促进基于需求的数据可用性。此功能不仅可以实现数据集成，还可以保护和保障整个数据管道的安全。

追加信息

要了解有关本文档中所述信息的更多信息，请查看以下文档和 / 或网站：

- 数据集： TuSimple

["https://github.com/TuSimple/tusimple-benchmark/tree/master/doc/lane_detection"](https://github.com/TuSimple/tusimple-benchmark/tree/master/doc/lane_detection)

- 深度学习网络架构：空间对流神经网络

["https://arxiv.org/abs/1712.06080"](https://arxiv.org/abs/1712.06080)

- 分布式深度学习培训框架： Horovod

["https://horovod.ai/"](https://horovod.ai/)

- 运行： AI 容器编排解决方案：运行： AI 产品简介

["https://docs.run.ai/home/components/"](https://docs.run.ai/home/components/)

- 运行： AI 安装文档

["https://docs.run.ai/Administrator/Cluster-Setup/cluster-install/#step-3-install-runai"](https://docs.run.ai/Administrator/Cluster-Setup/cluster-install/#step-3-install-runai)

["https://docs.run.ai/Administrator/Researcher-Setup/cli-install/#runai-cli-installation"](https://docs.run.ai/Administrator/Researcher-Setup/cli-install/#runai-cli-installation)

- 在运行中提交作业： AI CLI

["https://docs.run.ai/Researcher/cli-reference/runai-submit/"](https://docs.run.ai/Researcher/cli-reference/runai-submit/)

["https://docs.run.ai/Researcher/cli-reference/runai-submit-mpi/"](https://docs.run.ai/Researcher/cli-reference/runai-submit-mpi/)

- Azure 云资源: Azure NetApp Files

["https://docs.microsoft.com/azure/azure-netapp-files/"](https://docs.microsoft.com/azure/azure-netapp-files/)

- Azure Kubernetes Service

["https://azure.microsoft.com/services/kubernetes-service/-features"](https://azure.microsoft.com/services/kubernetes-service/-features)

- Azure VM SKUs

["https://azure.microsoft.com/services/virtual-machines/"](https://azure.microsoft.com/services/virtual-machines/)

- 采用 GPU SKU 的 Azure VM

["https://docs.microsoft.com/azure/virtual-machines/sizes-gpu"](https://docs.microsoft.com/azure/virtual-machines/sizes-gpu)

- NetApp Trident

["https://github.com/NetApp/trident/releases"](https://github.com/NetApp/trident/releases)

- 由 NetApp 提供支持的 Data Fabric

["https://www.netapp.com/data-fabric/what-is-data-fabric/"](https://www.netapp.com/data-fabric/what-is-data-fabric/)

- NetApp 产品文档

["https://www.netapp.com/support-and-training/documentation/"](https://www.netapp.com/support-and-training/documentation/)

版权信息

版权所有 © 2024 NetApp, Inc.。保留所有权利。中国印刷。未经版权所有者事先书面许可，本文档中受版权保护的任何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：政府使用、复制或公开本文档受 DFARS 252.227-7013（2014 年 2 月）和 FAR 52.227-19（2007 年 12 月）中“技术数据权利 — 非商用”条款第 (b)(3) 条规定的限制条件的约束。

本文档中所含数据与商业产品和/或商业服务（定义见 FAR 2.101）相关，属于 NetApp, Inc. 的专有信息。根据本协议提供的所有 NetApp 技术数据和计算机软件具有商业性质，并完全由私人出资开发。美国政府对这些数据的使用权具有非排他性、全球性、受限且不可撤销的许可，该许可既不可转让，也不可再许可，但仅限在与交付数据所依据的美国政府合同有关且受合同支持的情况下使用。除本文档规定的情形外，未经 NetApp, Inc. 事先书面批准，不得使用、披露、复制、修改、操作或显示这些数据。美国政府对国防部的授权仅限于 DFARS 的第 252.227-7015(b)（2014 年 2 月）条款中明确的权利。

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。